



Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Grižtamojo ryšio valdymo realizacija trupmeninės eilės sistemose

Baigiamasis magistro studijų projektas

Martas Ulozevičius

Projekto autorius

Doc. dr. Tadas Telksnys

Vadovas

Kaunas, 2025



Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Grižtamojo ryšio valdymo realizacija trupmeninės eilės sistemose

Baigiamasis magistro studijų projektas

Taikomoji matematika (6211AX006)

Martas Ulozevičius

Projekto autorius

Doc. dr. Tadas Telksnys

Vadovas

Doc. dr. Rasa Šmidlaitė

Recenzentė

Kaunas, 2025



Kauno technologijos universitetas

Matematikos ir gamtos mokslų fakultetas

Martas Ulozevičius

Grižtamojo ryšio valdymo realizacija trupmeninės eilės sistemose

Akademinio sąžiningumo deklaracija

Patvirtinu, kad:

1. baigiamąjį projektą parengiau savarankiškai ir sąžiningai, nepažeisdama(s) kitų asmenų autoriaus ar kitų teisių, laikydamasi(s) Lietuvos Respublikos autorių teisių ir gretutinių teisių įstatymo nuostatų, Kauno technologijos universiteto (toliau – Universitetas) intelektinės nuosavybės valdymo ir perdavimo nuostatų bei Universiteto akademinės etikos kodekse nustatytų etikos reikalavimų;
2. baigiamajame projekte visi pateikti duomenys ir tyrimų rezultatai yra teisingi ir gauti teisėtai, nei viena šio projekto dalis nėra plagijuota nuo jokių spausdintinių ar elektroninių šaltinių, visos baigiamojo projekto tekste pateiktos citatos ir nuorodos yra nurodytos literatūros sąrašė;
3. įstatymų nenumatytų piniginių sumų už baigiamąjį projektą ar jo dalis niekam nesu mokėjęs (-usi);
4. suprantu, kad išaiškėjus nesąžiningumo ar kitų asmenų teisių pažeidimo faktui, man bus taikomos akademinės nuobaudos pagal Universitete galiojančią tvarką ir būsiu pašalinta(s) iš Universiteto, o baigiamasis projektas gali būti pateiktas Akademinės etikos ir procedūrų kontrolieriaus tarnybai nagrinėjant galimą akademinės etikos pažeidimą.

Martas Ulozevičius

Patvirtinta elektroniniu būdu

Ulozevičius, Martas. Grįžtamojo ryšio valdymo realizacija trupmeninės eilės sistemose. Magistro studijų baigiamasis projektas / vadovas doc. dr. Tadas Telksnys; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Taikomoji matematika (Matematikos mokslai).

Reikšminiai žodžiai: Grįžtamasis ryšis, trupmeninės eilės sistemos, trupmeninės išvestinės, netiesinė dinamika, chaoso stabilizacija, Pyrago metodas

Kaunas, 2025. 40 p.

Santrauka

Projekte tiriamos trupmeninės eilės išvestinių sistemos, kaip jų dinamika keičiasi kintant eilei. Paimamas keletas trupmeninės eilės apibendrinimų esamoms žymioms sistemoms kaip Rossler'io atraktorius, Brusselatorius, Van der Pol osciliatorius, ir randami parametrai kuriems egzistuoja chaosas arba ribiniai ciklai, tuomet pritaikomas grįžtamojo ryšio metodas sistemų stabilizacijai. Aptariami skaitiniai metodai.

Ulozevičius, Martas. Feedback Control Realization in Fractional-Order Systems. Master's Final Degree Project / supervisor lect. dr. Tadas Telksnys; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Applied Mathematics (Mathematical Sciences).

Keywords: Feedback control, fractional-order systems, fractional derivatives, nonlinear dynamics, chaos stabilization, Pyragas method.

Kaunas, 2025. 40 p.

Summary

In this project we investigate fractional-order derivative systems, how their dynamics change by varying the order. We take the fractional-order generalizations of a couple of well-known systems such as the Rossler attractor, the Brusselator, Van der Pol oscillator, and find parameters for which either chaos or limit cycles exist, then apply feedback control to stabilize the system. Numerical methods are discussed.

