

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY



INTERAKTÍVNA ILUSTRÁCIA TRAJEKTÓRIÍ
DYNAMICKÝCH SYSTÉMOV

BAKALÁRSKA PRÁCA

UNIVERZITA KOMENSKÉHO V BRATISLAVE
FAKULTA MATEMATIKY, FYZIKY A INFORMATIKY

**INTERAKTÍVNA ILUSTRÁCIA TRAJEKTÓRIÍ
DYNAMICKÝCH SYSTÉMOV**

BAKALÁRSKA PRÁCA

Študijný program: Ekonomická a finančná matematika
Študijný odbor: 1114 Aplikovaná matematika
Školiace pracovisko: Katedra aplikovanej matematiky a štatistiky
Vedúci práce: prof. RNDr. Pavel Brunovský, DrSc.



Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ján Komadel
Študijný program: ekonomická a finančná matematika (Jednoodborové štúdium, bakalársky I. st., denná forma)
Študijný odbor: 9.1.9. aplikovaná matematika
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: slovenský

Názov: Interaktívna ilustrácia trajektórií dynamických systémov

Cieľ: Cieľom práce je vypracovať interaktívnu grafickú ilustráciu trajektórií diskrétného logistického populačného modelu a fázových portrétov konzervatívnych systémov s jedným stupňom voľnosti v prostredí MATLAB.

Vedúci: prof. RNDr. Pavel Brunovský, DrSc.

Katedra: FMFI.KAMŠ - Katedra aplikovanej matematiky a štatistiky

Dátum zadania: 15.10.2011

Dátum schválenia: 27.10.2011

doc. RNDr. Margaréta Halická, CSc.
garant študijného programu

.....
študent

.....
vedúci práce

Pod'akovanie Rád by som poďakoval vedúcemu mojej bakalárskej práce prof. RNDr. Pavlovi Brunovskému, DrSc. za námet, usmerňovanie pri písaní a predovšetkým za jeho pozitívny prístup a záujem o túto prácu.

Abstrakt

KOMADEL, Ján: *Interaktívna ilustrácia trajektórií dynamických systémov* [Bakalárska práca], Univerzita Komenského v Bratislave, Fakulta matematiky, fyziky a informatiky, Katedra aplikovanej matematiky a štatistiky; školiteľ: prof. RNDr. Pavel Brunovský, DrSc, Bratislava, 2011, 35 s.

Predmetom tejto práce je vytvorenie dvoch programov v Matlabe na interaktívnu ilustráciu trajektórií dvoch druhov dynamických systémov. V prvej časti sa pozrieme na diskkrétne dynamické systémy zastúpené logistickým modelom populačnej dynamiky. Odvodíme tento model a preberieme jeho zaujímavé vlastnosti, predovšetkým periodické body a ich stabilitu. Tiež zavedieme pojem pavučinového diagramu, ktorý nám pomôže skúmať správanie sa trajektórií. Druhá časť sa zaoberá spojitými dynamickými systémami, ktoré sú zastúpené konzervatívnymi systémami s jedným stupňom voľnosti. Použijeme príklad mechanických systémov pre lepšie porozumenie a tiež zavedieme pojem fázového portréту, ktorý zobrazuje nami skúmané trajektórie. Hlavným prínosom tejto práce sú dva interaktívne programy, ktoré používateľovi umožňujú bližšie preskúmať dané dynamické systémy a tým sa s nimi lepšie zoznámiť.

Kľúčové slová: Dynamický systém, logistický model, pavučinový diagram, konzervatívny systém, fázový portrét

Abstract

KOMADEL, Ján: *Interactive illustration of the trajectories of dynamical systems* [Bachelor Thesis], Comenius University in Bratislava, Faculty of Mathematics, Physics and Informatics, Department of Applied Mathematics and Statistics; Supervisor: prof. RNDr. Pavel Brunovský, DrSc, Bratislava, 2011, 35 p.

The subject of this work is to develop two Matlab programs for interactive graphical illustration of trajectories of two kinds of dynamical systems. In the first part we look at discrete dynamical systems represented by the logistic model of population dynamics (often referred to as logistic map). We derive the model and discuss its interesting features concentrating mainly on the periodical points and their stability. We also introduce the concept of cobweb diagram which will help us explore the behavior of trajectories. The second part concerns continuous dynamical systems represented by conservative systems with one degree of freedom. We use the example of mechanical systems for better understanding. We also introduce the concept of phase portraits which depict the trajectories in which we are interested. The main contribution of this work are the two interactive programs which allow the user to examine dynamical systems more closely and thus get more familiar with them.

Keywords: Dynamical system, logistic map, cobweb diagram, conservative system, phase portrait

Obsah

Zoznam použitých symbolov	8
Úvod	9
1 Logistický model	10
1.1 Definícia pojmov	10
1.2 Odvodenie modelu	11
1.2.1 Malthusov populačný model	12
1.2.2 Logistický model populačnej dynamiky	12
1.3 Vlastnosti	13
1.3.1 Pevné body a ich stabilita	13
1.3.2 Pavučinový diagram	15
1.3.3 Atraktivita bodov	16
1.3.4 Periodické body	20
1.3.5 Chaos	24
1.4 Program LogistickyModel	25
1.4.1 Spustenie programu	25
1.4.2 Prehľad funkcií	25
2 Konzervatívne systémy	29
2.1 Definícia pojmov	29
2.2 Zachovanie energie	31
2.2.1 Gulička v miske	31
2.3 Fázový portrét	32
2.3.1 Postup zakresľovania	35
2.3.2 Inerpretácia	36
2.4 Program FazovyPortret	38
2.4.1 Spustenie programu	38
2.4.2 Užívateľské rozhranie	38
2.4.3 Nedostatok programu	41
Záver	43

Zoznam použitej literatúry	44
Príloha A - Lagrangeova veta o strednej hodnote	45
Príloha B - Zdrojový kód programu LogistickyModel	46
Príloha C - Zdrojový kód programu FazvyPortret	55

Zoznam použitých symbolov

\mathbb{R}	množina reálnych čísel
\mathbb{R}^+	množina kladných reálnych čísel
$\mu \approx 0$	μ sa približne rovná nule
$\{x(t)\}_{t=1,2,\dots}$	Postupnosť $x(1), x(2), x(3), \dots$
$f \in C^1$	funkcia f a jej prvá derivácia sú spojité
$f^2(x)$	$f^2(x) := f[f(x)]$
$\mathcal{O}(x)$	okolie bodu x ; $\mathcal{O}(x) = (x - \delta, x + \varepsilon)$, $\delta, \varepsilon \in \mathbb{R}^+$
\dot{x}	$\dot{x} := \frac{dx}{dt}$
\ddot{x}	$\ddot{x} := \frac{d^2x}{dt^2}$

Úvod

Človek je odjakživa fascinovaný javmi okolo seba a snaží sa im porozumieť. V tom mu často pomáha aj matematika, pomocou ktorej je možné rôzne procesy opísať a často aj predpovedať. Na to neraz slúžia dynamické systémy, či už diskkrétne alebo spojité, ktoré sa používajú v matematickom modelovaní.

V prvej kapitole sa budeme venovať logistickému modelu populačnej dynamiky, ktorý veľmi jednoducho opisuje vývoj homogénnej populácie jedného druhu žijúcej izolovane od iných druhov. Logistický model predstavuje jediná diferenčná rovnica, no napriek tomu bol v posledných desaťročiach predmetom záujmu viacerých autorov. To nebolo zapríčinené tým, ako sa dá tento model aplikovať na reálny svet, ale pozoruhodné sú jeho matematické vlastnosti. Tým sa budeme venovať aj my a vyvinieme interaktívny program, pomocou ktorého sa budú dať graficky ilustrovať.

V druhej kapitole prejdeme k spojitým dynamickým systémom. Konkrétne sa zameriame na konzervatívne systémy s jedným stupňom voľnosti, ktoré budeme aplikovať na mechanické systémy. Tu si ukážeme, ako sa dá napríklad opísať pohyb guľičky v miske pod vplyvom gravitácie pomocou diferenciálnej rovnice. Tak ako v prvej kapitole, aj v tejto časti vytvoríme interaktívny program, ktorý bude systém ilustrovať a tým nám uľahčí vysvetľovanie danej problematiky.

Naším cieľom je priblížiť dynamické systémy čitateľovi a poskytnúť mu možnosť lepšie sa zoznámiť s ich trajektóriami pomocou dvoch interaktívnych programov na ich ilustráciu. Tieto programy môžu tiež slúžiť ako didaktické pomôcky pri výučbe, keďže umožňujú vyučujúcemu poukázať na dôležité vlastnosti dynamických systémov.

1 Logistický model populačnej dynamiky

V tejto kapitole sa budeme zaoberať logistickým modelom populačnej dynamiky, ktorý je jedným z najjednoduchších matematických modelov opisujúcich prírodné javy (v texte budeme používať skrátené pomenovanie logistický model). Zameriava sa na vývoj izolovanej homogénnej populácie, kde početnosť súčasnej generácie určuje početnosť tej nasledujúcej a dajú sa na ňom pozorovať zaujímavé matematické vlastnosti, ktoré ako prvý opísal R. May v 70. rokoch v [4]. Skôr ako sa budeme venovať tomuto modelu podrobnejšie, je potrebné zadať niekoľko pojmov.

1.1 Definícia pojmov

Pod dynamickým systémom budeme v tejto kapitole rozumieť diskretný systém daný rekurentným vzťahom

$$x(t+1) = f(x(t)), \quad (1)$$

ktorého postupnosti $\{x(t)\}_t$ majú hodnoty v stavovom priestore $X \subset \mathbb{R}$. Pokračujeme definovaním základných pojmov, s ktorými budeme pracovať, a taktiež uvedieme niekoľko tvrdení o dynamických systémoch. Vychádzať budeme z definícií a tvrdení uvedených v [2].

Definícia 1.1 (Trajektória). *Spomínanú postupnosť $\{x(t)\}_{t=0,1,2,\dots}$, ktorá spĺňa (1) a $x(0) = x_0$ nazývame trajektóriou dynamického systému (1) cez bod x_0 (skrátene trajektóriou bodu x_0).*

Definícia 1.2 (Pevný bod, periodický bod, r -cyklus). *Bod $\hat{x} \in \mathbb{R}$ je pevným bodom zobrazenia f , ak spĺňa*

$$\hat{x} = f(\hat{x}). \quad (2)$$

Pevný bod zobrazenia $f^r = f \circ f \circ \dots \circ f$ sa nazýva periodickým bodom periódy r ; ak r je jeho minimálna perióda, nazýva sa r -cyklom.

Pevný bod \hat{x} nazývame:

- stabilným, ak pre každé $\varepsilon > 0$ existuje $\delta > 0$ také, že pre ľubovoľný počiatočný bod x_0 zo stavového priestoru X platí, že ak vzdialenosť $|x_0 - \hat{x}| < \delta$, potom $|x(t, x_0) - \hat{x}| < \varepsilon$ pre všetky $t \geq 0$;

- asymptoticky stabilným, ak existuje také kladné ε_0 , že ak pre ľubovoľný počiatočný bod x_0 zo stavového priestoru X platí, že vzdialenosť $|x_0 - \hat{x}| < \varepsilon_0$, potom postupnosť $\{x(t, x_0)\}_t$ konverguje k \hat{x} pre $t \rightarrow \infty$;
- nestabilným, ak nie je stabilný.

Veta 1.3 (Kritéria stability). Nech f je C^1 a nech $\hat{x} \in \mathbb{R}$ je pevný bod f . Potom \hat{x} je

- asymptoticky stabilný, ak $|f'(\hat{x})| < 1$ a
- nestabilný, ak $|f'(\hat{x})| > 1$.

Dôkaz. Nech najprv $|f'(\hat{x})| < 1$. Potom existuje $\varepsilon > 0$ také, že $|f'(x)| < \alpha < 1$ pre $|x - \hat{x}| < \varepsilon$. Ak $|x - \hat{x}| < \varepsilon$, potom podľa Lagrangeovej vety o strednej hodnote (pozri Prílohu A)

$$|f(x) - \hat{x}| = |f(x) - f(\hat{x})| = |f'(\vartheta)||x - \hat{x}| \leq \alpha|x - \hat{x}|, \quad (3)$$

kde ϑ je medzi x a \hat{x} . Ak $|x(0) - \hat{x}| < \varepsilon$, potom z (3) dostávame indukciou

$$|x(t) - \hat{x}| < \varepsilon, \quad |x(t) - \hat{x}| < \alpha^t|x(0) - \hat{x}|.$$

Ak $|f'(\hat{x})| > 1$, podobne existuje $\varepsilon > 0$ a $\alpha > 1$ pre $|x - \hat{x}| < \varepsilon$. Ak $|x - \hat{x}| < \varepsilon$, platí z podobných dôvodov ako v predchádzajúcom prípade,

$$|f(x) - \hat{x}| \geq \alpha|x - \hat{x}|,$$

preto

$$|x(t) - \hat{x}| \geq \alpha^t|x(0) - \hat{x}|, \text{ pokiaľ } |x(s) - \hat{x}| \leq \varepsilon \text{ pre } 0 \leq s \leq t.$$

Nech by x_0 bolo ľubovoľne blízke \hat{x} , pre $t \geq \frac{\ln \varepsilon - \ln |x(0) - \hat{x}|}{\ln \alpha}$, platí

$$|x(t) - \hat{x}| \geq \alpha^t|x(0) - \hat{x}| \geq \varepsilon.$$

□

1.2 Odvodenie modelu

Podobne ako v [2], logistický model odvodíme pomocou jednoduchšieho modelu a použijeme rovnaké značenie.

1.2.1 Malthusov populačný model

Pravdepodobne najjednoduchším populačným modelom je Malthusov model, ktorý lineárnym vzťahom opisuje populáciu s neprekrývajúcimi sa generáciami:

$$x(t+1) = bx(t), \quad (4)$$

kde $x(t)$ je početnosť v čase t a $b > 0$ je priemerný počet potomkov jedinca. Z matematického predpisu modelu (4) je zjavné, že populácia

- rastie, ak $b > 1$,
- stagnuje, ak $b = 1$, a
- znižuje sa, ak $b < 1$.

1.2.2 Logistický model populačnej dynamiky

Logistický model dostaneme, ak v Malthusovom modeli zohľadníme, že rast populácie $r(x)$ nebude už rovný konštante b , ale bude sa lineárne spomaľovať s rastúcou populáciou, napr. kvôli obmedzeným zdrojom potravy:

$$r(x) = q(K - x).$$

Parameter K možno interpretovať ako maximálny možný stav populácie, nakoľko bude nasledujúca generácia nulová, ak jej súčasná veľkosť x dosiahne hodnotu K a teda populácia vyhynie. Rýchlosť rastu je vyjadrená pomocou q . Dostávame tak logistický model v tvare:

$$x(t+1) = f(x(t)), \quad (5)$$

kde

$$f(x) = r(x)x = q(K - x)x.$$

Použitím transformácie $x = Ky$, $qK = \mu$ dostávame z (5) normalizovaný tvar:

$$y(t+1) = \mu y(1 - y).$$

Transformovaná veličina $y(t)$ už nie je absolútna početnosť, ale hovorí nám, akú časť maximálnej možnej populácie v čase t tvorí súčasná populácia. Je to teda číslo medzi

nulou a jednotkou. Parameter μ v sebe zahŕňa mieru reprodukcie ako aj obmedzenia zdrojov.

Ďalej budeme v texte pod logistickým modelom populačnej dynamiky rozumieť vzťah

$$x(t+1) = f_\mu(x(t)), \quad (6)$$

kde

$$f_\mu(x) = \mu x(1-x).$$

1.3 Vlastnosti

Ako sme už uviedli v časti 1.2.2, zmysluplné sú len hodnoty $x \in [0, 1]$, preto tento interval zvolíme za stavový priestor X . Obmedzením parametra μ na $0 \leq \mu \leq 4$ dosiahneme, že pre ľubovoľnú počiatočnú hodnotu $x_0 \in [0, 1]$ bude celá postupnosť generovaná vzťahom (6) pre $t = 0, 1, 2, \dots$ patriť do $X = [0, 1]$:

$$\max_{x \in [0,1]} f_\mu(x) = \max_{x \in [0,1]} \{\mu x(1-x)\} = \mu \frac{1}{2} \left(1 - \frac{1}{2}\right) = \frac{\mu}{4} \leq 1.$$

1.3.1 Pevné body a ich stabilita

Pevnými bodmi logistického modelu (6) sú riešenia rovnice

$$\hat{x} = \mu \hat{x}(1 - \hat{x}). \quad (7)$$

Zjavným riešením je bod $\hat{x}_0 = 0$. Pre $\mu \geq 1$ je druhým riešením $\hat{x}_1 = 1 - \frac{1}{\mu}$.

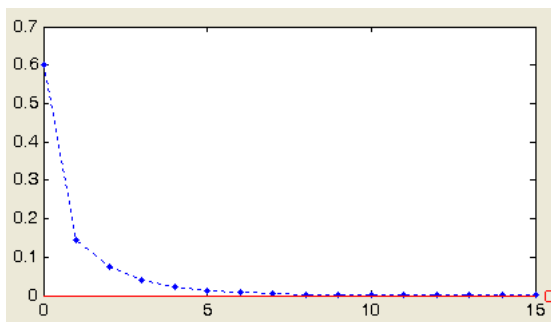
Venujme sa teraz ich stabilite, ktorú vieme podľa Vety 1.3 posúdiť na základe hodnoty

$$f'_\mu(\hat{x}) = \mu(1 - 2\hat{x}). \quad (8)$$

Platí

$$f'_\mu(\hat{x}_0) = f'_\mu(0) = \mu,$$

preto je bod $\hat{x}_0 = 0$ asymptoticky stabilný pre $0 \leq \mu < 1$ a nestabilný pre $\mu > 1$. Obr. 1 ilustruje stabilitu tohto pevného bodu pre $\mu = 0,6$. Zvolený je počiatočný bod $x_0 = 0,3$. Červenou farbou je zakreslený pevný bod a modrou trajektória $\{x(t)\}_{t=0}^{15}$, ktorá zjavne klesá k bodu $\hat{x}_0 = 0$.