Turinys

Lentelių sąrašas	7
Paveikslų sąrašas	8
Įvadas	9
1. Literatūros apžvalga	10
1.1. Trupmeninės išvestinės	11
1.2. Skaitiniai metodai trupmeninės eilės išvestinėms	14
1.2.1. Skaitinis diferencijavimas	14
1.2.2. Diferencialinių lygčių sprendimas	16
1.2.3. Diferencialinių lygčių sistemų sprendimas	19
1.3. Chaosas trupmeninėse sistemose	21
1.4. Grįžtamo ryšio metodas	23
2. Duomenys ir tyrimų metodai	28
2.1. Naudojama programinė įranga	28
3. Tyrimų rezultatai ir jų aptarimas	29
3.1. Trupmeninė Van der Pol sistema	29
3.2. Trupmeninė Rossler sistema	36
Išvados	38
Literatūros sąrašas	39
Priedai	41
1 priedas. Programos kodas	41

Lentelių sąrašas

1 lentelė. Metodų maksimalios absoliučios paklaidos skirtingiems žingsnio dydžiams	18
2 lentelė. Stabilus trupmeninio VDP ribinio ciklo periodai	20

Paveikslų sąrašas

1 pav. $\sin(t)$ trupmeninės išvestinės	12
2 pav. Skaitinio metodo palyginimas su teoriniu rezultatu	15
3 pav. Skaitinio metodo paklaidos.	15
4 pav. Euler ir PC-FFT metodų absoliučios paklaidos	18
5 pav. VDP ribiniai ciklai su skirtingomis α vertėmis	18
6 pav. Sprendinio trajektorijos su įvairiomis α vertėmis	20
7 pav. Rossler sistemos trajektorijos vienoje plokštumoje	24
8 pav. Pirmo varianto trajektorija kelioms skirtingoms K vertėms	25
9 pav. Antro varianto trajektorija kelioms skirtingoms K vertėms	26
10 pav. Rossler atraktorius, kai $\tau \in [11; 16]$	27
11 pav. Stabilios trupmeninės VDP sistemos fazinis vaizdas	29
12 pav. Paskutinis 10% sistemos iteracijų	30
13 pav. Paskutinių sistemos iteracijų priartintas vaizdas	30
14 pav. Stabilios trupmeninės VDP sistemos pseudo-periodinis ribinis ciklas	31
15 pav. Nestabilios trupmeninės VDP sistemos fazinis vaizdas	31
16 pav. Nestabilios trupmeninės VDP sistemos pradinių taškų dinamika	32
17 pav. Išskirta nestabilaus ribinio ciklo kreivė	32
18 pav. Grįžtamo ryšio grafikai naudojant stabilų ribinį ciklą	33
19 pav. Grįžtamo ryšio grafikai naudojant nestabilų ribinį ciklą	34
20 pav. Sukurta pseudo-periodinė trajektorija	34
21 pav. Grįžtamojo ryšio funkcijos grafikas	34
22 pav. Trajektorių x koordinatės su keliomis τ vertėmis	36
23 pav. Originalios ir stabilizuotos trupmeninės Rossler sistemos 3D vaizdas	37
24 pav. Stabilizuotos sistemos grįžtamojo ryšio funkcijos grafikas	34
25 pav. Stabilizuotos sistemos y ir z koordinatės	37

Įvadas

Šiame darbe teikiame dėmesį diferencialinio operatoriaus apibendrinimui, jo lygčių sprendimui bei sistemų dinamikai.

Trupmeninės išvestinės naudojamos modeliuoti tokius procesus, kurių būseną priklauso ne tik nuo lokalių verčių, bet ir nuo visų praeities verčių – t.y. procesus, turinčius atmintį. Tai gali būti klampių, viskoelastinių medžiagų fizika, skysčių srautas, ar tam tikra elektrochemija. Tokiuose procesuose neretai susidaro įvairūs nestabilumai, kurie praktikoje dažniausiai yra nenaudingi.