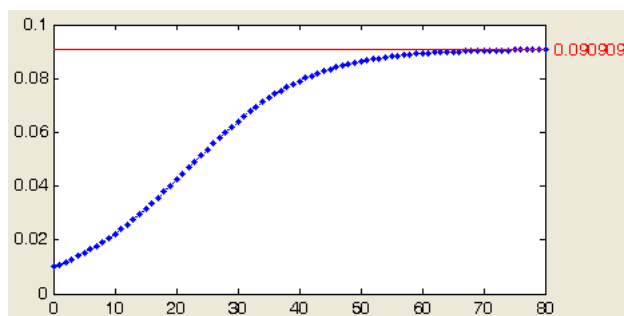


Obr. 1: Logistický model s parametrami $\mu = 0,6$ a $x_0 = 0,3$. Zobrazené sú $x(0) - x(15)$.

Pre bod \hat{x}_1 dostávame z (8)

$$f'_\mu(\hat{x}_1) = f'_\mu\left(1 - \frac{1}{\mu}\right) = \mu\left(-1 + \frac{2}{\mu}\right) = 2 - \mu,$$

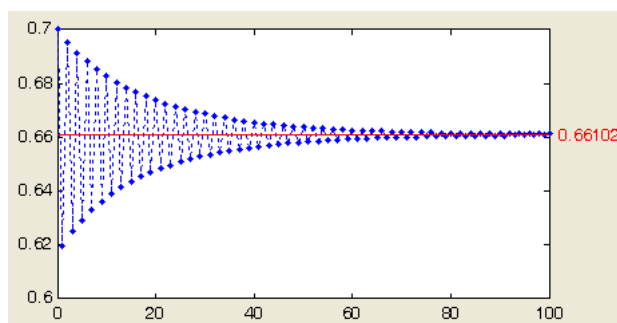
takže $\hat{x}_1 = 1 - \frac{1}{\mu}$ je stabilný, ak $1 < \mu < 3$. Platí teda, že pre $0 \leq \mu < 1$ má logistický model asymptoticky stabilný pevný bod $\hat{x}_0 = 0$. Ten však stráca stabilitu pri $\mu = 1$ a práve vtedy ju získava bod \hat{x}_1 . Na Obr. 2 vidíme, že pri hodnote $\mu = 1,1$ a počiatocnom bode $x_0 = 0,01$ zvolenom blízko nestabilného $\hat{x}_0 = 0$ sa trajektória vzdáľuje od tohto bodu a blíži sa k stabilnému $\hat{x}_1 = 1 - \frac{1}{\mu} = 1 - \frac{1}{1,1} \approx 0,0909$.



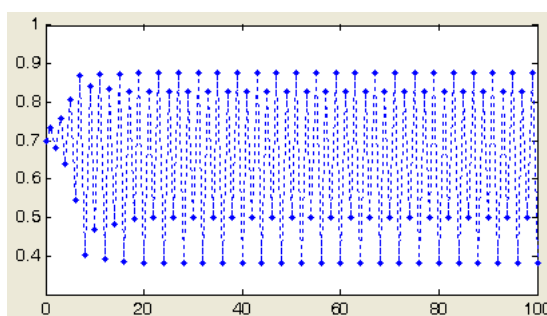
Obr. 2: Logistický model s parametrami $\mu = 1,1$ a $x_0 = 0,01$. Zobrazené sú $x(0) - x(80)$.

Obr. 3 zobrazuje logistický model s $\mu = 2,95$, kedy je ešte bod \hat{x}_1 stabilný. Vidíme, že $x(t)$ oscilujú okolo $\hat{x}_1 \approx 0,66102$ a postupne sa k nemu blížia, pričom $x(100) \approx 0,6612$. Rýchlosť, ktorou sa trajektória blíži k pevnému bodu je už nižšia, čo je prejavom toho, že μ je zvolené blízko trojky, kedy \hat{x}_1 stráca stabilitu.

Pre hodnoty $\mu \geq 3$ neexistuje žiaden stabilný pevný bod, čo sa dá pozorovať na Obr. 4, kde je znázornený model s $\mu = 3,5$ a $x_0 = 0,7$. Trajektória osciluje a hodnoty sa neblížia k spoločnej limite.



Obr. 3: Logistický model s parametrami $\mu = 2,95$ a $x_0 = 0,7$. Zobrazené sú $x(0) - x(100)$.



Obr. 4: Logistický model s parametrami $\mu = 3,5$ a $x_0 = 0,7$. Zobrazené sú $x(0) - x(100)$.

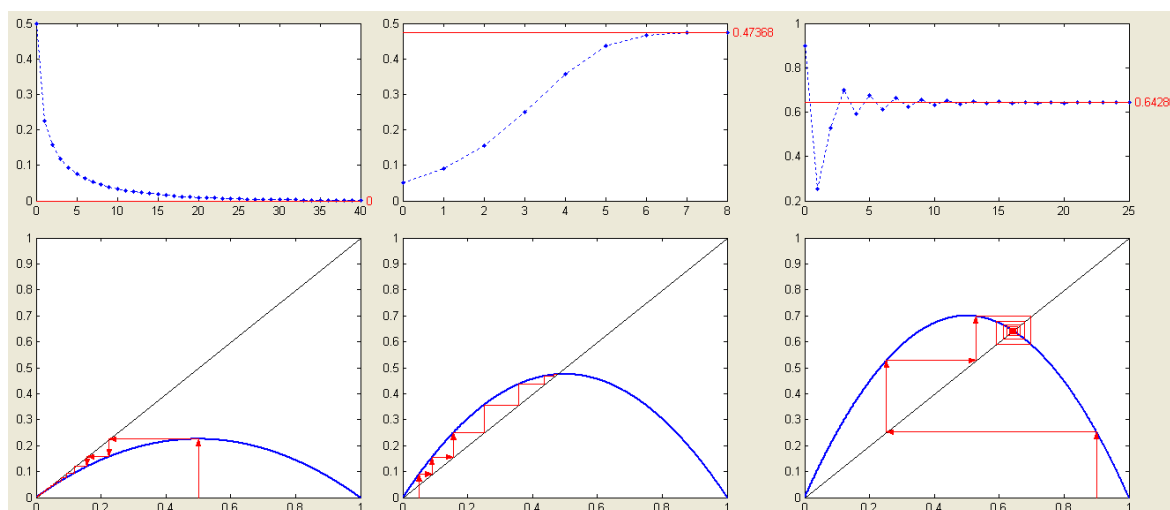
1.3.2 Pavučinový diagram

Iným spôsobom znázornenia logistického modelu, ako zakreslenie samotnej trajektórie $\{x(t)\}$ v závislosti od t , je pavučinový diagram. V tomto prípade zakreslíme do jedného obrázku grafy $y = f_\mu(x)$ a $y = x$. Postupnosť následne generujeme graficky, počnúc zakreslením bodu $[x(0), 0]$ a pokračujeme nasledovnými úsečkami:

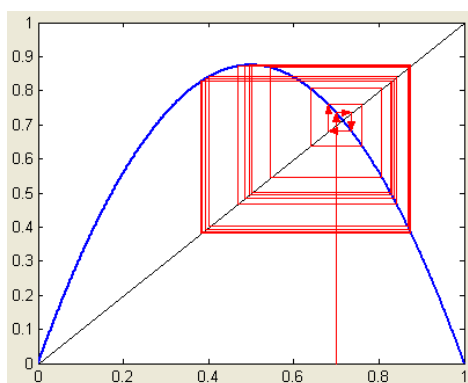
$$[x(0), 0] \rightarrow [x(0), x(1)] \rightarrow [x(1), x(1)] \rightarrow [x(1), x(2)] \rightarrow [x(2), x(2)] \rightarrow [x(2), x(3)] \rightarrow \dots,$$

kde $x(t+1) = f_\mu(x(t))$. Prakticky to znamená spájanie grafov $y = f_\mu(x)$ a $y = x$ striedavo vo vertikálnom a horizontálnom smere. Pavučinové diagramy pre logistické modely s hodnotami parametra μ rovnými 0,9; 1,9 a 2,8 môžeme vidieť v spodnej časti Obr. 5. Z obrázku je zreteľné, že pevné body, ktoré sme získali zo vzťahu (7), skutočne ležia na priesečníkoch grafov.

Uviedli sme príklady, kedy mal model pevný bod, no vráťme sa k prípadu z Obr. 4 a ukážme si, ako vyzerá pavučinový diagram vtedy. Obr. 6 potvrdzuje, že trajektória sa neustáli v jednom bode, ale zdá sa, že tam je určitá pravidelnosť, ktorú možno pozorovať aj na Obr. 4 a ktorej sa budeme ďalej v tejto práci venovať.



Obr. 5: Logistický model s parametrami $(\mu = 0,9; x_0 = 0,5)$; $(\mu = 1,9; x_0 = 0,05)$; resp. $(\mu = 2,8; x_0 = 0,9)$ a príslušné pavučinové diagramy. Zobrazené sú klesajúca, rastúca a oscilujúca trajektória, pričom všetky majú limitu v pevnom bode.



Obr. 6: Pavučinový diagram pre parametre $\mu = 3,5$ a $x_0 = 0,7$. Zakreslených je prvých 50 iterácií.

1.3.3 Atraktivita bodov

Stabilitu pevných bodov sme už rozoberali v časti 1.3.1, ale teraz sa na ňu pozrieme z iného pohľadu.

$\mu < 1$:

Začnime bodom $\hat{x}_0 = 0$, o ktorom sme povedali, že je stabilný pre $0 \leq \mu < 1$. Zdôvodnili sme to vyčíslením derivácie f'_μ v tomto bode a použitím Vety 1.3. Je však \hat{x}_0 skutočne jediným atraktorom v danom modeli?

Lahko sa môžeme presvedčiť, že

$$f_\mu(x) < x \quad \forall x \in (0, 1], \quad (9)$$

lebo f_μ aj x sú spojité funkcie, rovnosť $f_\mu(x) = x$ nastáva podľa vzťahu (7) len v $\hat{x}_0 = 0$ a $\hat{x}_1 = 1 - \frac{1}{\mu} < 1$ a napríklad $f_\mu(\frac{1}{2}) = \frac{\mu}{4} < \frac{1}{2}$. Z toho plynie (9) a teda každý bod $x \in (0, 1]$ sa zobrazením f_μ zmenší. Preto je postupnosť daná vzťahom

$$x_{n+1} = f_\mu(x_n)$$

klesajúca a keďže je zdola ohraničená nulou, platí

$$\lim_{n \rightarrow \infty} x_n = 0$$

pre ľubovoľné $x_0 \in (0, 1]$. Ak by sme za x_0 zvolili nulu, dostaneme identicky nulovú postupnosť, takže predchádzajúci vzťah platí pre každé $x_0 \in X$. Tým sme ukázali, že $\hat{x}_0 = 0$ je skutočne jediným atraktorom v logistickom modeli s $0 \leq \mu < 1$.

$1 < \mu < 2$:

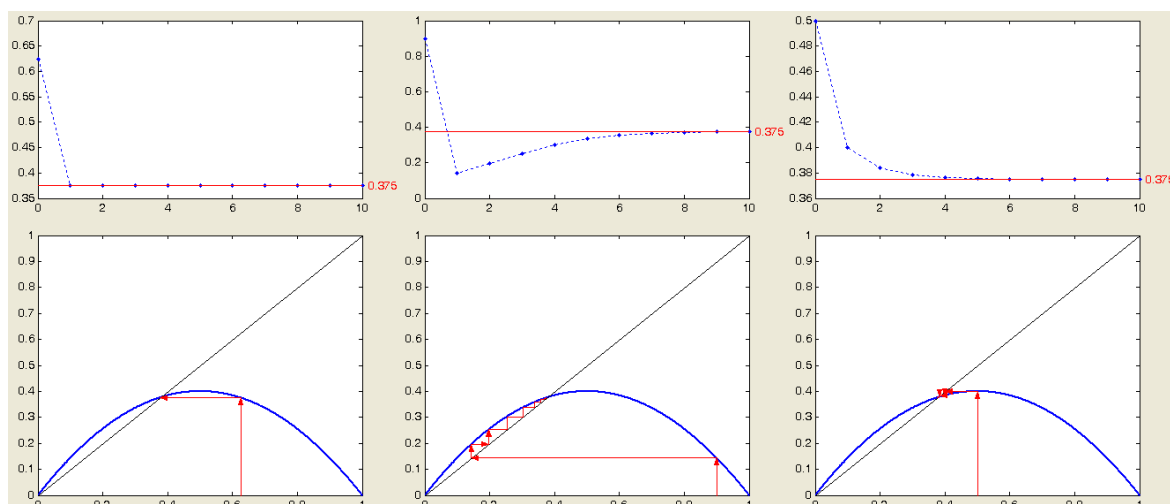
O $\hat{x}_1 = 1 - \frac{1}{\mu}$ sme povedali, že je stabilný pre $1 < \mu < 3$. Zoberme najprv $\mu \in (1, 2)$, pre ktoré podľa (8) platí $f'_\mu(\hat{x}_1) = 2 - \mu > 0$. To znamená, že f_μ je v okolí bodu \hat{x}_1 rastúca. Navyše

$$\hat{x}_1 > f_\mu(x) > x \quad \text{pre } x < \hat{x}_1 \quad \text{a} \quad f_\mu(x) < x \quad \text{pre } x > \hat{x}_1,$$

preto ak zvolíme $x_0 < \hat{x}_1$, dostávame rastúcu postupnosť zhora ohraničenú bodom \hat{x}_1 . Trajektória teda bude monotónne konvergovať k tomuto bodu. V prípade, že zvolíme $x_0 > \hat{x}_1$, môžu nastať tri možnosti:

1. $f_\mu(x_0) = \hat{x}_1$ a trajektória je ďalej konštantná,
2. $f_\mu(x_0) < \hat{x}_1$ a trajektória ďalej vyzerá ako pri voľbe $x_0 < \hat{x}_1$,
3. $f_\mu(x_0) > \hat{x}_1$ a trajektória klesá k \hat{x}_1 , lebo je to opäť monotónna, ohraničená postupnosť.

Na Obr. 7 sú zobrazené všetky tri možnosti pre $\mu = 1,6$.



Obr. 7: Logistický model s $\mu = 1,6$ a počiatočnými bodmi $x_0 = 0,625$; $x_0 = 0,9$ a $x_0 = 0,5$.

$\mu = 2$:

Pre $\mu = 2$ platí $f'_\mu(\hat{x}_1) = 0$ a situácia je podobná, len nemôže nastať tretia možnosť, že trajektória do \hat{x}_1 klesá, lebo f_μ má v pevnom bode \hat{x}_1 maximum.

$2 < \mu < 3$:

Pozrime sa teraz na $\mu \in (2, 3)$, pre ktoré $f'_\mu(\hat{x}_1) \in (0, -1)$. S použitím nasledujúcej vety ukážeme, že pre takýto logistický model budú trajektórie oscilovať (okrem špeciálnych prípadov).

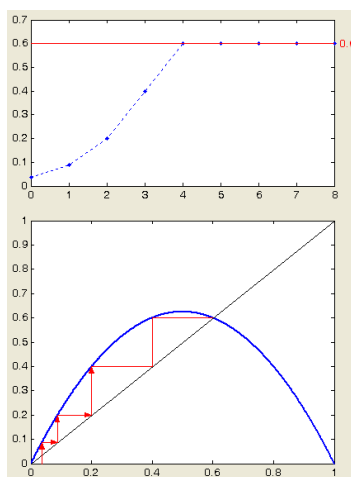
Veta 1.4 (Oscilujúca trajektória). *Nech f je C^1 a nech $\hat{x} \in \mathbb{R}$ je pevný bod f , pričom platí $f'(\hat{x}) < 0$. Potom existuje také $\delta > 0$, že trajektórie dynamického systému daného funkciou f , ktoré pre nejaké t splňajú $0 < |x(t) - \hat{x}| < \delta$, oscilujú.*

Dôkaz. Zo zápornosti derivácie v \hat{x} vyplýva, že f je na nejakom okolí $(\hat{x} - \delta_1, \hat{x} + \delta_2)$ bodu \hat{x} klesajúca. Za δ zvolíme menšie z čísel δ_1, δ_2 . Pre všetky $x \neq \hat{x}$ vzdialené od \hat{x} menej ako δ platí

$$f(x) > \hat{x} > x \quad \text{pre } x < \hat{x} \quad \text{a} \quad f(x) < \hat{x} < x \quad \text{pre } x > \hat{x}.$$

Preto sa $x < \hat{x}$ z takéhoto okolia zobrazí na $x' > \hat{x} > x$ a to sa následne zobrazí na $x'' < \hat{x} < x'$, čím vzniká oscilácia. \square

Poznámka: V danom systéme môžu existovať aj neoscilujúce trajektórie, ktoré konvergujú do iného pevného bodu, alebo sa z oblasti, kde je f neklesajúca, zobrazia priamo do \hat{x} . Takých trejektórií je však len spočítateľne veľa, lebo sú to špeciálne prípady, ktoré sa v konečnom čase zobrazia priamo do pevného bodu. Príklad takejto trajektórie je na Obr. 8.



Obr. 8: Logistický model s $\mu = 2,6$ a počiatočným bodom $x_0 = 0,0364$. Trajektória monotónne konverguje do $\hat{x}_1 = 0,6$, hoci platí $f'_\mu(\hat{x}_1) < 0$.

Logistický model s $\mu \in (2, 3)$ a pevným bodom \hat{x}_1 spĺňa predpoklady Vety 1.4 a teda trajektórie budú oscilovať. Ukážeme, že daný bod je jediným atraktorom. Podobne ako v predošlom prípade, aj tu platí

$$f_\mu(x) < \hat{x}_1 < x \quad \text{pre } x > \hat{x}_1,$$

takže body väčšie ako \hat{x}_1 sa zobrazia na body menšie ako tento bod. Tiež platí

$$f_\mu(x) > x \quad \text{pre } x < \hat{x}_1,$$

čiže body menšie ako \hat{x}_1 sa zobrazením zväčšia. Preto sa trajektória (s výnimkou pre štartovacie body 0 a 1, ktoré sa zobrazia do nuly) po nejakom čase t_0 dostane do oblasti spomínanej v prvej časti dôkazu Vety 1.3, ktorá je daná vzťahom $|x(t) - \hat{x}| < \varepsilon$ a platí v nej kontraktívnosť

$$|f_\mu(x) - \hat{x}_1| \leq \alpha |x - \hat{x}_1|,$$

kde $\alpha < 1$. Zo spomínaného dôkazu vyplýva, že bod \hat{x}_1 je naozaj jediným atraktorom v logistickom modeli s $\mu \in (2, 3)$.

1.3.4 Periodické body

Periodický bod sme definovali v Defínícii 1.2 ako pevný bod zobrazenia f^r . Zamerajme sa najprv na periódu 2, podobne ako [2], ale trochu iným spôsobom. Budú nás teda zaujímať pevné body zobrazenia

$$\begin{aligned} f_\mu^2(x) &= f_\mu[f_\mu(x)] = \\ &= \mu[\mu x(1-x)][1-\mu x(1-x)] \\ &= \mu^2(x-x^2)(1-\mu x+\mu x^2) \\ &= -\mu^3 x^4 + 2\mu^3 x^3 - (\mu^3 + \mu^2)x^2 + \mu^2 x. \end{aligned}$$

Periodické body periódy 2 budú spĺňať

$$\mu^3 \hat{x}^4 - 2\mu^2 \hat{x}^3 + (\mu^3 + \mu^2) \hat{x}^2 - \mu^2 \hat{x} + \hat{x} = 0. \quad (10)$$

Keďže pevné body $f_\mu \hat{x}_0 = 0$ a $\hat{x}_1 = 1 - \frac{1}{\mu}$ sú určite korene (10), môžeme polynóm na ľavej strane vydeliť koreňovými činiteľmi:

$$\begin{aligned} [\mu^3 x^4 - 2\mu^2 x^3 + 2(\mu^3 + \mu^2)x^2 - \mu^2 x + x] : \left\{ x \left[x - \left(1 - \frac{1}{\mu} \right) \right] \right\} = \\ = \mu^3 x^2 - (\mu^3 + \mu^2)x + \mu^2 + \mu \end{aligned}$$

Tento polynóm ešte vydelíme μ a označme

$$F(x, \mu) = \mu^2 x^2 - (\mu^2 + \mu)x + \mu + 1.$$

V tomto značení sú periodické body periódy 2 riešenia rovnice $F(x, \mu) = 0$, ktorými sú

$$\begin{aligned} \hat{x}_2 &= \frac{\mu^2 + \mu + \sqrt{\mu^4 + 2\mu^3 + \mu^2 - 4\mu^2(\mu + 1)}}{2\mu^2} = \frac{\mu + 1 + \sqrt{(\mu - 3)(\mu + 1)}}{2\mu}, \\ \hat{x}_3 &= \frac{\mu^2 + \mu - \sqrt{\mu^4 + 2\mu^3 + \mu^2 - 4\mu^2(\mu + 1)}}{2\mu^2} = \frac{\mu + 1 - \sqrt{(\mu - 3)(\mu + 1)}}{2\mu}. \end{aligned}$$

Dané body existujú len pre $\mu \geq 3$ (záporné μ nepripúšťame). Pozrime sa na ich stabilitu.

Označme

$$g_\mu(x) = (f_\mu^2)'(x) = -4\mu^3 x^3 + 6\mu^3 x^2 - 2(\mu^3 + \mu^2)x + \mu^2$$

a skúmame najprv hodnoty

$$\begin{aligned} g_\mu(\hat{x}_2) &= -4\mu^3 \left(\frac{\mu+1+\sqrt{(\mu-3)(\mu+1)}}{2\mu} \right)^3 + 6\mu^3 \left(\frac{\mu+1+\sqrt{(\mu-3)(\mu+1)}}{2\mu} \right)^2 - \\ &\quad - 2(\mu^3 + \mu^2) \left(\frac{\mu+1+\sqrt{(\mu-3)(\mu+1)}}{2\mu} \right) + \mu^2, \end{aligned}$$

odkiaľ po úprave dostávame

$$g_\mu(\hat{x}_2) = -\mu^2 + 2\mu + 4.$$

Na posúdenie stability 2-cyklu, t.j. pevného bodu systému

$$x(t+1) = f_\mu^2(x(t)),$$

nás zaujíma, kedy $|g_\mu(\hat{x})| < 1$. Ľahko sa dá presvedčiť, že $g_\mu(\hat{x}_2) < 1$ pre $\mu > 3$ a $g_\mu(\hat{x}_2) > -1$ pre $\mu \in (1 - \sqrt{6}, 1 + \sqrt{6})$. Odtiaľ podľa Vety 1.3 dostávame, že periodický bod \hat{x}_2 je stabilný pre $3 < \mu < 1 + \sqrt{6} \approx 3,449$. Navyše platí

$$g_\mu(\hat{x}_3) = g_\mu(\hat{x}_2) = -\mu^2 + 2\mu + 4,$$

takže \hat{x}_3 je stabilný pre rovnaké hodnoty μ ako \hat{x}_2 . Preverme ešte, či f_μ zobrazuje tieto dva periodické body jeden na druhý:

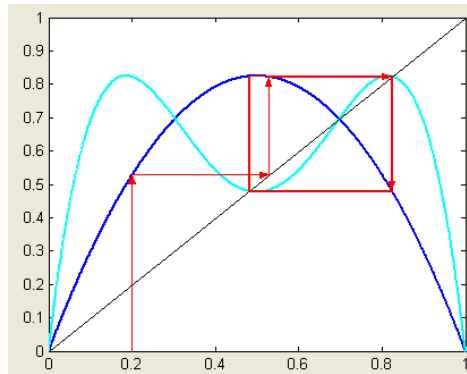
$$\begin{aligned} f_\mu(\hat{x}_2) = \mu \hat{x}_2(1 - \hat{x}_2) &= \mu \frac{\mu+1+\sqrt{(\mu-3)(\mu+1)}}{2\mu} \left(1 - \frac{\mu+1+\sqrt{(\mu-3)(\mu+1)}}{2\mu} \right) \\ &= \frac{1}{4\mu} \left(\mu + 1 + \sqrt{(\mu-3)(\mu+1)} \right) \left(\mu - 1 - \sqrt{(\mu-3)(\mu+1)} \right) \\ &= \frac{1}{4\mu} \left[\mu^2 - \left(1 + \sqrt{(\mu-3)(\mu+1)} \right)^2 \right] \\ &= \frac{\mu+1-\sqrt{(\mu-3)(\mu+1)}}{2\mu} = \hat{x}_3. \end{aligned}$$

Analogicky sa ukáže $f_\mu(\hat{x}_3) = \hat{x}_2$.

Tým sme dokázali stabilitu uvedeného 2-cyklu pre $3 < \mu < 1 + \sqrt{6} \approx 3,449$. Ak by sme pokračovali v hľadaní stabilných cyklov podobným spôsobom, potrebovali by sme riešiť rovnice vyššieho rádu. Tie nevieme vo všeobecnosti riešiť analyticky, takže by bolo treba použiť numerické metódy a celý proces by sa skomplikoval. Periodickými bodmi logistického modelu sa bližšie zaoberá [3].

Na Obr. 9 je zobrazený pavučinový diagram logistického modelu pre $\mu = 3,3$. Bledomodrou je zakreslená funkcia $f_{3,3}^2$, ktorá má štyri priesečníky s identickou funkciou $y = x$. Sú nimi, pochopiteľne, body $\hat{x}_0 = 0$ a $\hat{x}_1 = 1 - \frac{1}{\mu} \approx 0,7$, ktoré však už nie sú stabilné. Ostatné dva priesečníky sú zjavne periodickými bodmi periódy 2, keďže vidíme, že sa vzájomne zobrazujú jeden na druhý. Ako sme už spomínali vyššie, tieto body sú riešením rovnice $F(x, \mu)|_{\mu=3,3} = 0$, t.j.

$$3,3^2 x^2 - (3,3^2 + 3,3) x + 3,3 + 1 = 0.$$



Obr. 9: Pavučinový diagram pre parametre $\mu = 3,3$ a $x_0 = 0,2$. Zakreslených je prvých 30 iterácií a funkcia $f_{3,3}^2$.

Odtiaľ dostávame

$$\hat{x}_{2,3} = \frac{3,3 + 1 \pm \sqrt{0,3 \cdot 4,3}}{2 \cdot 3,3} \approx \begin{cases} 0,8236 \\ 0,4794 \end{cases}.$$

Môžeme ľahko overiť, že skutočne

$$f_{3,3}(0,8236) = 3,3 \cdot 0,8236 \cdot (1 - 0,8236) \approx 0,4794$$

a

$$f_{3,3}(0,4794) = 3,3 \cdot 0,4794 \cdot (1 - 0,4794) \approx 0,8236.$$

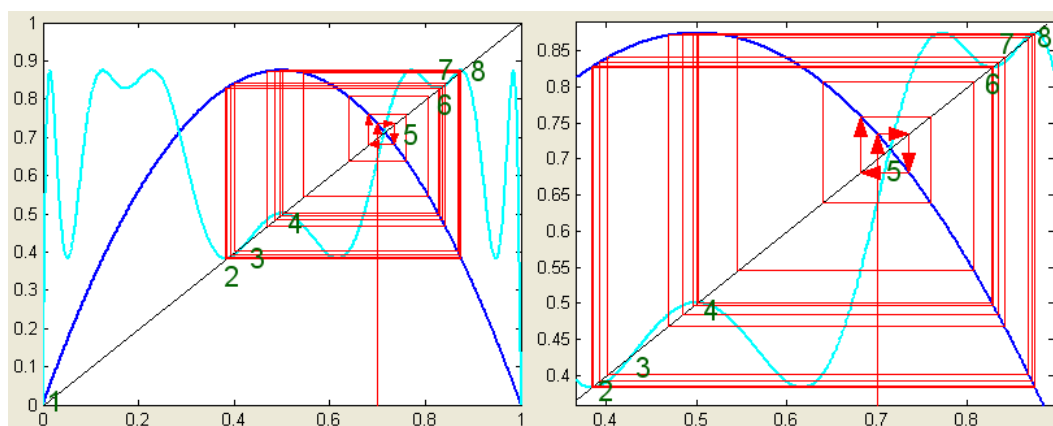
Podľa [3] ďalej platí, že po strate stability 2-cyklu okolo $\mu = 3,449$ získava stabilitu 4-cyklos a existuje postupnosť $\mu_1 < \mu_2 < \mu_3 < \dots$ hodnôt parametra μ taká, že keď μ prekročí hodnotu μ_k , stabilný 2^{k-1} -cyklus stratí stabilitu a získa ju 2^k -cyklus. Prvých niekoľko členov tejto postupnosti je

$$\mu_1 = 3; \quad \mu_2 = 3,449; \quad \mu_3 = 3,544; \quad \mu_4 = 3,564; \quad \dots$$

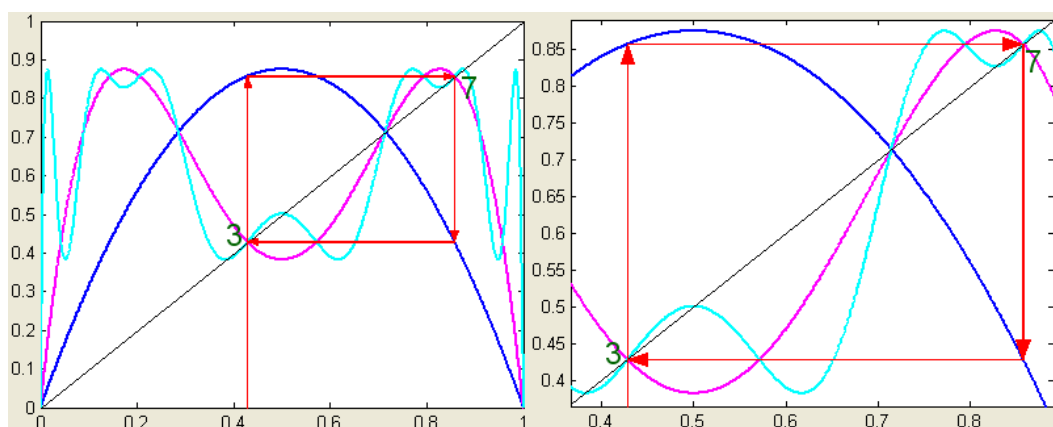
Táto postupnosť konverguje s rastúcim k k číslu $\mu_\infty \approx 3,570$ a navyše platí, že pre rozdiely medzi po sebe idúcimi parametrami platí

$$\lim_{k \rightarrow +\infty} \frac{\mu_k - \mu_{k-1}}{\mu_{k+1} - \mu_k} \approx 4,6692.$$

Na ilustráciu posluží Obr. 10, ktorý zobrazuje rovnaký prípad ako Obr. 4 a Obr. 6, len je tam zakreslená aj funkcia $f_{3,5}^4$. Podľa prvých 50 iterácií sa dá usúdiť, že sú tam prítomné štyri periodické body, ktorými sú priesečníky číslo 2, 4, 6 a 8 zobrazených



Obr. 10: Pavučinový diagram pre parametre $\mu = 3,5$ a $x_0 = 0,7$. Zakreslených je prvých 50 iterácií a funkcia $f_{3,5}^4$. Napravo je približená centrálna časť diagramu.

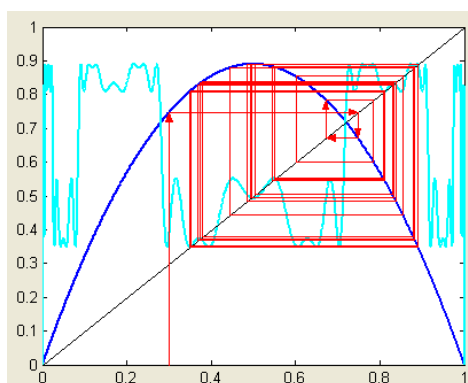


Obr. 11: Pavučinový diagram pre parametre $\mu = 3,5$ a $x_0 = 0,4285$. Zakreslených je prvých 10 iterácií a funkcie $f_{3,5}^2$ (fialová) a $f_{3,5}^4$ (bledomodrá). Napravo je približená centrálna časť diagramu.

grafov. Čísla 1 a 5 označujú nestabilné pevné body \hat{x}_0 a \hat{x}_1 a priesečníky 3 a 7 vysvetlíme pomocou Obr. 11.

Vidíme, že body 3 a 7 z Obr. 10 sú spoločnými priesečníkmi grafov funkcií $f_{3,5}^2$ (fialová) a $f_{3,5}^4$ (bledomodrá) s identickou funkciou. Ako je na obrázku ukázané, tieto dva body tvoria 2-cyklus, ktorý síce pri $\mu = 3,5$ nie je stabilný, ale stále sa v modeli vyskytuje. Vznikom 2^k -cyklu teda 2^{k-1} -cyklus nezaniká, len stráca stabilitu.

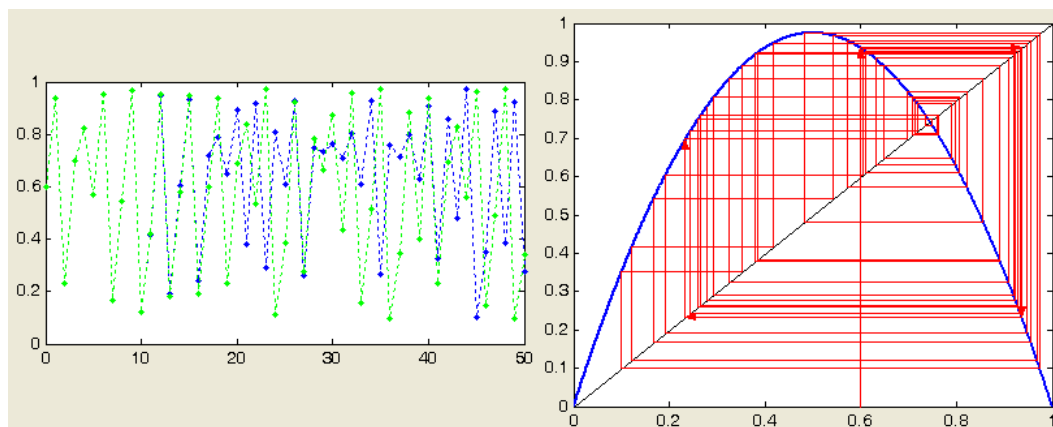
Obr. 12 znázorňuje logistický model s $\mu = 3,56$, kde môžeme pozorovať stabilný 8-cyklus, ktorý však nie je úplne zreteľný, lebo na ustálenie by bolo potrebných výrazne viac ako 50 iterácií.



Obr. 12: Pavučinový diagram pre parametre $\mu = 3,56$ a $x_0 = 0,3$. Zakreslených je prvých 50 iterácií a funkcia $f_{3,5}^8$.

1.3.5 Chaos

Doteraz sme teda opísali dynamiku logistického modelu pre hodnoty $0 < \mu \leq 3,570$. Zaujímavé je pozrieť sa aj na hodnoty blízke 4, kde sa dá systém označiť za chaotický. To znamená, že je veľmi citlivý na zmenu vstupných údajov a minimálna zmena počiatočného bodu spôsobí veľké zmeny výstupov systému. Ukážeme si to na Obr. 13, kde je zobrazený model s $\mu = 3,9$ pre počiatočné body $x_0 = 0,6$ (modrá) a $x_0 = 0,60001$ (zelená).



Obr. 13: Logistický model s $\mu = 3,9$ a $x_0 = 0,6$ (modrá), resp. $x_0 = 0,60001$ (zelená). Zakreslených je prvých 50 iterácií. Na pavučinovom diagrame je zobrazený len systém pre $x_0 = 0,6$.