Klasikinių sistemų atveju sukurta daug metodų chaosui slopinti, ir vienas iš populiariausių yra grįžtamasis ryšys dėl savo paprastumo ir lengvo praktinio įgyvendinimo.

Projekto tikslas – pritaikyti klasikinių sistemų grįžtamojo ryšio metodą trupmeninių sistemų nestabilumams panaikinti.

Uždaviniai:

1. peržvelgti skaitinius metodus klasikinėms sistemoms;
2. pritaikyti skaitinius metodus trupmeninėms sistemoms;
3. palyginti klasikinių ir trupmeninių sistemų dinamiką;
4. aptarti metodų efektyvumą.

1. Literatūros apžvalga

Nors trupmeninės eilės išvestinės pirmą kartą paminėtos 1695 m. [1] ir formaliai pristatytos 1832 m. [2], labai ilgą laiką jos buvo tik matematinis smalsumas – nebuvo nei intuityvaus fizinio reiškinių, nei praktinės naudos.

Pirmi praktiniai taikymai atsirado 1890 m. modeliuojant elektros linijas [3]. 2008-2014 m. trupmeninės išvestinės sėkmingai pritaikytos sprendžiant skysčių srauto problemas [4, 5]. Šiuo metu trupmeniniai metodai ypač populiariūs tiriant viskoelastines, tamprias medžiagas [6, 7, 8], įvairius biocheminius procesus [9, 10], epidemijų dinamiką [11, 12, 13], elektrinius reiškinius [14, 15], biomediciną [16], termodinamiką [17, 18], skysčių mechaniką [19, 20], vaizdų manipuliaciją [21, 22], ar net kvantinę mechaniką [23, 24].

Kas pasikeitė, kad trupmeninės išvestinės staiga tapo naudingos modeliuoti tam tikrus fizinius procesus? Pasirodo, kad jos turi atmintį – funkcijos vertė viename taške priklauso nuo visų praeities verčių, kuomet sveiko skaičiaus išvestinėms aktuali tik lokali informacija. Jei klasikinėms sistemoms, naujai būsenai apskaičiuoti užtenka tik esamos būsenos, tai trupmeninėms sistemos reikia žinoti visas buvusias būsenas, kas įeiną į minėtų fizinių reiškinių charakteristikas.

Nepaisant atminties savybės, didelis trupmeninių išvestinių savybių skaičius sutampa su klasikinėmis. Taip pat egzistuoja ekvivalentiškų chaoso, ir ribinio ciklo reiškinių [25, 26, 27].

1.1. Trupmeninės išvestinės

Skirtingai nei klasikinei išvestinei, trupmeninės eilės išvestinėms nėra sutarto vienintelio apibrėžimo, nes yra be galo daug formų suteikiančių tas pačias norimas savybes.

Du populiariausi apibrėžimai yra Riemann-Liouville [28] ir Caputo [29] tipai:

$${}^{RL}D_{t_0}^{\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dt^n} \int_{t_0}^t f(u)(t-u)^{n-\alpha-1} du; \quad (1.1.1)$$

$${}^CD_{t_0}^{\alpha}f(t) = \frac{1}{\Gamma(n-\alpha)} \int_{t_0}^t f^{(n)}(u)(t-u)^{n-\alpha-1} du; \quad (1.1.2)$$

čia Γ – gerai žinoma gama funkcija; t_0 – apatinio režio parametras; α – išvestinės eilės parametras; $n = [\alpha]$ – pastarasis parametras apvalintas iki didesnio sveiko skaičiaus. Taria, kad $\alpha \geq 0$.

Iš tikro, šitie apibrėžimai yra beveik ekvivalentūs, nes turime išraišką:

$${}^{RL}D_{t_0}^{\alpha}f(t) = {}^CD_{t_0}^{\alpha}f(t) + \frac{(t-t_0)^{-\alpha}}{\Gamma(1-\alpha)} f(t_0), \quad 0 \leq \alpha < 1. \quad (1.1.3)$$

T.y. jie skiriasi tik elementaria išraiška kuri, priklausomai nuo nagrinėjamų funkcijų pradinių verčių, gali būti nulinė.