Vidíme, že spočiatku sú trajektórie veľmi blízke, na obrázku sa prekrývajú, ale okolo 15. iterácie sa začínajú výrazne líšiť a ďalej sa správajú úplne odlišne. Na pavučinovom diagrame, ktorý je len pre $x_0 = 0,6$, nepozorujeme nijakú pravidelnosť a systém je

ťažko predvídateľný. Systém z Obr. 13 je príkladom *efektu motýlieho krídla* (butterfly effect), ako americký matematik a meteorológ Edward Lorenz označil citlivú závislosť na počiatočných podmienkach. Tento názov pochádza z teoretického príkladu, že niečo na prvý pohľad bezvýznamné ako mávnutie krídiel motýľa môže spôsobiť hurikán na druhej strane Zeme. Podobne bezvýznamne môže pôsobiť zmena počiatočného bodu o jednu stotisícinu, ktorá však zapríčinila významnú zmenu výsledkov.

Práve chaotické vlastnosti logistického modelu majú praktické využitie v informatike. Používajú sa napríklad pri generovaní pseudonáhodných čísel, o ktorom viac píše autori v [7], alebo pri kódovaní obrázkov, ktorému sa venuje [6].

1.4 Program LogistickyModel

Motiváciou tejto práce bolo vypracovanie programov, ktoré budú môcť byť použité pri výučbe na interaktívnu grafickú ilustráciu trajektórií dynamických systémov. Prvým z týchto programov je LogistickyModel, ktorý slúži na vysvetlenie logistického modelu populačnej dynamiky. Tento program sme vytvorili v jazyku Matlab a pozostáva zo súborov LogistickyModel.fig a LogistickyModel.m. Prvý súbor vznikol v prostredí GUIDE (Graphical User Interface Development Environment), ktoré umožňuje používateľovi jednoducho navrhnuť grafické rozhranie. Druhý súbor obsahuje jadro programu. Bol vygenerovaný pomocou GUIDE a následne sme naprogramovali jeho funkcie.

1.4.1 Spustenie programu

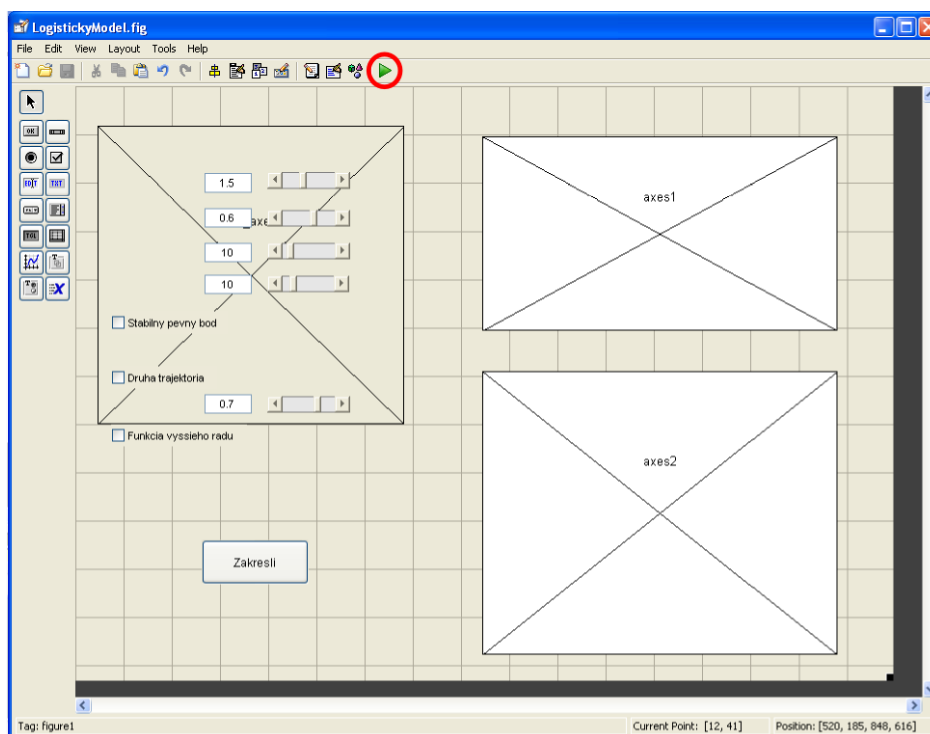
Pred samotným spustením programu treba zadať príkaz

```
guide LogistickyModel
```

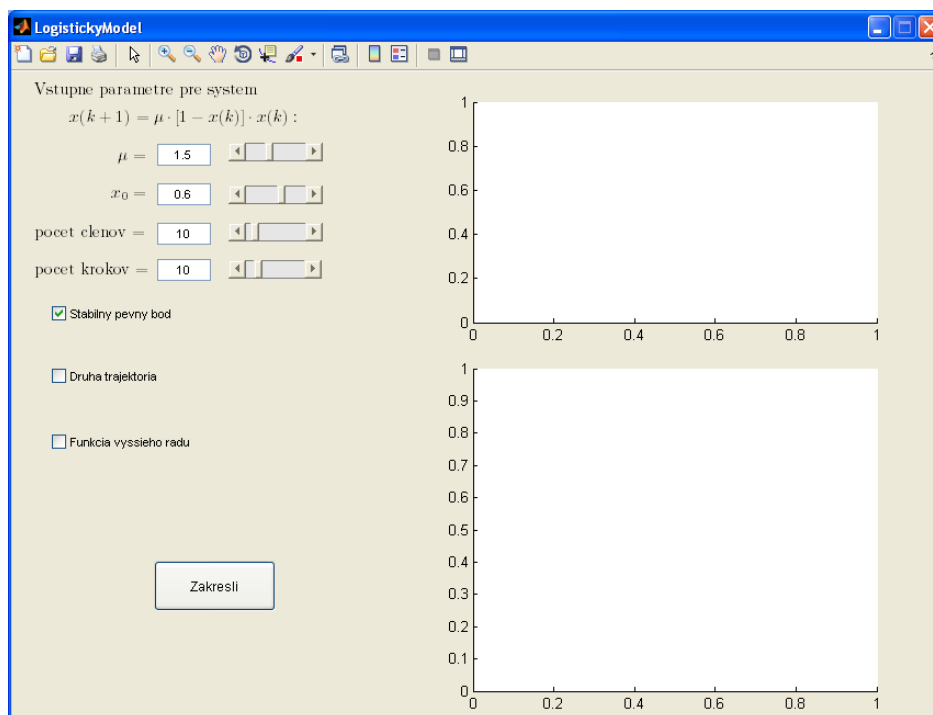
do príkazového okna. Otvorí sa súbor LogistickyModel.fig v prostredí GUIDE, ktoré dovoľuje upravovať jednotlivé objekty, alebo prejsť k spusteniu programu stlačením zeleného tlačidla v hornom menu (Obr. 14).

1.4.2 Prehľad funkcií

Po spustení programu sa otvorí okno s užívateľským rozhraním programu (Obr. 15). V hornej časti je základný ovládací panel, ktorý umožňuje interaktívnu prácu s vykresľovacími plochami. Pre náš program zaujímavé funkcie sú hlavne priblíženie, oddialenie,



Obr. 14: Súbor LogistickyModel.fig so všetkými objektami. Zelené tlačidlo na spustenie programu je v hornom ovládacom paneli.



Obr. 15: Užívateľské rozhranie programu LogistickyModel po spustení s pôvodnými nastaveniami.

hýbanie grafu a zistenie súradníc bodov. Pod ovládacím panelom sa nachádza časť, kde užívateľ nastaví parametre, pre ktoré chce logistický model (6) zakresliť. Sú to hodnoty μ a x_0 , ktoré určujú model, ako aj hodnoty *pocet clenov*, ktorá určuje, koľko členov trajektórie $\{x(t)\}$ sa zakreslí, a *pocet krokov*, ktorá zohľadňuje počet iterácií zakreslených do pavučinového diagramu. Nastavenie je možné zadaním želanej hodnoty priamo alebo pomocou posuvnej lišty.

Parameter	h_0	h_{min}	h_{max}
μ	1,5	0	4
x_0	0,6	0	1
pocet clenov	10	2	100
pocet krokov	10	2	50

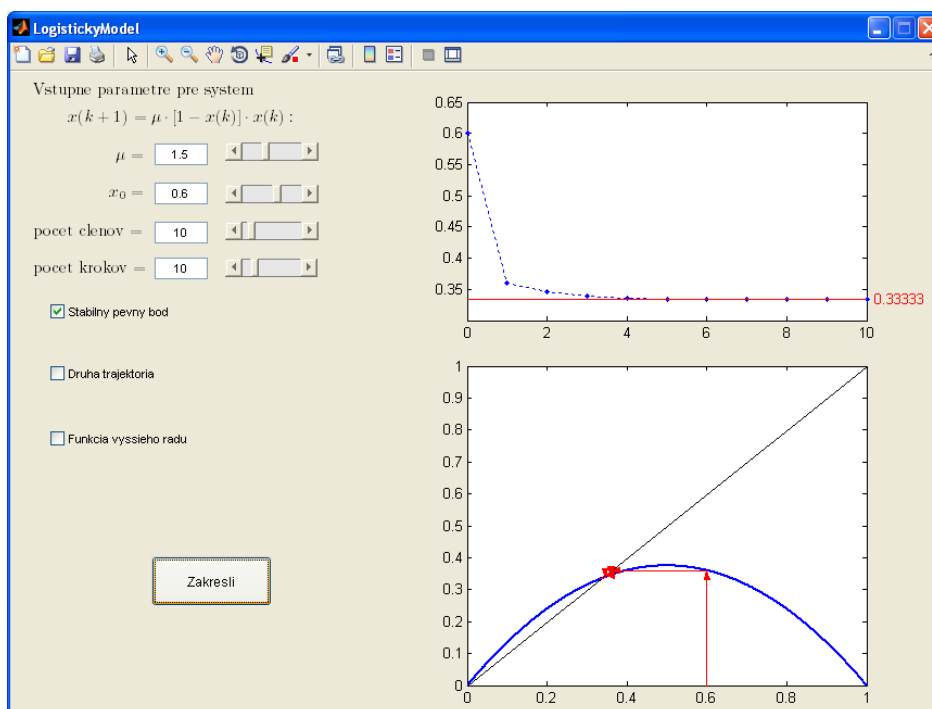
Tabuľka 1: Hodnoty parametrov: h_0 je prednastavená hodnota, h_{min} je minimálna a h_{max} maximálna možná hodnota

Prednastavené hodnoty, ako aj minimálne a maximálne možné hodnoty sú uvedené v Tabuľke 1. V prípade, že užívateľ zadá číselnú hodnotu mimo povoleného rozsahu, bude automaticky zvolená povolená hodnota najbližšia tej zadanej. Ak zadá nečíselnú hodnotu (napr. písmeno), parameter sa nastaví na predvolenú hodnotu.

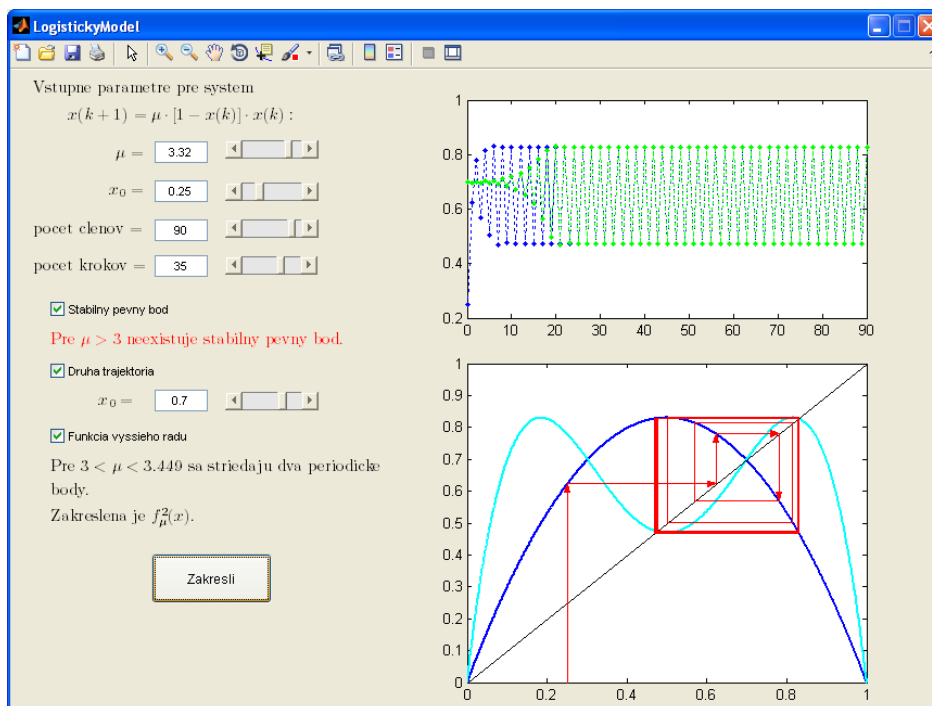
Nižšie si môžeme vybrať, či sa pri trajektórii zakreslí aj pevný bod (ak pre zvolenú hodnotu μ existuje), či sa zakreslí okrem pôvodnej trajektórie ešte druhá a či sa v pavučinovom modeli zobrazí aj funkcia vyššieho rádu f_μ^r , pričom r sa volí v závislosti od μ tak, aby sa demonštrovali 2-, 4- a 8-cykly spomínané v časti 1.3.4.

Vykreslenie sa spustí tlačidlom *Zakresli*. Výstup pri prednastavených hodnotách je na Obr. 16. Trajektória je vykreslená v horných osiach a keďže je zvolená možnosť *Stabilny pevny bod*, je tam červenou znázornený aj pevný bod $x_1 \approx 0,33333$. Do spodných osí sa zakresľuje pavučinový diagram, pričom prvých niekoľko iterácií sa vykresľuje postupne (použili sme príkaz *pause*, viď Príloha B, r. 276 a 285) a so šípkou v smere kreslenia, ostatné sa dokreslia naraz bez šípky.

Obr. 17 je príkladom výstupu programu pri zmenených hodnotách. Je síce zvolená možnosť *Stabilny pevny bod*, ale pre $\mu = 3,32$ stabilný pevný bod neexistuje, čo sa pre užívateľa vypíše červenou farbou. Taktiež je zakreslená aj druhá trajektória a obe



Obr. 16: Výstup programu pri prednastavených hodnotách.



Obr. 17: Výstup programu pri zmenených hodnotách.

ustália na dvoch stabilných periodických bodoch. Tie sú priesečníkmi funkcií $f_\mu^2(x)$ a $y = x$, ako to vidíme na pavučinovom diagrame.

2 Konzervatívne systémy s jedným stupňom voľnosti

V druhej kapitole sa zameriame na konzervatívne systémy s jedným stupňom voľnosti. Kým logistický model by dal zaradiť do biológie, tu použijeme, rovnako ako [1] a [2], peknú aplikáciu z fyziky na jednoduché mechanické systémy, kde sa zachováva celková mechanická energia, ktorá sa skladá z potenciálnej a kinetickej energie. Ide teda opäť o využitie matematiky na opísanie bežného javu z reálneho sveta, aj keď zjednodušeného. Podobne ako v kapitole 1, začneme definíciou pojmov.

2.1 Definícia pojmov

Pojmom dynamický systém budeme v tejto kapitole označovať vzťah daný diferenciálnou rovnicou

$$\dot{x} = f(x), \quad (11)$$

kde $x = x(t) \in \mathbb{R}^n$ a $f \in C^1$. Na rozdiel od prvej kapitoly pracujeme teraz so spojitým systémom, kde premenná t , predstavujúca čas, nadobúda ľubovoľné reálne hodnoty. Nasledujúce definície čerpáme z [2].

Definícia 2.1 (Riešenie a trajektória diferenciálnej rovnice). Riešením diferenciálnej rovnice (11) je funkcia $x(t)$, ktorá spĺňa (11).

Trajektóriou diferenciálnej rovnice (11) budeme nazývať množinu $\{x(t); t \in A\}$, kde $x(t)$ je riešením (11) a A je jeho interval existencie.

Definícia 2.2 (Integrál diferenciálnej rovnice). Integrálom diferenciálnej rovnice (11) nazveme funkciu $I \in C^1$, ktorá je konštantná na trajektóriach systému (11).

Poznámka: Ak $x(t)$ je riešením (11), dostávame

$$\frac{d}{dt}I[x(t)] = \frac{dI}{dx}[x(t)] \cdot \dot{x}(t) = \frac{dI}{dx}[x(t)] \cdot f[x(t)].$$

Keďže derivácia konštanty je nulová, I je integrálom (11) práve vtedy, keď

$$DI(x) f(x) = 0$$

pre všetky x , kde sú I aj f definované.

Po uvedení pomocných definícií môžeme definovať konzervatívny systém tak, ako to robí [1].

Definícia 2.3 (Konzervatívny systém s jedným stupňom voľnosti). *Pod konzervatívnym systémom s jedným stupňom voľnosti (skrátene budeme hovoriť konzervatívny systém) rozumieme systém opísaný diferenciálnou rovnicou*

$$\ddot{x} = f(x), \quad (12)$$

kde f je diferencovateľná funkcia definovaná na intervale $A \subset \mathbb{R}$. Vzťah (12) je ekvivalentný so systémom

$$\begin{cases} \dot{x} = y, \\ \dot{y} = f(x), \end{cases} \quad (x, y) \in A \times \mathbb{R}. \quad (13)$$

Definícia 2.4 (Pevný bod). *Majme systém daný*

$$\dot{x} = f(x),$$

kde $x \in \mathbb{R}^n$ a $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Pevným bodom tohto systému budeme nazývať bod $\hat{x} \in \mathbb{R}^n$ taký, že riešenie $x(t)$, pre ktoré platí $x(0) = \hat{x}$, splňa $x(t) \equiv \hat{x}$ pre všetky t . Bod \hat{x} je pevným bodom práve vtedy, keď

$$f(\hat{x}) = 0. \quad (14)$$

Definícia 2.5 (Klasifikácia pevných bodov). *Majme systém daný*

$$\dot{x} = Ax,$$

kde $x \in \mathbb{R}^2$ a A je regulárna matica s rozličnými vlastnými hodnotami. Potom nastane jeden z nasledujúcich prípadov:

- *vlastné hodnoty sú reálne a majú rovnaké znamienko - pevný bod je uzol,*
- *vlastné hodnoty sú reálne s rôznymi znamienkami - pevný bod je sedlo,*
- *vlastné hodnoty sú komplexne združené - pevný bod je fokus (nenulová reálna časť vlastných hodnôt) alebo centrum (rýdzo imaginárne vlastné hodnoty).*

Definícia 2.6 (Linearizácia systému). *Majme dynamický systém*

$$\begin{cases} \dot{x} = f(x, y) \\ \dot{y} = g(x, y). \end{cases}$$

Linearizáciou *tohto systému v bode* (\hat{x}, \hat{y}) *budeme označovať systém*

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x}(\hat{x}, \hat{y}) & \frac{\partial f}{\partial y}(\hat{x}, \hat{y}) \\ \frac{\partial g}{\partial x}(\hat{x}, \hat{y}) & \frac{\partial g}{\partial y}(\hat{x}, \hat{y}) \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}.$$

2.2 Zachovanie energie

V tejto časti budeme konzervatívne systémy po vzore [1] interpretovať na jednoduchom mechanickom systéme, kde platí zákon zachovania energie. V tejto interpretácii je v (13)

$x(t) \in A \subset \mathbb{R}$	horizontálna súradnica polohy objektu v čase t ,
$y(t) = \dot{x}(t)$	rýchlosť objektu v čase t ,
$\dot{y}(t) = \ddot{x}(t)$	zrýchlenie v čase t ,
$f(x)$	sila pôsobiaca na objekt.

Označme

$T(\dot{x}) = \frac{\dot{x}^2}{2} = \frac{y^2}{2}$	kinetickú energiu,
$U(x) = -\int_{x_0}^x f(\xi) d\xi$	potenciálnu energiu a
$E(x, \dot{x}) = U(x) + T(\dot{x})$	celkovú mechanickú energiu.

Zjavne $f(x) = -U'(x)$ a systém (13) je daný potenciálnou energiou U . Nakoľko platí

$$\begin{aligned} \frac{d}{dt} E[x(t), y(t)] &= \frac{d}{dt} (U[x(t)] + T[y(t)]) \\ &= \frac{d}{dt} \left(U[x(t)] + \frac{y^2(t)}{2} \right) \\ &= U'(x) y + y \dot{y} \\ &= -f(x) y + y f(x) = 0, \end{aligned}$$

celková mechanická energia E je integrálom (13).

2.2.1 Gulička v miske

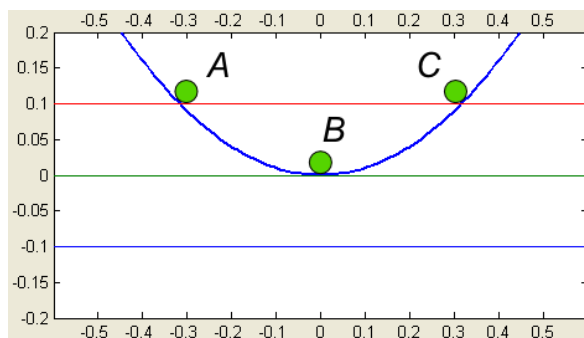
Pre lepšiu predstavu uvažujme nasledujúci príklad. Majme dokonale guľaté závažie jednotkovej hmotnosti a misku tvaru rotačného eliptického paraboloidu. Pri predpoklade, že závažie sa v miske pod vplyvom gravitácie kotúľa bez trenia a akéhokoľvek odporu, ide o konzervatívny systém. Ak guľičku umiestnime na dno misky, jeho potenciálna energia U bude nulová a ostane na tom istom mieste. Ide teda o pevný bod daného

systemu. Ak však umiestnime guľičku na iné miesto v miske, bude sa kotúľať tam a späť. Na začiatku bude mať nenulovú potenciálnu energiu $U = \gamma$, ktorá sa pri kotúľaní dole celá zmení na kinetickú energiu T a pri pohybe nahor znova na potenciálnu. Následne sa začne rovnaký proces v opačnom smere. Tento systém možno zapísať vzťahom

$$\ddot{x} = -2x, \quad (15)$$

alebo ekvivalentne

$$\begin{cases} \dot{x} = y, \\ \dot{y} = -2x. \end{cases} \quad (16)$$



Obr. 18: Jednoduchý mechanický konzervatívny systém.

Keďže na začiatku pokladáme guľičku do misky s nulovou rýchlosťou, platí $y(0) = 0$ a teda $T[y(0)] = 0$ a $U[x(0)] = \gamma$. Preto existuje trajektória len pre nezáporné hodnoty γ . Pre $\gamma = 0$ je guľička na dne misky (viď Obr. 18 bod B) a trajektóriou je len bod $(0,0)$. Ak však zvolíme γ kladné (Obr. 18 prípad A alebo C), guľička sa hýbe po parabole tam a späť. Platí pritom vzťah

$$\gamma = E(x, y) = T(y) + U(x) = \frac{1}{2}y^2 + U(x), \quad (17)$$

odkiaľ sa dá vyjadriť rýchlosť ako

$$y = \pm\sqrt{2[\gamma - U(x)]}. \quad (18)$$

Reálne riešenia existujú, ak $\gamma \geq U(x)$. Ďalej sa im budeme venovať v časti 2.3.

2.3 Fázový portrét

Dvojrozmerný dynamický systém sa často zakresľuje do fázového portréту, kde sú zakreslené trajektórie v rovine $x-y$. Pre systém (16) sú trajektórie opísané vzťahom

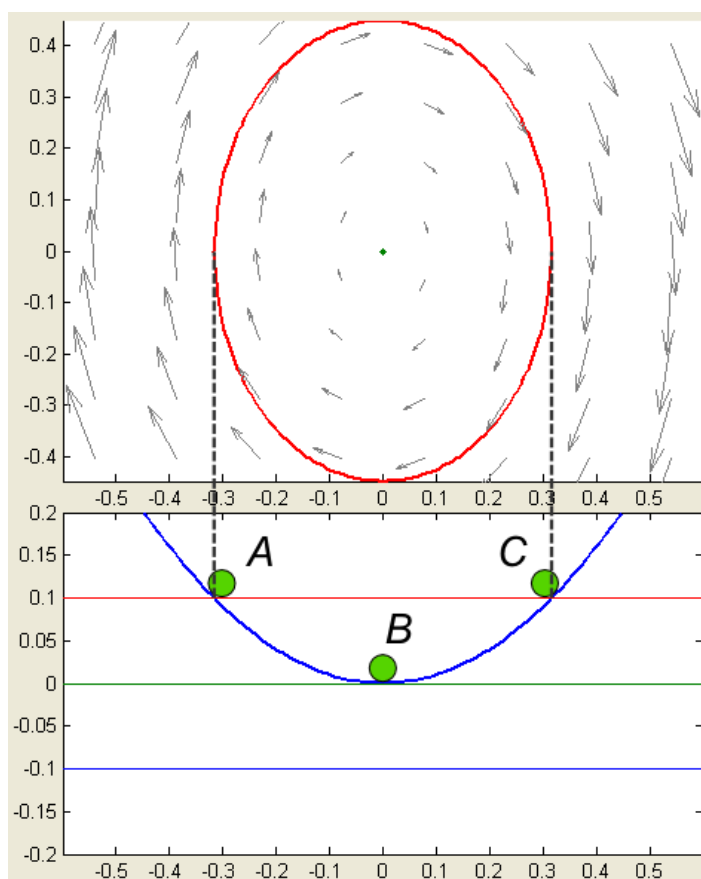
(18), ktorý po dosadení za $U(x)$ vyzerá nasledovne

$$y = \pm\sqrt{2(\gamma - x^2)}.$$

Pre dané γ existuje trajektória pre tie x , pre ktoré platí $x^2 \leq \gamma$. Na Obr. 19 je zakreslený fázový portrét pre (16). Na energetickej úrovni $\gamma_0 = -0,1$ neexistuje trajektória. Úrovni $\gamma_1 = 0$ (prípád b, guľička je na dne), zodpovedá bod $(0,0)$ a pre úroveň $\gamma_2 = 0,1$ existuje trajektória tvaru elipsy

$$x^2 + \frac{y^2}{2} = 0,1$$

(pevný bod je teda centrom).



Obr. 19: Fázový portrét pre jednoduchý mechanický konzervatívny systém.

Táto eliptická trajektória sa dá vysvetliť tak, že z počiatočného bodu A sa guľička kotúľa dole a jej rýchlosť y rastie, až kým prejde bodom B , kde má maximálnu rýchlosť a kinetickú energiu $T(y) = \gamma$. Odtiaľ sa guľička už pohybuje hore kopcom, čiže y klesá, až príde do bodu C , v ktorom zastane a má znova potenciálnu energiu $U(x) = \gamma$.

To zodpovedá hornej polovici elipsy. Tá spodná opisuje spätný pohyb, ktorý sa začne v momente, keď guľička dosiahne bod C .

Označme E_γ trajektóriu, pre ktorú platí $E(x, \dot{x}) = \gamma$. Uvedieme tvrdenie prebraté z [2].

Tvrdenie 2.7. *Pre systém (13) platia nasledujúce tvrdenia:*

1. *Trajektórie E_γ sú symetrické vzhľadom na x -ovú os, t.j.*

$$(x, y) \in E_\gamma \Rightarrow (x, -y) \in E_\gamma.$$

2. *Také y , že $(x, \pm y) \in E_\gamma$, existuje práve vtedy, keď $U(x) \leq \gamma$. Navyše platí*

$$v = 0 \Leftrightarrow U(x) = \gamma, \quad v \neq 0 \Leftrightarrow U(x) < \gamma.$$

3. *(\hat{x}, \hat{y}) je pevným bodom (13) práve vtedy, keď $\hat{y} = 0$ a \hat{x} je stacionárnym bodom U (t.j. $U'(\hat{x}) = 0$). (\hat{x}, \hat{y}) je sedlo, ak $U''(\hat{x}) < 0$ a centrum, ak $U''(\hat{x}) > 0$.*

Dôkaz. 1. Plynie z toho, že trajektórie sú definované pomocou y^2 ako

$$\gamma = \frac{1}{2}y^2 + U(x).$$

2. Toto tvrdenie sme už spomínali v texte. Vyplýva z (18).

3. Určite platí, že $\dot{x} = 0$ práve vtedy, keď $y = 0$ a $\dot{y} = 0$, keď $f(x) = -U(x) = 0$.

Linearizácia (13) v pevnom bode (\hat{x}, \hat{y}) je reprezentovaná maticou

$$\begin{pmatrix} 0 & 1 \\ f'(\hat{x}) & 0 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -U''(\hat{x}) & 0 \end{pmatrix}$$

s charakteristickou rovnicou

$$\lambda^2 + U''(\hat{x}) = 0.$$

Vlastné hodnoty $\lambda_{1,2} = \pm\sqrt{-U''(\hat{x})}$ sú reálne s rôznymi znamienkami pre $U''(\hat{x}) < 0$ a združené imaginárne pre $U''(\hat{x}) > 0$. To, že $(\hat{x}, 0)$ je naozaj centrum, je dôsledkom prvého tvrdenia, lebo podľa neho musia byť trajektórie v okolí tohto bodu uzavreté.

□

Aplikovaním časti 3 Tvrdenia 2.7 na systém (15) môžeme potvrdiť, že pevný bod $(0, 0)$ je centrum, lebo $U''(0) = 2 > 0$.

2.3.1 Postup zakresľovania

Podobným spôsobom, ako sme zakreslili fázový portrét pre systém (15), vieme ilustrovať aj zložitejšie systémy. Ukážme si postup zakresľovania fázového portrétu na systéme

$$\ddot{x} = f(x) = -x^3 + 5x^2 - 4x - 1,$$

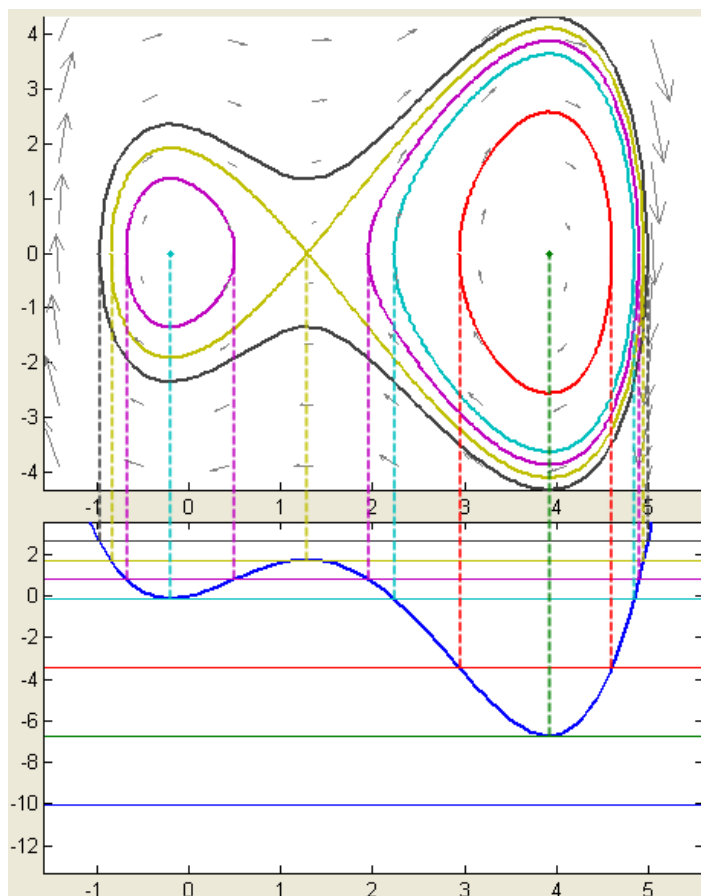
ktorý je ekvivalentný s

$$\begin{cases} \dot{x} = y, \\ \dot{y} = f(x) = -x^3 + 5x^2 - 4x - 1. \end{cases} \quad (19)$$

Najprv vyjadríme potenciálnu energiu

$$U(x) = - \int f(x) dx = \frac{x^4}{4} - \frac{5x^3}{3} + 2x^2 + x, \quad (20)$$

ktorá je zakreslená v spodnej časti Obr. 20.



Obr. 20: Fázový portrét pre mechanický konzervatívny systém (19).

Vidíme, že opäť možno použiť prirovnanie guľičky v miske z časti 2.2.1, len profil misky má tvar s dvoma údoliami a jedným kopcom medzi nimi. Systém má teda tri pevné body, ktorými sú dná údolí a vrchol kopca v strede. Úrovne γ volíme tak, aby sme pokryli "zlomové body", ktorými sú hodnoty funkcie U v stacionárnych bodoch, teda v koreňoch funkcie f :

$$\hat{x}_1 \approx -0,1987; \quad \hat{x}_2 \approx 1,2865; \quad \hat{x}_3 \approx 3,9122.$$

Príslušné hodnoty sú

$$U(\hat{x}_1) \approx -0,1063; \quad U(\hat{x}_2) \approx 1,7327; \quad U(\hat{x}_3) \approx -6,7098.$$

Platí $U(\hat{x}_3) < U(\hat{x}_1) < U(\hat{x}_2)$. Keďže nás zaujímajú aj iné trajektórie, zakreslíme aj úrovne γ medzi týmito hodnotami a aj jednu nad $U(\hat{x}_2)$ a jednu pod $U(\hat{x}_3)$. Hodnoty γ použité na Obr. 20 sú v Tabuľke 2.

γ_0	$\gamma_1 = U(\hat{x}_3)$	γ_2	$\gamma_3 = U(\hat{x}_1)$	γ_4	$\gamma_5 = U(\hat{x}_2)$	γ_6
-10,0115	-6.7098	-3.4080	-0.1063	0.8132	1.7327	2.6522

Tabuľka 2: Zvolené úrovne γ na Obr. 20

Keď máme zvolené hladiny γ , môžeme začať s kreslením, pričom sa budeme odvolávať na Obr. 20. Do grafu $U(x)$ ich zakresľujeme postupne od najnižšej. Pre danú γ podľa 2.7 existuje trajektória tam, kde je graf pod touto hladinou alebo zároveň s ňou. Vidíme, že pre γ_0 (modrá) je celý graf nad ňou, takže trajektória neexistuje. Pre $\gamma_1 = U(\hat{x}_3)$ sa U dotýka v bode \hat{x}_3 zelenej hladiny a inak je nad ňou. To znamená, že v bode $(\hat{x}_3, 0)$ je jednobodová trajektória a tento bod zaznačíme do fázového portréту. Pozrime sa ešte na γ_2 . Táto hladina (červená) pretína graf $U(x)$ v dvoch bodoch a medzi nimi je graf pod touto úrovnňou. Práve tam teda existuje trajektória, ktorú zakreslíme do portrétu podľa vzťahu (18). Pri ostatných hladinách postupujeme analogicky, až dostaneme fázový portrét z Obr. 20.