Projekte pasirinkta Caputo išraiška dėl jos savybių skaičiuojant antros eilės sistemas – sprendimas priklauso tik nuo funkcijos pradinių verčių sveiko skaičiaus išvestinėms [30], bei konstantos išvestinė visoms α vertėms yra nulinė funkcija. Toliau operatorius ${}^CD_a^{\alpha}$ žymimas tik kaip D_a^{α} , arba D^{α} jei $a = 0$, ar apatinis režis tiesiog neaktualus. Taip pat laikoma, kad $0 \leq \alpha < 1$.

Svarbiausios Caputo trupmeninės išvestinės savybės [31]:

$$\bullet \quad D_{t_0}^0 f(t) = f(t) - f(t_0); \quad (1.1.4)$$

$$\bullet \quad \lim_{\alpha \rightarrow 1^-} D_{t_0}^{\alpha} f(t) = f'(t), \quad t > a; \quad (1.1.5)$$

$$\bullet \quad D_{t_0}^{\alpha} f(t_0) = 0; \quad (1.1.6)$$

$$\bullet \quad D_{t_0}^{\alpha} [D_{t_0}^{\beta} f(t)] = D_{t_0}^{\alpha+\beta} f(t); \quad (1.1.7)$$

$$\bullet \quad D_{t_0}^{\alpha} [c \cdot f(t) + d \cdot g(t)] = c \cdot D_{t_0}^{\alpha} f(t) + d \cdot D_{t_0}^{\alpha} g(t). \quad (1.1.8)$$

Dėl šių savybių, turima geresnė nuojautas trupmeninėms išvestinėms. Pvz. $D^{0.5}$ yra toks operatorius, kuris dukart taikytas funkcijai $f(t)$ gražins jos išvestinę $f'(t)$.

Toliau pateikiama keletas trupmeninių išvestinių pavyzdžių elementarioms funkcijoms:

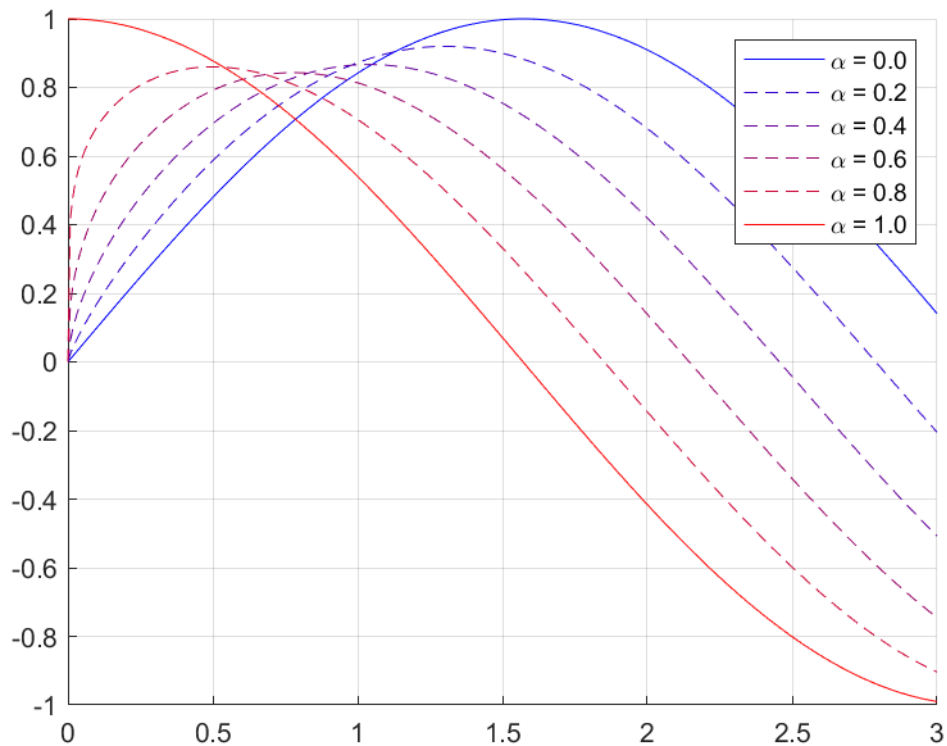
$$D_{t_0}^\alpha [t^p] = \frac{\Gamma(p+1)}{\Gamma(p+1-\alpha)} [t^{p-\alpha} - t_0^{p-\alpha}], \quad p > 0; \quad (1.1.9)$$

$$D_0^\alpha [e^{pt}] = pt^{1-\alpha} E_{1,2-\alpha}(pt); \quad (1.1.10)$$

$$D_0^\alpha [\sin(pt)] = pt^{1-\alpha} E_{2,2-\alpha}(-p^2 t^2); \quad (1.1.11)$$

$$D_0^\alpha [\cos(pt)] = t^{-\alpha} E_{2,1-\alpha}(-p^2 t^2); \quad (1.1.12)$$

čia $E_{u,v}(t) = \sum_{k=0}^{\infty} t^k / \Gamma(uk + v)$ – Mittag-Leffler funkcija. Dėl (1.1.8) ir (1.1.9) savybių galima rasti panašias išraiškas daugumai laipsninių eilučių, bet formų naudojančių tik elementarias funkcijas neverta tikėtis [32].



1 pav. $\sin(t)$ trupmeninės išvestinės

Matoma, kad trupmeninės išvestinės sugeneruoja tarpines kreives tarp originalios funkcijos ir jos išvestinės. Verta paminėti, jog nors visoms $0 \leq \alpha < 1$ vertėms $D^\alpha f(0) = 0$, riba iš dešinės pusės sutampa su išvestinės verte tame taške. T.y.

$$\lim_{t \rightarrow 0^+} \lim_{\alpha \rightarrow 1^-} D^\alpha f(t) = f'(0), \text{ bet } \lim_{\alpha \rightarrow 1^-} \lim_{t \rightarrow 0^+} D^\alpha f(t) = 0.$$

Vėliau taip pat prireiks ir atvirkštinio operatoriaus – trupmeninės eilės integralo. Tam užrašomas Riemann-Liouville apibrėžimas:

$$I_{t_0}^\alpha f(t) = \frac{1}{\Gamma(\alpha)} \int_{t_0}^t f(u) (t-u)^{\alpha-1} du. \quad (1.1.13)$$

Jo paskirtis yra apibendrinti integralinį operatorių:

$$\mathcal{J}^1 f(t) = \int_0^t f(u) du, \quad \mathcal{J}^n f(t) = \int_0^t \mathcal{J}^{n-1} f(u) du. \quad (1.1.14)$$

Savybės:

$$\bullet \quad I_0^n f(t) = \mathcal{J}^n f(t), \quad n \in \mathbb{N}; \quad (1.1.15)$$

$$\bullet \quad D_{t_0}^\alpha [I_{t_0}^\alpha f(t)] = f(t); \quad (1.1.16)$$

$$\bullet \quad I_{t_0}^\alpha [D_{t_0}^\alpha f(t)] = f(t) - T_{m-1}[f; t_0](t); \quad (1.1.17)$$

čia $T_{m-1}[f; t_0](t) = \sum_{k=0}^{m-1} \frac{f^{(k)}(a)(t-t_0)^k}{k!}$, $m = [\alpha]$ ($m - 1$ eilės Teiloro polinomas).

1.2. Skaitiniai metodai trupmeninės eilės išvestinėms

Žiūrint (1.1.2) apibrėžimą aiškiai matyti, kad skaitinis diferencijavimas trupmeninėms eilėms bus daug sudėtingesnis nei sveikiems skaičiams. Klasikinės išvestinės vertė taške t gali būti apskaičiuota tik naudojant du įverčius - $f(t)$ ir $f(t + h)$ kažkokiam poslinkiui h , o trupmeninei reikia visų įverčių $f(u)$, $u \in (0, t)$. Taip pat funkcija $g(u) = (t - u)^{-\alpha}$ turi asimptotę ties viršutiniu rėžiu t .