2.3.2 Inerpretácia

Charakter pevných bodov možno určiť podľa znamienka $U''(\hat{x})$ s použitím tvrdenia 2.7, alebo z fázového portrétu, odkiaľ vidno, že \hat{x}_1 a \hat{x}_3 sú cetrá a \hat{x}_2 je sedlo. Môžeme

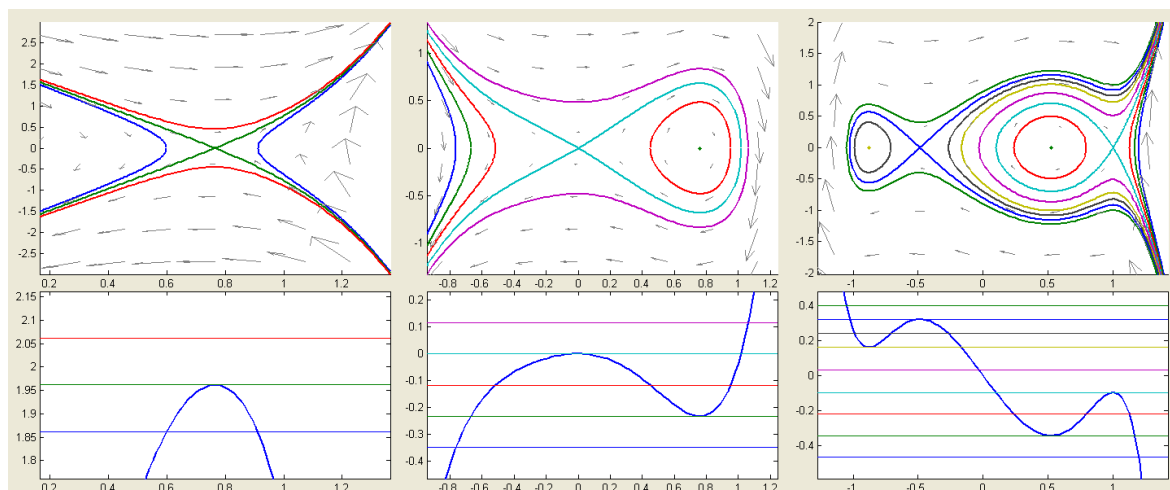
si všimnúť, že U má v \hat{x}_1 a \hat{x}_3 lokálne minimá, takže hodnota U'' tam bude skutočne kladná a v \hat{x}_2 je lokálne maximum, čo implikuje zápornú druhú deriváciu. Pre overenie

$$U''(\hat{x}_1) = -f'(\hat{x}_1) = 3x^2 - 10x + 4|_{x \approx -0,1063} \approx 5,0969;$$

$$U''(\hat{x}_2) \approx U''(1,2865) \approx -3,8998;$$

$$U''(\hat{x}_3) \approx U''(3,9122) \approx 10,7939.$$

Na základe skúmania systémov (16) a (19) sa môžeme domnievať, že pevný bod má charakter centra, ak v ňom má potenciálna energia U lokálne minimum. Ak je tam maximum, pevný bod je sedlo. To je konzistentné s 2.7, lebo znamienko U'' sa správa v extrémoch tak, ako sme spomínali vyššie. Interpretovať sa to dá tak, že v prvom prípade ide o údolie, v okolí ktorého sú eliptické trajektórie. Tie sú tam preto, lebo guľička zrýchľuje, keď sa približuje ku dnu, a potom pokračuje v pohybe rovnakým smerom a spomaľuje. Druhý prípad predstavuje kopec, takže guľička spomaľuje, keď sa blíži k vrcholu. Pokiaľ nemá dostatočnú energiu, na vrchol nedôjde a začne sa hýbať opačným smerom. Ak však má dosť energie, vrcholom prejde a začne zrýchľovať, nekoľko sa pohybuje dole kopcom. Môže ešte nastať tretia možnosť, že guľička má práve toľko energie, že vyjde na vrchol a tam zastane. Zhrnúť to možno tak, že údolie na grafe U implikuje centrum a kopec sedlo.



Obr. 21: Fázové portréty pre konzervatívne systémy $\ddot{x} = f(x) = 4x^7 - 3x^5 + x^2 + 6x - 5$, $\ddot{x} = f(x) = -4x^4 + x^2 + x$, $\ddot{x} = f(x) = 5x^6 - 2x^5 - x^4 + x^3 - 4x^2 + 1$.

Overme si to na Obr. 21, ktorý zobrazuje systémy

- $\ddot{x} = f(x) = 4x^7 - 3x^5 + x^2 + 6x - 5$,
- $\ddot{x} = f(x) = -4x^4 + x^2 + x$,
- $\ddot{x} = f(x) = 5x^6 - 2x^5 - x^4 + x^3 - 4x^2 + 1$.

Prvý systém má len jeden pevný bod, ktorým je sedlo v bode lokálneho maxima potenciálnej energie U . Druhý systém má jeden kopec a jedno údolie, ktorým zodpovedajú sedlo, resp. centrum. Tretí systém má najzložitejší graf U s dvomi údoliami a dvomi kopcami a na fázovom portréte vidíme dve centrá a dve sedlá. Všetky tri systémy z Obr. 21 teda potvrdili našu domnienku.

2.4 Program FazovyPortret

Druhým z dvojice programov, ktoré sú hlavným prínosom tejto práce, je program FazovyPortret. Podobne ako LogistickyModel, aj tento program vznikol v Matlabe a jeho účelom je grafická ilustrácia konzervatívnych systémov pomocou fázových portrétov. Môže byť použitý pri výučbe na demonštráciu kreslenia fázových portrétov a tiež si na ňom študenti môžu sami preskúmať trajektórie konzervatívnych systémov. Rovnako ako program z časti 1.4, aj FazovyPortret pozostáva z .fig súboru a .m súboru, ktoré majú rovnaké úlohy ako to bolo pri prvom programe.

2.4.1 Spustenie programu

Postup spustenia je analogický ako pri programe LogistickyModel, len na začiatku zadáme do príkazového okna príkaz

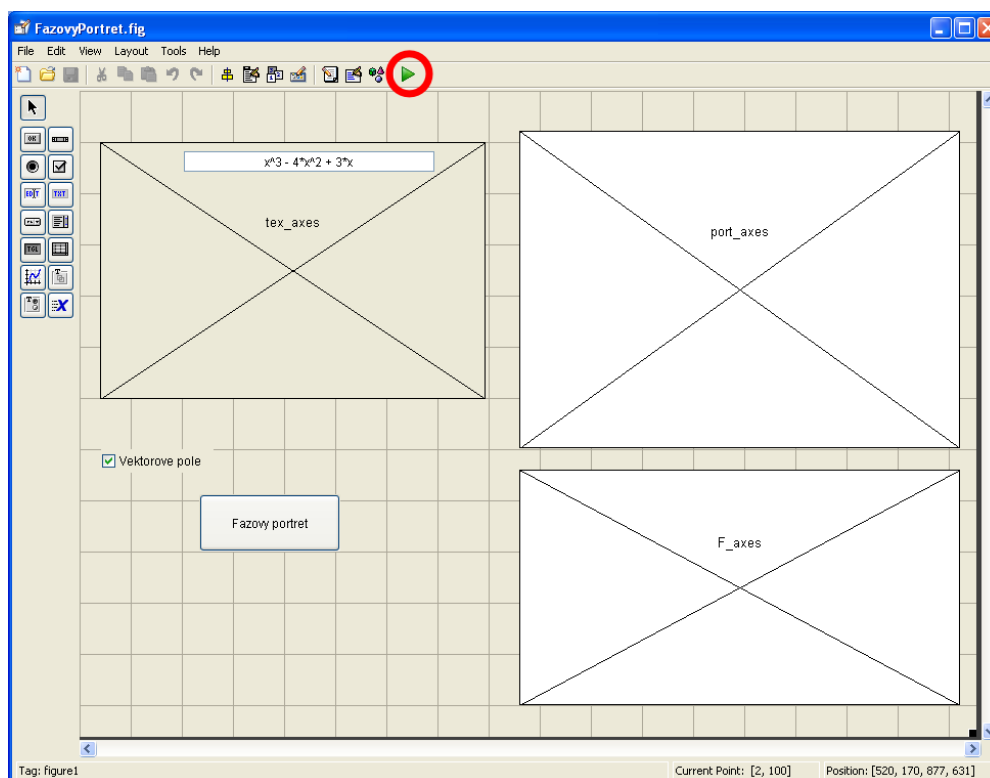
```
guide FazovyPortret,
```

ktorý otvorí súbor FazovyPortret.fig v prostredí GUIDE (Obr. 22).

Po kliknutí na zelenú šípku v hornom ovládacom paneli sa otvorí okno s užívateľským rozhraním, ktoré je zobrazené na Obr. 23.

2.4.2 Užívateľské rozhranie

Jediným vstupným údajom je funkcia $f(x)$, ktorej prednastavená hodnota je $\ddot{x} = f(x) = x^3 - 4x^2 + 3x$. Na nájdenie koreňov zadanej funkcie je použitá funkcia `solve` (Príloha C, riadok 51), ktorá spoľahlivo nájde všetky korene len pre polynómy,



Obr. 22: Súbor FazovyPortret.fig so všetkými objektami. Zelené tlačidlo na spustenie programu je v hornom ovládacom paneli.

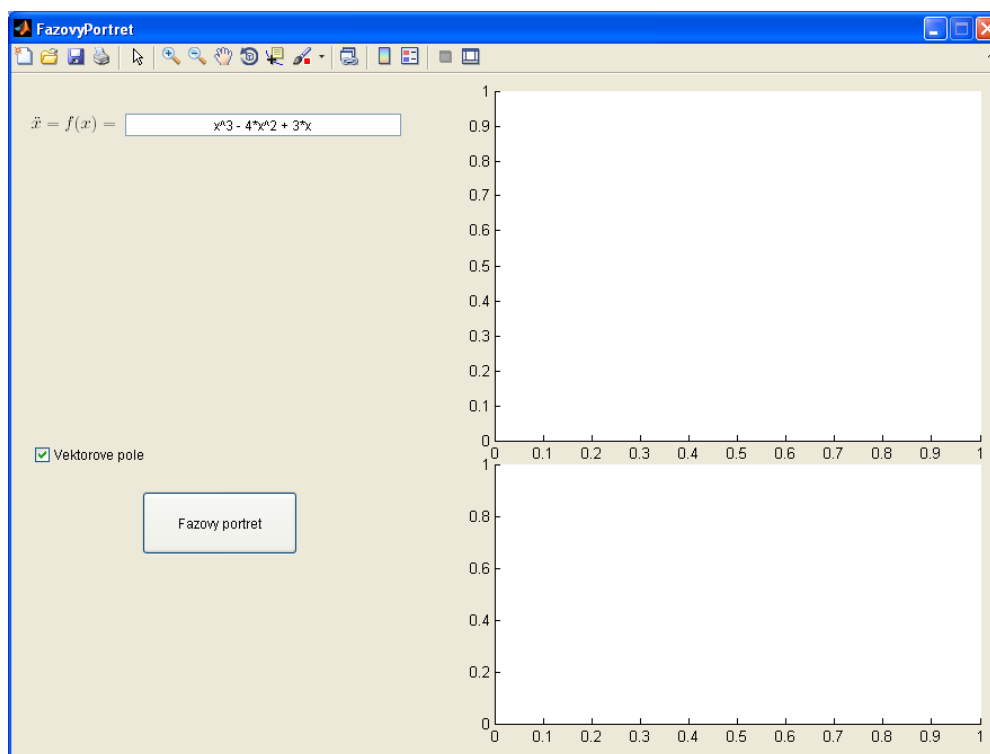
preto odporúčame pracovať len s takými funkciami. Okrem toho si môže užívateľ zvoliť, či chce zakresliť aj vektorové pole, ktoré znázorňuje smer trajektórií. Pri spustení je vybratá táto možnosť.

Kreslenie sa spustí kliknutím na tlačidlo *Fazovy portret* a výstup je na Obr. 24. Na obrazovke sa vypíše predpis pre $U(x)$, systém s pomocnou premennou y , aj predpis riešení. Na pravej strane v spodnej časti je zakreslená potenciálna energia $U(x)$ s automaticky zvolenými úrovňami γ . Tými sú hodnoty extrémov funkcie U , stredná hodnota medzi každými dvoma susednými extrémami a jedna hodnota pod najnižším a jedna nad najvyšším extrémom. Pre tieto hladiny sú v osiach nad tým zakreslené riešenia systému. Hladina γ a zodpovedajúce riešenie sú vždy zakreslené rovnakou farbou a farby sa striedajú pre väčšiu prehľadnosť.

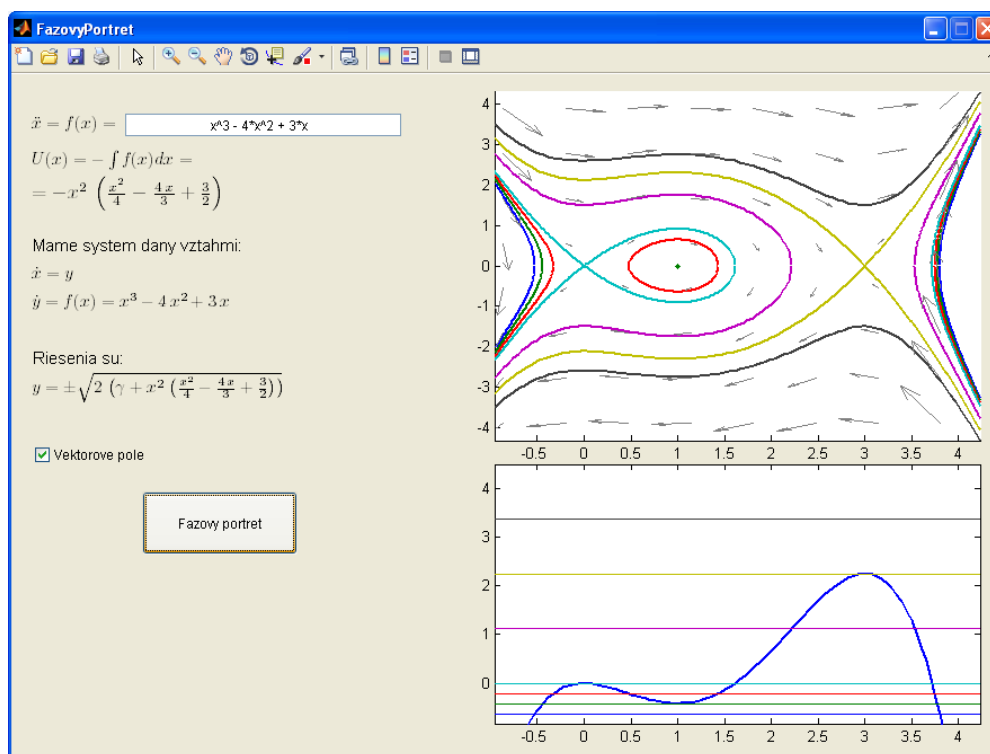
Na Obr. 25 je výstup programu FazovyPortret pre systém

$$\ddot{x} = f(x) = -2x^5 + 3x^4 - 7x^3 - 14x^2 + 15x + 5,$$

pričom nie je zvolená možnosť zakreslenia vektorového poľa. Fázový portrét tak pôsobí

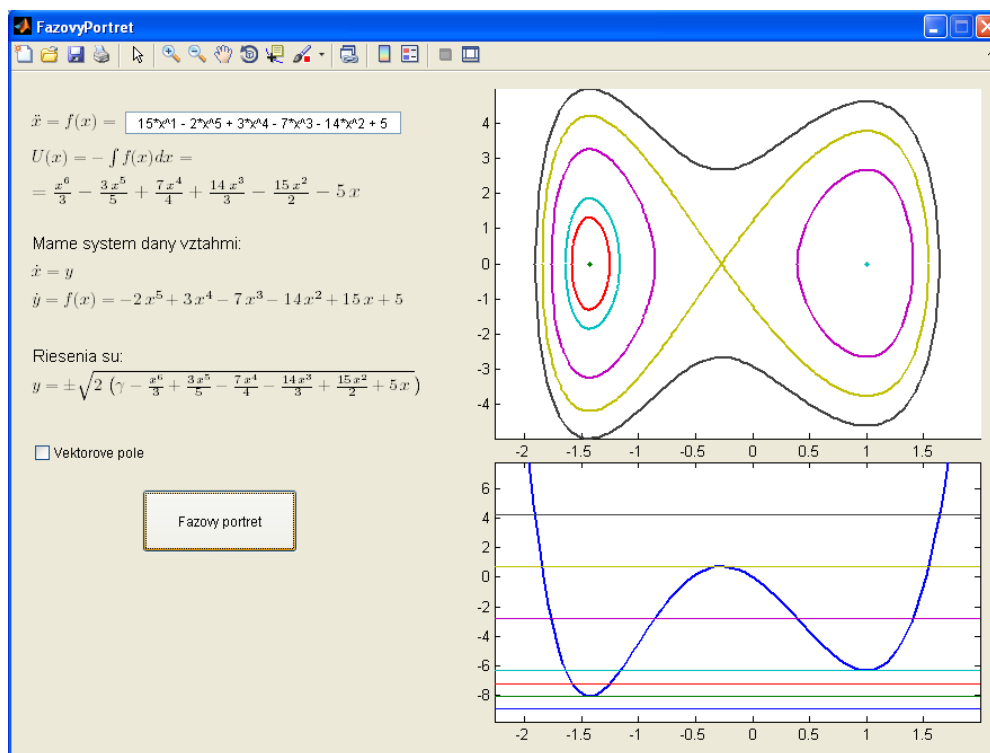


Obr. 23: Uživatľské rozhranie programu FazovyPortret po spustení s pôvodnými nastaveniami.



Obr. 24: Výstup programu pri prednastavených hodnotách.

prehľadnejšie, ale nie je v ňom informácia o smere trajektórií. Môžeme si všimnúť, že užívateľ nezadal funkciu f v zaužívanom tvare (členy polynómu nie sú zoradené od najvyššej mocniny), ale nižšie už je funkcia vypísaná v tomto tvare, čo automaticky dosiahneme použitím funkcie `latex` (Príloha C, riadok 89).



Obr. 25: Výstup programu pri zmenených hodnotách.