Teoriškai, asimptotė nėra problema nes $\int_0^t (t - u)^{-\alpha} du = \frac{t^{1-\alpha}}{1-\alpha}$, $0 \leq \alpha < 1$. T.y. jei funkcija tolydi intervale $[0, t]$, tai integralas visada egzistuos. Tačiau, praktiškai tai smarkiai pasunkina skaičiavimus, nes negalima skaitiškai integruoti ties šiuo tašku, bet jo taip pat negalima ir ignoruoti, nes didžioji dalis integralo rezultato sukoncentruota prie asimptotės. Galima integralą išskaidyti į intervalus $[0, t - h]$ ir $[t - h, t]$ ir antrame aproksimuoti $\int_{t-h}^t f'(u)(t - u)^{-\alpha} du \approx f'(t) \cdot \frac{h^{1-\alpha}}{1-\alpha}$, tai iš tikro duoda visai neblogus rezultatus, bet galima sukurti ir kažką geresnio [33 p. 3].

1.2.1. Skaitinis diferencijavimas

Pasirinkę poslinkį $h > 0$ ir pažymėję $t_n = kh$, $f_n = f(t_n)$ turime:

$$\begin{aligned} \Gamma(1 - \alpha) \cdot D^\alpha f(t_n) &= \int_0^{t_n} f'(u)(t - u)^{-\alpha} du = \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} f'(u)(t - u)^{-\alpha} du = \\ &= \sum_{k=0}^{n-1} \frac{f(t_{k+1}) - f(t_k)}{h} \int_{t_k}^{t_{k+1}} (1 + \mathcal{O}(h))(t - u)^{-\alpha} du = \\ &= \frac{h^{-\alpha}}{1 - \alpha} \sum_{k=0}^{n-1} [f(t_{k+1}) - f(t_k)][(n - k)^{1-\alpha} - (n - k - 1)^{1-\alpha}] + \mathcal{O}(h^{2-\alpha}). \end{aligned} \quad (1.2.1)$$

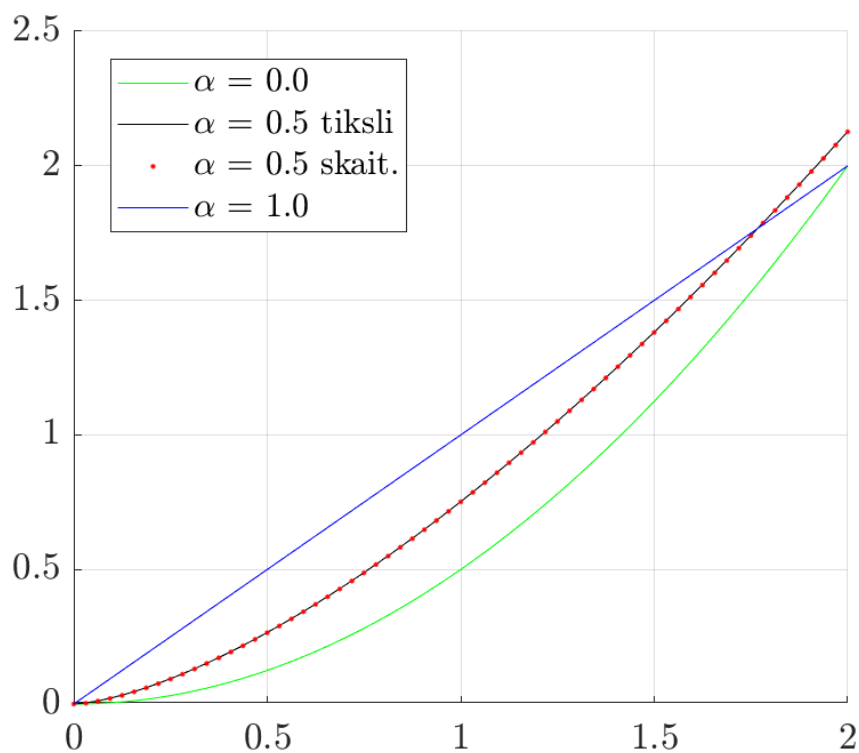
Sugrupavus narius ties $f(t_k)$ vertėmis gauname:

$$D^\alpha f(t_n) = \frac{h^{-\alpha}}{\Gamma(2 - \alpha)} \sum_{k=0}^n f(t_{n-k}) \varphi_{n,k}^\alpha + \mathcal{O}(h^{2-\alpha}); \quad (1.2.2)$$

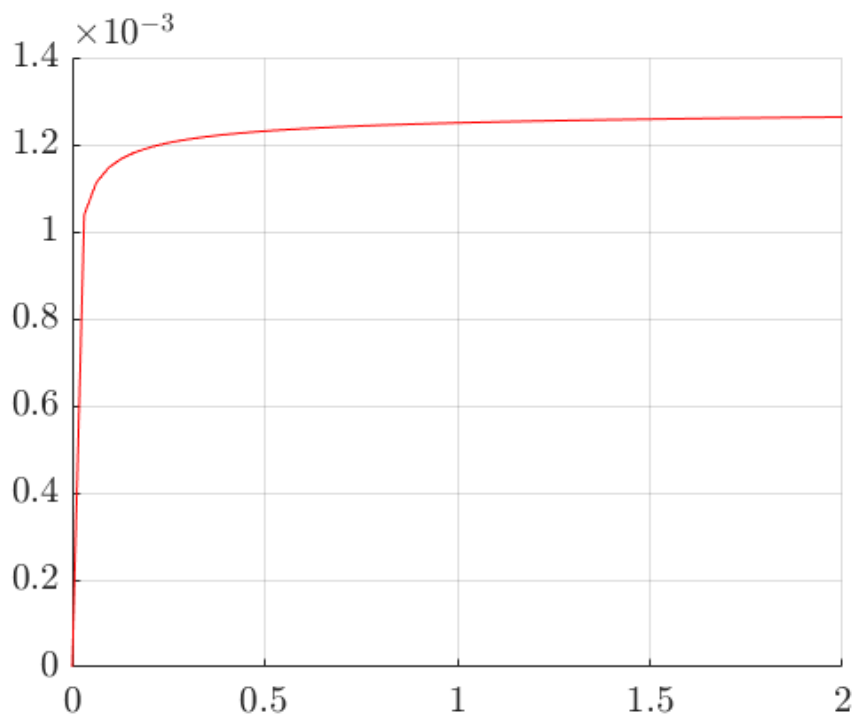
$$\text{čia } \varphi_{n,k}^\alpha = \begin{cases} 1, & k = 0 \\ (k - 1)^{1-\alpha} - 2k^{1-\alpha} + (k + 1)^{1-\alpha}, & 1 \leq k \leq n - 1. \\ (n - 1)^{1-\alpha} - n^{1-\alpha}, & k = n \end{cases} \quad (1.2.3)$$

Formulė (1.2.1) įprastai vadinama L1-tipo aproksimacija [34 p. 2]. Iš integralinės formos matyti, kad norint apskaičiuoti išvestinės vertę taške t_n reikia žinoti originalios funkcijos vertes taškuose $t_0, t_1, t_2, \dots, t_n$.

Trumpam metodo tikslumo pamatavimui, imame funkciją $f(t) = t^2/2$. Jos teorinė trupmeninės išvestinės išraiška yra $\frac{2}{\Gamma(3-\alpha)} t^{2-\alpha}$. Išvestinės eilė tegu bus $\alpha = 0,5$, o žingsnio dydis pasirinktas $h = 1/32 = 0,03125$.



2 pav. Skaitinio metodo palyginimas su teoriniu rezultatu



3 pav. Skaitinio metodo paklaidos.

Bent šitai paprastai funkcijai maksimali absoliuti paklaida buvo 0,0013.