2.4.3 Nedostatok programu

Hlavným nedostatkom programu FazovyPortret je, že spoľahlivo funguje len pre polynomiálne funkcie. Ako sme už spomínali v časti 2.4.2, je to spôsobené použitím funkcie `solve` na hľadanie koreňov. Tento nedostatok ukážeme na príklade kyvadla, ktoré je tiež konzervatívnym systémom s jedným stupňom voľnosti. Takýto systém je daný vzťahom

$$\ddot{x} = -\sin(x), \quad (21)$$

kde x je uhol, ktorý kyvadlo zvierá so spodnou vertikálnou pozíciou. Je zrejmé, že x a $x + 2k\pi$ reprezentujú totožné body pre všetky celé k , preto sa zameriame len na interval $(-\pi, \pi]$. Korene funkcie $f(x) = -\sin(x)$ na danom intervale sú body 0 a π , ktoré predstavujú spodnú a vrchnú vertikálnu polohu (viď Obr. 26). Potenciálna

energia je daná ako

$$U(x) = - \int -\sin(x)dx = -\cos(x)$$

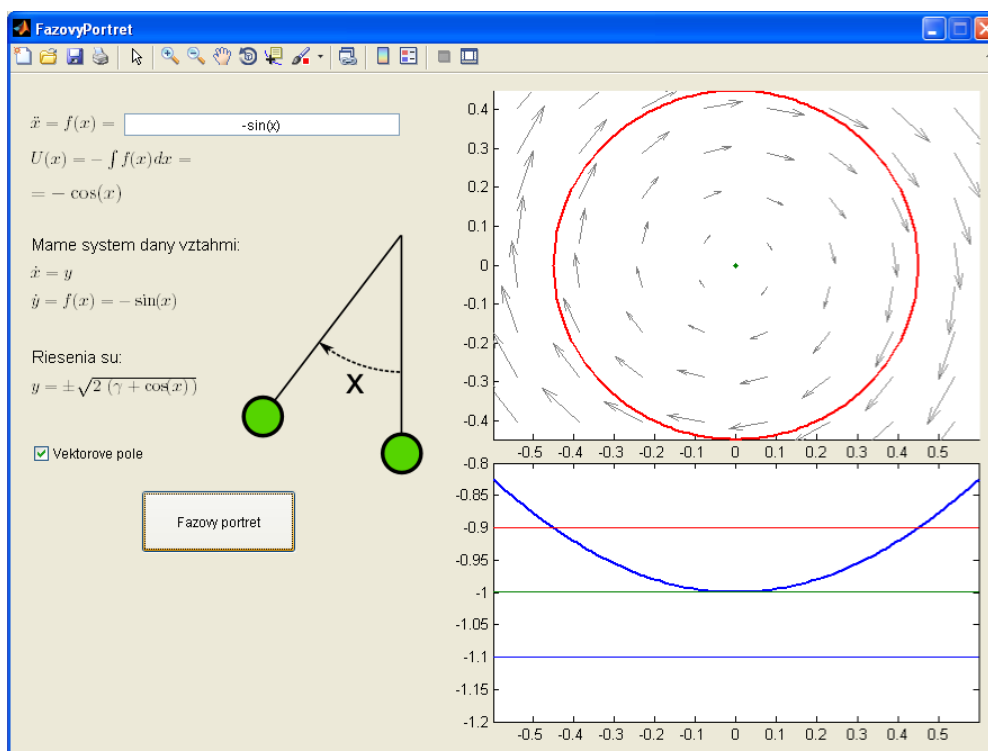
a platí

$$U''(0) = \cos(0) = 1 > 0,$$

$$U''(\pi) = \cos(\pi) = -1 < 0.$$

Z toho plynie, že spodná pozícia je centrum a horná sedlo.

Náš program však nájde len jediný koreň periodickej funkcie $f(x) = -\sin(x)$, ktorým je nula. Z toho dôvodu skúma len okolie tohto bodu a vzniknutý fázový portrét je síce v danej časti správny, ale je nekompletný, nakoľko nezobrazuje okolie bodu π . Výstup je zobrazený na Obr. 26.



Obr. 26: Výstup programu pre prípad kyvadla. Fázový portrét je nekompletný.

Záver

V našej práci sme sa zamerali na grafickú ilustráciu trajektórií dynamických systémov. Vybrali sme si logistický model populačnej dynamiky a konzervatívne systémy s jedným stupňom voľnosti. K obom sme vyvinuli programy, ktoré užívateľovi umožňujú skúmať dynamický systém pomocou interaktívnej ilustrácie jeho trajektórií. Tieto programy môžu oceniť študenti snažiaci sa porozumieť problematike dynamických systémov ako aj prednášajúci, ktorým ako didaktické pomôcky uľahčia vysvetľovanie.

Obe kapitoly majú podobnú štruktúru. Po uvedení čitateľa do problematiky v prvých dvoch častiach kapitoly v tretej časti vysvetľujeme, ako vznikajú a čo hovoria používané ilustrácie (v prvej kapitole pavučinový diagram, v druhej fázový portrét). Pri logistickom modeli sme sa venovali hlavne pevným a periodickým bodom a ich stabilite pre rôzne hodnoty parametrov a spomenuli sme aj chaotické vlastnosti modelu pri hodnotách parametra μ blízkyh hraničnej hodnote 4. Pri konzervatívnych systémoch nás zaujímal najmä charakter pevných bodov, priebeh trajektórií a dôraz sme kládli na vysvetlenie zakresľovania fázových portrétov. V poslednej časti kapitoly predstavujeme príslušný program a na vzorových príkladoch ukážeme, aké možnosti ponúka. Pri programe *FazovyPortret* v druhej kapitole na konci spomíname aj nedostatok programu, ktorým je obmedzenie na polynomiálne funkcie.

Na koniec nám ostáva len vyjadriť želanie, že naša práca napomôže záujemcom k lepšiemu porozumeniu dynamických systémov.

Zoznam použitej literatúry

- [1] Arnold, V. I.: *Ordinary Differential Equations, translated from the Russian by Richard A. Silverman*, The MIT Press, Cambridge a Londýn, 1973
- [2] Brunovský, P.: *Diferenčné a diferenciálne rovnice*, učebné texty, dostupné na internete (10.11.2011):
http://www.iam.fmph.uniba.sk/institute/brunovsky/ddrtext_09-10.pdf
- [3] Hale, J., Koçak, H.: *Dynamics and Bifurcations*, Springer-Verlag, New York, 1991
- [4] May, R. M.: *Simple Mathematical Models with Very Complicated Dynamics*, Nature, Vol. 261, (1976), strana 459, dostupné na internete (4.12.2011):
http://lpce.cnrs-orleans.fr/ddwit/enseignement/logistic_May_nature.pdf
- [5] Kollár, M.: *Matematická analýza (1)*, prednášky, FMFI UK, Bratislava, 2009
- [6] Pareek, N.K., Patidar, V., Sud, K.K.: *Image encryption using chaotic logisticmap*, Image and Vision Computing, Vol. 24, (2006), strana 926
- [7] Phatak, S.C., Rao, S.S.: *Logistic map: A possible random-number generator*, Physical Review E, Vol. 51, (1995), strana 3670

Príloha A - Lagrangeova veta o strednej hodnote

Nasledujúce dve pomocné sme prebrali z [5]. Rollovu vetu využijeme v dôkaze Lagrangeovej vety.

Veta A.0.8 (Rollova veta o strednej hodnote). *Nech $f : \langle a, b \rangle \rightarrow \mathbb{R}$ je spojitá a diferencovateľná. Nech navyše platí $f(a) = f(b)$. Potom existuje také $c \in (a, b)$, že $f'(c) = 0$.*

Dôkaz. Ak je f konštantná, platí

$$f'(c) = 0 \quad \forall c \in (a, b).$$

Pokiaľ f nie je konštantná, zo spojitosti vyplýva, že existuje $c \in (a, b)$ tak, že

$$f(c) = \max_{x \in (a, b)} f(x)$$

a z diferencovateľnosti plynie $f'(c) = 0$. □

Veta A.0.9 (Lagrangeova veta o strednej hodnote). *Nech $f : \langle a, b \rangle \rightarrow \mathbb{R}$ je spojitá a diferencovateľná. Potom existuje také $c \in (a, b)$, že platí*

$$f(b) - f(a) = f'(c)(b - a).$$

Dôkaz. Definujme

$$g(x) = f(x) - \frac{f(b) - f(a)}{b - a}(x - a),$$

ktorá je rovnako ako f spojitá a diferencovateľná. Taktiež platí $g(a) = f(a)$ a

$$g(b) = f(b) - \frac{f(b) - f(a)}{b - a}(b - a) = f(a).$$

Sú teda splnené predpoklady Rollovej vety a existuje také $c \in (a, b)$, že $g'(c) = 0$.

Dostávame

$$0 = g'(c) = f'(c) - \frac{f(b) - f(a)}{b - a}.$$

□

Príloha B - Zdrojový kód programu LogistickyModel

```

1 function varargout = LogistickyModel(varargin)
2 gui_Singleton = 1;
3 gui_State = struct('gui_Name',       mfilename, ...
4                   'gui_Singleton',  gui_Singleton, ...
5                   'gui_OpeningFcn', @LogistickyModel_OpeningFcn, ...
6                   'gui_OutputFcn',  @LogistickyModel_OutputFcn, ...
7                   'gui_LayoutFcn',  [], ...
8                   'gui_Callback',   []);
9 if nargin && ischar(varargin{1})
10     gui_State.gui_Callback = str2func(varargin{1});
11 end
12
13 if nargout
14     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
15 else
16     gui_mainfcn(gui_State, varargin{:});
17 end
18
19 function LogistickyModel_OpeningFcn(hObject, eventdata, handles, varargin)
20 handles.output = hObject;
21
22 guidata(hObject, handles);
23 set(hObject, 'toolbar', 'figure'); %prida klasicke menu
24
25 function varargout = LogistickyModel_OutputFcn(hObject, eventdata, handles)
26 varargout{1} = handles.output;
27 set(handles.check_stabilny, 'Value', 1);
28 global stac_bod;
29 stac_bod = 1;
30 global druha;
31 druha = 0;
32 global vyssia;
33 vyssia = 0;
34
35 axes(handles.tex_axes);
36 text('Interpreter','latex','String','Vstupne parametre pre system','Position',[0 1],...
37      'FontSize',11);
38 text('Interpreter','latex','String','$x(k+1) = \mu \cdot [1-x(k)] \cdot x(k):$',...
39      'Position',[0.1 .92], 'FontSize',11);
40 text('Interpreter','latex','String','$\mu = $','Position',[.24 .8], 'FontSize',11);
41 text('Interpreter','latex','String','$x_0 = $','Position',[.22 .69], 'FontSize',11);
42 text('Interpreter','latex','String','pocet clenov = ','Position',[0 .58], 'FontSize',11);
43 text('Interpreter','latex','String','pocet krokov = ','Position',[0 .47], 'FontSize',11);

```

```
44
45 function edit_mi_Callback(hObject, eventdata, handles)
46 sliderValue = get(handles.edit_mi, 'String');
47 sliderValue = str2num(sliderValue);
48 if (isempty(sliderValue))
49     set(handles.slider_mi, 'Value', 1.5);
50     set(handles.edit_mi, 'String', '1.5');
51 elseif (sliderValue < 0)
52     set(handles.slider_mi, 'Value', 0);
53     set(handles.edit_mi, 'String', '0');
54 elseif (sliderValue > 4)
55     set(handles.slider_mi, 'Value', 4);
56     set(handles.edit_mi, 'String', '4');
57 else
58     set(handles.slider_mi, 'Value', sliderValue);
59 end
60
61 function edit_mi_CreateFcn(hObject, eventdata, handles)
62 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
63     set(hObject, 'BackgroundColor', 'white');
64 end
65
66 function edit_x0_Callback(hObject, eventdata, handles)
67 sliderValue = get(handles.edit_x0, 'String');
68 sliderValue = str2num(sliderValue);
69 if (isempty(sliderValue))
70     set(handles.slider_x0, 'Value', 0.6);
71     set(handles.edit_x0, 'String', '0.6');
72 elseif(sliderValue < 0)
73     set(handles.slider_x0, 'Value', 0);
74     set(handles.edit_x0, 'String', '0');
75 elseif (sliderValue > 1)
76     set(handles.slider_x0, 'Value', 1);
77     set(handles.edit_x0, 'String', '1');
78 else
79     set(handles.slider_x0, 'Value', sliderValue);
80 end
81
82 function edit_x0_CreateFcn(hObject, eventdata, handles)
83 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
84     set(hObject, 'BackgroundColor', 'white');
85 end
86
87 function edit_iter_Callback(hObject, eventdata, handles)
88 sliderValue = get(handles.edit_iter, 'String');
89 sliderValue = str2num(sliderValue);
90 if (isempty(sliderValue))
91     set(handles.slider_iter, 'Value', 10);
```



```
92     set(handles.edit_iter, 'String', '10');
93 elseif(sliderValue < 2)
94     set(handles.slider_iter, 'Value', 2);
95     set(handles.edit_iter, 'String', '2');
96 elseif (sliderValue > 100)
97     set(handles.slider_iter, 'Value', 100);
98     set(handles.edit_iter, 'String', '100');
99 else
100    set(handles.slider_iter, 'Value', sliderValue);
101 end
102
103 function edit_iter_CreateFcn(hObject, eventdata, handles)
104 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUicontrolBackgroundColor'))
105     set(hObject, 'BackgroundColor', 'white');
106 end
107
108 function pushbutton1_Callback(hObject, eventdata, handles)
109 w = get(handles.edit_mi, 'String');
110 w = str2num(w);
111 x0 = get(handles.edit_x0, 'String');
112 x0 = str2num(x0);
113 iter = get(handles.edit_iter, 'String');
114 iter = str2num(iter);
115 krok = get(handles.edit_krok, 'String');
116 krok = str2num(krok);
117
118 y = zeros(iter+1,1);
119 y(1) = x0;
120 for i=1:iter
121     y(i+1) = w*(1-y(i))*y(i);
122 end
123 x = 0:1:iter;
124 axes(handles.axes1);
125 set(gca, 'xLim', [0 iter]);
126 plot(handles.axes1, x, y, ':.')
127
128 global druha;
129 if( druha==1 )
130     y0 = get(handles.edit_druha, 'String');
131     y0 = str2num(y0);
132     y = zeros(iter+1,1);
133     y(1) = y0;
134     for i=1:iter
135         y(i+1) = w*(1-y(i))*y(i);
136     end
137     x = 0:1:iter;
138     hold(handles.axes1);
139     plot(handles.axes1, x, y, ':.', 'Color', 'green');
```

```

140     hold off;
141 end
142
143 global stac_bod;
144 if(stac_bod==1)
145     axes(handles.tex_axes);
146     text('Interpreter','latex','String','$\hspace{80 mm}$','Position',[.05 .25],...
147         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
148     if(w>3)
149         text('Interpreter','latex','String','Pre  $\mu > 3$  neexistuje stabilny pevny bod.',...
150             'Position',[.05 .25],'FontSize',11,'color','red');
151     else
152         if (w < 1 || x0==0 || x0==1)
153             y = 0;
154         elseif (w ≥ 1 && w ≤ 3)
155             y = 1-1/w;
156         else
157             y = NaN;
158         end
159         axes(handles.axes1);
160         line([0,iter],[y,y],'Color','red');
161         text(1.02*iter,y,num2str(y),'Color','red');
162         set(gca,'xLim',[0 iter]);
163     end
164 else
165     axes(handles.tex_axes);
166     text('Interpreter','latex','String','$\hspace{80 mm}$','Position',[.05 .25],...
167         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
168 end
169 axes(handles.axes1);
170 set(gca,'xLim',[0 iter]);
171
172 x = 0:.001:1;
173 y = w*(1-x).*x;
174 plot(handles.axes2,x,y,'LineWidth',1.5)
175
176 global vyssia;
177 if( vyssia ==1 )
178     axes(handles.tex_axes);
179     text('Interpreter','latex','String','$\hspace{85 mm}$','Position',[.05 -.13],...
180         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
181     text('Interpreter','latex','String','$\hspace{50 mm}$','Position',[.05 -.2],...
182         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
183     text('Interpreter','latex','String','$\hspace{50 mm}$','Position',[.05 -.28],...
184         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
185     if( w ≤ 3 )
186         text('Interpreter','latex','String',...
187             'Pre  $\mu \leq 3$  existuje jediny stabilny pevny bod.',...

```

```

188         'Position',[.05 -.13], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
189     rad = 0;
190     elseif(w > 3 && w < 3.449)
191         text('Interpreter','latex','String',...
192             'Pre $3 < \mu < 3.449$ sa striedaju dva periodicke',...
193             'Position',[.05 -.13], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
194     text('Interpreter','latex','String','body.','Position',[.05 -.2],...
195         'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
196     text('Interpreter','latex','String','Zakreslena je $f_{\mu}^2(x)$','Position',...
197         [.05 -.28], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
198     rad = 1;
199     elseif(w ≥ 3.449 && w < 3.544 )
200     text('Interpreter','latex','String',...
201         'Pre $3.449 < \mu < 3.544$ sa striedaju styri periodicke',...
202         'Position',[.05 -.13], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
203     text('Interpreter','latex','String','body.','Position',[.05 -.2],...
204         'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
205     text('Interpreter','latex','String','Zakreslena je $f_{\mu}^4(x)$',...
206         'Position',[.05 -.28], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
207     rad = 2;
208     elseif(w ≥ 3.544 && w < 3.564 )
209     text('Interpreter','latex','String',...
210         'Pre $3.544 < \mu < 3.564$ sa strieda osem periodickych',...
211         'Position',[.05 -.13], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
212     text('Interpreter','latex','String','stabilnych bodov.','Position',[.05 -.2],...
213         'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
214     text('Interpreter','latex','String','Zakreslena je $f_{\mu}^8(x)$',...
215         'Position',[.05 -.28], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
216     rad = 3;
217     else
218     text('Interpreter','latex','String',...
219         'Pre $\mu > 3.570$ je dynamika systemu komplikovana.',...
220         'Position',[.05 -.13], 'FontSize',11, 'BackgroundColor',[.925 .914 .847]);
221     rad = 100;
222     end
223     if(rad > 0 && rad < 4)
224         for i=1:(2^rad-1)
225             y = w*(1-y).*y;
226         end
227         hold(handles.axes2);
228         plot(handles.axes2,x,y, 'Color','c','LineWidth',1.5);
229         axes(handles.axes2)
230         hold off;
231     end
232     else
233         axes(handles.tex_axes);
234         text('Interpreter','latex','String','$\hspace{80 mm}$','Position',[.05 -.13],...
235             'FontSize',11, 'BackgroundColor',[.925 .914 .847]);

```

```
236     text('Interpreter','latex','String','$\hspace{50 mm}$','Position',[.05 -.2],...
237         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
238     text('Interpreter','latex','String','$\hspace{50 mm}$','Position',[.05 -.28],...
239         'FontSize',11,'BackgroundColor',[.925 .914 .847]);
240 end
241
242 axes(handles.axes2)
243 line([0,1],[0,1],'Color','black')
244 arrowline([x0,x0],[0,w*(1-x0)*x0],'Color','red')
245 x = [x0,w*(1-x0)*x0];
246 for i=1:krok
247     if( abs(x(1)-x(2)) > .03 && i<5)
248         pause(.1);
249     end
250     if(i<3)
251         arrowline([x(1),x(2)],[x(2),x(2)],'Color','red')
252     else
253         line([x(1),x(2)],[x(2),x(2)],'Color','red')
254     end
255     x = [x(2),w*(1-x(2))*x(2)];
256     if( abs(x(1)-x(2)) > .03 && i<5 )
257         pause(.2);
258     end
259     if(i<3)
260         arrowline([x(1),x(1)],[x(1),x(2)],'Color','red')
261     else
262         line([x(1),x(1)],[x(1),x(2)],'Color','red')
263     end
264 end
265 axes(handles.axes2);
266 axis([0 1 0 1]);
267
268 function slider_mi_Callback(hObject, eventdata, handles)
269 sliderValue = get(handles.slider_mi,'Value');
270 sliderValue=round(sliderValue*10)/10;
271 set(handles.edit_mi,'String', num2str(sliderValue));
272 guidata(hObject, handles);
273
274 function slider_mi_CreateFcn(hObject, eventdata, handles)
275 if isequal(get(hObject,'BackgroundColor'),get(0,'defaultUicontrolBackgroundColor'))
276     set(hObject,'BackgroundColor',[.9 .9 .9]);
277 end
278
279 function slider_x0_Callback(hObject, eventdata, handles)
280 sliderValue = get(handles.slider_x0,'Value');
281 sliderValue=round(sliderValue*10)/10;
282 set(handles.edit_x0,'String', num2str(sliderValue));
283 guidata(hObject, handles);
```

```
284
285 function slider_x0_CreateFcn(hObject, eventdata, handles)
286 if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
287     set(hObject,'BackgroundColor',[.9 .9 .9]);
288 end
289
290 function slider_iter_Callback(hObject, eventdata, handles)
291 sliderValue = get(handles.slider_iter,'Value');
292 sliderValue=round(sliderValue);
293 set(handles.edit_iter,'String', num2str(sliderValue));
294 guidata(hObject, handles);
295
296 function slider_iter_CreateFcn(hObject, eventdata, handles)
297 if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
298     set(hObject,'BackgroundColor',[.9 .9 .9]);
299 end
300
301 function check_stabilny_Callback(hObject, eventdata, handles)
302 checkboxStatus = get(handles.check_stabilny,'Value');
303 global stac_bod;
304 if(checkboxStatus)
305     stac_bod = 1;
306 else
307     stac_bod = 0;
308 end
309
310 function edit_krok_Callback(hObject, eventdata, handles)
311 sliderValue = get(handles.edit_krok,'String');
312 sliderValue = str2num(sliderValue);
313 if (isempty(sliderValue))
314     set(handles.slider_krok,'Value',10);
315     set(handles.edit_krok,'String','10');
316 elseif(sliderValue < 2)
317     set(handles.slider_krok,'Value',2);
318     set(handles.edit_krok,'String','2');
319 elseif (sliderValue > 50)
320     set(handles.slider_krok,'Value',50);
321     set(handles.edit_krok,'String','50');
322 else
323     set(handles.slider_krok,'Value',sliderValue);
324 end
325
326 function edit_krok_CreateFcn(hObject, eventdata, handles)
327 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUiControlBackgroundColor'))
328     set(hObject,'BackgroundColor','white');
329 end
330
331 function slider_krok_Callback(hObject, eventdata, handles)
```

```
332 sliderValue = get(handles.slider_krok, 'Value');
333 sliderValue=round(sliderValue);
334 set(handles.edit_krok, 'String', num2str(sliderValue));
335 guidata(hObject, handles);
336
337 function slider_krok_CreateFcn(hObject, eventdata, handles)
338 if isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
339     set(hObject, 'BackgroundColor', [.9 .9 .9]);
340 end
341
342 function check_druha_Callback(hObject, eventdata, handles)
343 checkboxStatus = get(handles.check_druha, 'Value');
344 global druha;
345 if(checkboxStatus)
346     druha = 1;
347     axes(handles.tex_axes);
348     text('Interpreter', 'latex', 'String', '$x_0 = $', 'Position', [.19 .06], 'FontSize', 11);
349
350     set(handles.edit_druha, 'Visible', 'on');
351     set(handles.slider_druha, 'Visible', 'on');
352 else
353     druha = 0;
354     axes(handles.tex_axes);
355     text('Interpreter', 'latex', 'String', '$\hspace{7 mm}$', 'Position', [.19 .06], ...
356         'FontSize', 11, 'BackgroundColor', [.925 .914 .847]);
357     set(handles.edit_druha, 'Visible', 'off');
358     set(handles.slider_druha, 'Visible', 'off');
359 end
360
361 function edit_druha_Callback(hObject, eventdata, handles)
362 sliderValue = get(handles.edit_druha, 'String');
363 sliderValue = str2num(sliderValue);
364 if (isempty(sliderValue))
365     set(handles.slider_druha, 'Value', 0.7);
366     set(handles.edit_druha, 'String', '0.7');
367 elseif(sliderValue < 0)
368     set(handles.slider_druha, 'Value', 0);
369     set(handles.edit_druha, 'String', '0');
370 elseif (sliderValue > 1)
371     set(handles.slider_druha, 'Value', 1);
372     set(handles.edit_druha, 'String', '1');
373 else
374     set(handles.slider_druha, 'Value', sliderValue);
375 end
376
377 function edit_druha_CreateFcn(hObject, eventdata, handles)
378 if ispc && isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
379     set(hObject, 'BackgroundColor', 'white');
```

```
380 end
381
382 function slider_druha_Callback(hObject, eventdata, handles)
383 sliderValue = get(handles.slider_druha, 'Value');
384 sliderValue=round(sliderValue*10)/10;
385 set(handles.edit_druha, 'String', num2str(sliderValue));
386 guidata(hObject, handles);
387
388 function slider_druha_CreateFcn(hObject, eventdata, handles)
389 if isequal(get(hObject, 'BackgroundColor'), get(0, 'defaultUiControlBackgroundColor'))
390     set(hObject, 'BackgroundColor', [.9 .9 .9]);
391 end
392
393 function check_vyssia_Callback(hObject, eventdata, handles)
394 checkboxStatus = get(handles.check_vyssia, 'Value');
395 global vyssia;
396 if(checkboxStatus)
397     vyssia = 1;
398 else
399     vyssia = 0;
400 end
```

Príloha C - Zdrojový kód programu FazvyPortret

```

1 function varargout = FazovyPortret(varargin)
2 gui_Singleton = 1;
3 gui_State = struct('gui_Name',       mfilename, ...
4                   'gui_Singleton',  gui_Singleton, ...
5                   'gui_OpeningFcn', @FazovyPortret_OpeningFcn, ...
6                   'gui_OutputFcn',  @FazovyPortret_OutputFcn, ...
7                   'gui_LayoutFcn',  [], ...
8                   'gui_Callback',   []);
9 if nargin && ischar(varargin{1})
10     gui_State.gui_Callback = str2func(varargin{1});
11 end
12 if nargin
13     [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
14 else
15     gui_mainfcn(gui_State, varargin{:});
16 end
17
18 function FazovyPortret_OpeningFcn(hObject, eventdata, handles, varargin)
19 handles.output = hObject;
20 guidata(hObject, handles);
21 set(hObject, 'toolbar', 'figure'); %prida klasicke menu
22
23 function varargout = FazovyPortret_OutputFcn(hObject, eventdata, handles)
24 varargout{1} = handles.output;
25 axes(handles.tex_axes);
26 text('Interpreter','latex','String','$\ddot{x} = f(x)=$','Position',[0 .93],'FontSize',11);
27 global pole;
28 pole = 1;
29
30 function f_edit_Callback(~, eventdata, handles)
31
32 function f_edit_CreateFcn(hObject, eventdata, handles)
33 if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
34     set(hObject,'BackgroundColor','white');
35 end
36
37 function port_button_Callback(hObject, eventdata, handles)
38 eps=.0001;
39 syms x;
40 global f;
41 f=char(get(handles.f_edit,'String'));
42 pomo = double( solve(f) );
43 ext=[];

```



```

44 for k=1:length(pomo)
45     if( abs( imag( pomo(k) ) ) < eps )
46         pomo(k) = real( pomo(k) );
47     end
48     if( isreal(pomo(k)) )
49         ext = [ext;pomo(k)];
50     end
51 end
52 ext = unique(ext);
53
54 pocMex = length(ext);    %pocet korenov f(x), moznych extremov U(x)
55 if(pocMex==0)
56     mini = -.5;
57     maxi = .5;
58 elseif(pocMex==1)
59     mini = ext - .5;
60     maxi = ext + .5;
61 else
62     mini = min(ext);
63     maxi = max(ext);
64 end
65
66 U = -int(f,x);    %najde primitivnu F k f, U = -F
67 axes(handles.tex_axes);
68 cla(handles.tex_axes);
69 text('Interpreter','latex','String','$\ddot{x} = f(x) = $','Position',[0 .93],'FontSize',11);
70 text('Interpreter','latex','String',['$U(x)=-\int f(x)dx=$'],'Position',[0 .8],'FontSize',11);
71 text('Interpreter','latex','String',['$ = ' latex(simple(U)) '$'],...
72     'Position',[0 .68],'FontSize',13);
73
74 text('String','Mame system dany vztahmi:','Position',[0 .5],'FontSize',11);
75 text('Interpreter','latex','String','$\dot{x} = y $','Position',[0 .4],'FontSize',11);
76 text('Interpreter','latex','String',['$\dot{y} = f(x)= ' latex(sym(f)) '$'],...
77     'Position',[0 .3],'FontSize',11);
78 text('String','Riesenia su:','Position',[0 .1],'FontSize',11);
79 text('Interpreter','latex','String',...
80     ['$y=\pm\sqrt{2},\left(\gamma-\left(\text{latex}(U) \backslash,\right)\right)$'],...
81     'Position',[0 0],'FontSize',11);
82 if(pocMex==0)
83     c = [-1; 0; 1];
84 else
85     hodn = double( subs(U,x,ext) ); % vycisli U(x) pre korene f(x)=0
86
87     hodn = unique(hodn);
88     pocex = length(hodn);
89     if(pocex == 0)
90         c = [0];
91     elseif(pocex == 1)

```

```
92     c = [hodn(1)-.1; hodn(1) ; hodn(1)+.1];
93     else
94         c = [(3*hodn(1)-hodn(2))/2];
95         for i=1:pocex-1
96             rozd = hodn(i+1)-hodn(i);
97             c = [c ; hodn(i) ; hodn(i)+rozd/2];
98         end
99         c = [c ; hodn(pocex) ; (3*hodn(pocex)-hodn(pocex-1))/2];
100     end
101 end
102
103 i=1;
104 pockor=0;
105 while(pockor==0)
106     pomo = sort( double( solve(U-c(i)) ) );
107     kor=[];
108     for k=1:length(pomo)
109         if( isreal(pomo(k)) )
110             kor = [kor;pomo(k)];
111         end
112     end
113     kor = unique(kor);
114     pockor = length(kor);
115     i = i+1;
116 end
117 if(kor(1) < mini)
118     mini = kor(1);
119 end
120 if(kor(pockor) > maxi)
121     maxi = kor(pockor);
122 end
123
124 i=length(c);
125 pockor=0;
126 while(pockor==0)
127     pomo = sort( double( solve(U-c(i)) ) );
128     kor=[];
129     for k=1:length(pomo)
130         if( isreal(pomo(k)) )
131             kor = [kor;pomo(k)];
132         end
133     end
134     kor = unique(kor);
135     pockor = length(kor);
136     i = i-1;
137 end
138 if(kor(1) < mini)
139     mini = kor(1);
```

```

140 end
141 if(kor(pockor) > maxi)
142     maxi = kor(pockor);
143 end
144
145 roz = maxi-mini;
146 u = linspace(mini-roz/10,maxi+roz/10,100);
147 v = subs(U,x,u);
148 cla(handles.F_axes)      %vymazanie osi
149 cla(handles.port_axes)
150 plot(handles.F_axes,u,v,'LineWidth',1.5);  %nakreslenie U(x)
151 axes(handles.F_axes);
152 ymax = c(length(c)) + ( c( length(c) ) - c( length(c)-1 ) );
153 ymin = c(1) - ( c(2)-c(1) );
154
155 set(gca,'yLim',[ymin ymax]);
156 set(gca,'xLim',[mini-roz/10 maxi+roz/10]); %nastavenie rozsahu osi
157 axes(handles.port_axes);
158 set(gca,'xLim',[mini-roz/10 maxi+roz/10]);
159 farby = lines(length(c));
160
161 for i=1:length(c)
162     farby(i,:) = farby(i,:);
163     axes(handles.F_axes);
164     line([mini-roz/10,maxi+roz/10],[c(i),c(i)],'Color',farby(i,:))
165     axes(handles.port_axes);
166     pomo = sort( double( solve(U-c(i)) ) );
167     pomo = round(1000000*pomo)/1000000;
168     kor=[];
169     for k=1:length(pomo)
170         if( isreal(pomo(k)) )
171             kor = [kor;pomo(k)];
172         end
173     end
174     kor = unique(kor);
175     pockor = length(kor);  %pocet extremov U(x)
176     if(pockor == 0)
177         if(subs(U,x,0) - c(i)<0)
178             if(pocMex==1)
179                 u = linspace(mini-roz/10,maxi+roz/10,100);
180                 v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );
181                 hold on;
182                 plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
183                 hold on;
184                 plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
185             else
186                 u = linspace(mini-roz/10,maxi+roz/10,100);
187                 v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );

```

```

188         hold on;
189         plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
190         hold on;
191         plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
192     end
193 end
194 elseif(pockor==1)
195     if(subs(U,x,kor(1)-eps)-c(i)<0)
196         u = linspace(mini-roz/10,kor(1),100);
197         v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );
198         hold on;
199         plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
200         hold on;
201         plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
202     end
203     if(subs(U,x,kor(1)+eps)-c(i)<0)
204         u = linspace(kor(1),maxi+roz/10,100);
205         v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );
206         hold on;
207         plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
208         hold on;
209         plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
210     end
211     if(abs( subs(U,x,kor(1))-c(i) )<eps && subs(U,x,kor(1)+eps)-c(i)>0 && ...
212         subs(U,x,kor(1)-eps)-c(i)>0)
213         plot(handles.port_axes,kor(1),0, '.', 'Color',farby(i,:), 'LineWidth',1.6)
214     end
215 else
216     if(subs(U,x,kor(1)-eps)-c(i)<0)
217         u = linspace(mini-roz/10,kor(1),100);
218         v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );
219         hold on;
220         plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
221         hold on;
222         plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
223     end
224     if(subs(U,x,kor(pockor)+eps)-c(i)<0)
225         u = linspace(kor(pockor),maxi+roz/10,100);
226         v = real( sqrt( 2*(-subs(U,x,u) + c(i)) ) );
227         hold on;
228         plot(handles.port_axes,u,v,'Color',farby(i,:), 'LineWidth',1.6);
229         hold on;
230         plot(handles.port_axes,u,-v,'Color',farby(i,:), 'LineWidth',1.6);
231     end
232
233     k=1;
234     while(k<pockor)
235         rozd = kor(k+1) - kor(k);

```

```

236         if(abs( subs(U,x,kor(k))-c(i) )<eps && subs(U,x,kor(k)+eps)-c(i)>0 && ...
237             subs(U,x,kor(k)-eps)-c(i)>0)
238             plot(handles.port_axes,kor(k),0, '.', 'Color', farby(i,:), 'LineWidth', 1.6);
239         elseif(subs(U,x,kor(k)+rozd/2)-c(i)<0)
240             u = linspace(kor(k),kor(k+1),100);
241             v = real( real( sqrt( 2*(-subs(U,x,u) + c(i)) ) ) );
242             hold on;
243             plot(handles.port_axes,u,v, 'Color', farby(i,:), 'LineWidth', 1.6);
244             hold on;
245             plot(handles.port_axes,u,-v, 'Color', farby(i,:), 'LineWidth', 1.6);
246         end
247         k = k+1;
248     end
249     if(abs( subs(U,x,kor(k))-c(i) )<eps && subs(U,x,kor(k)+eps)-c(i)>0 && ...
250         subs(U,x,kor(k)-eps)-c(i)>0)
251         plot(handles.port_axes,kor(k),0, '.', 'Color', farby(i,:))
252     end
253 end
254 end
255
256 axes(handles.port_axes);
257 axis tight;
258 set(gca, 'xLim', [mini-roz/10 maxi+roz/10]);
259
260 global pole;
261 if(pole)
262     uLim = get(handles.port_axes, 'xLim')*.9;
263     u = linspace(uLim(1),uLim(2),8);
264     vLim = get(handles.port_axes, 'yLim')*.9;
265     v = linspace(vLim(1),vLim(2),8);
266     [u,v] = meshgrid(u,v);
267     dv = subs(f,x,u);
268     axes(handles.port_axes);
269     quiver(u,v,v,dv, 'Color', [.5 .5], 'LineWidth', .5);
270 end
271
272 function check_pole_Callback(hObject, eventdata, handles)
273 checkboxStatus = get(handles.check_pole, 'Value');
274 global pole;
275 if(checkboxStatus)
276     pole = 1;
277 else
278     pole = 0;
279 end

```