1.2.2. Diferencialinių lygčių sprendimas

Turint tinkamą skaitinio diferencijavimo metodą, galime kalbėti apie diferencialines lygtis.

Jei klasikinės bendros formos diferencialinės lygtis turi tokią išraišką:

$$\begin{cases} y'(t) = f(t, y(t)), \\ y(t_0) = y_0 \end{cases} \quad (1.2.4)$$

tai trupmeninės eilės lygtys atrodytų taip:

$$\begin{cases} D^\alpha y(t) = f(t, y(t)) \\ y(t_0) = y_0 \end{cases} \quad (1.2.5)$$

Jei neturėtume apribojimo $0 < \alpha < 1$, tai reiktų pridėti pradines sąlygas

$$y'(t_0) = y'_0, y''(t_0) = y''_0, \dots, y^{(m-1)}(t_0) = y_0^{(m-1)}, m = [\alpha].$$

Kaip minėta pirmame punkte, naudojant Caputo apibrėžimą unikaliam sprendiniui egzistuoti užtenka užrašyti jo pradinio laiko momento vertes sveiko skaičiaus išvestinėms. Naudojant Riemann-Liouville apibrėžimą, reiktų pradinių verčių trupmeninės eilės išvestinėms.

Lygties sprendimui galima pasitelkti praeitos skilties skaitinio diferencijavimo metodą. Pažymėjus $t = t_n, C_\alpha = h^{-\alpha}/\Gamma(1 - \alpha)$, aproksimuojama:

$$D^\alpha y(t_n) \approx C_\alpha \sum_{k=0}^n y_{n-k} \varphi_{n,k}^\alpha = f(t_n, y_n) \Rightarrow y_n = C_\alpha^{-1} f(t_n, y_n) - \sum_{k=1}^n y_{n-k} \varphi_{n,k}^\alpha. \quad (1.2.6)$$

Kadangi tipiškai išspręsti lygtį $y_n = f(t_n, y_n) + \text{const}$ vertei y_n yra per daug sudėtinga, viduje funkcijos f atliekama dar viena aproksimacija: $y_n \approx y_{n-1} + (y_{n-1} - y_{n-2}) = 2y_{n-1} - y_{n-2}$ ir tariame, kad $y_0 = y_{-1} = y_{-2}$. Galutinis algoritmas atrodo taip:

$$y_n = C_\alpha^{-1} f(t_n, 2y_{n-1} - y_{n-2}) - \sum_{k=1}^n y_{n-k} \varphi_{n,k}^\alpha, \quad n = 1, 2, 3, \dots \quad (1.2.7)$$

Tai yra rekursinis metodas, apskaičiuojantis sekančią y_n vertę, naudojant visas ankstesnes vertes $y_{n-1}, y_{n-2}, \dots, y_1, y_0$. Algoritmas yra Euler metodo apibendrinimas trupmeninės eilės išvestinėms. Tačiau, kaip Euler metodas, jis nėra pakankamai efektyvus turint didelį iteracijų skaičių.

Tam imama (1.2.5) išraiška ir pasinaudojama savybe (1.1.17):

$$y(t) = T_{m-1}[y; a](t) + \frac{1}{\Gamma(\alpha)} \int_a^t f(u, y(u))(t-u)^{\alpha-1} du. \quad (1.2.8)$$

Žymima $t_n = a + nh$, $y_n = y(t_n)$ ir išskaidomas integralas į smulkesnius intervalus:

$$y_n = T_{m-1}[y; a](t_n) + \frac{1}{\Gamma(\alpha)} \sum_{k=0}^{n-1} \int_{t_k}^{t_{k+1}} f(u, y(u))(t - u)^{\alpha-1} du. \quad (1.2.9)$$

Priklausomai nuo taikomos aproksimacijos funkcijai $f(u, y(u))$ gaunamos skirtingos skaitinės schemos. Aproksimuojant funkciją jos galutinėmis vertėmis – $f(t_k, y_k)$ arba $f(t_{k+1}, y_{k+1})$, sukuriamos tikslios (1.2.10) ir numanomos (1.2.11) stačiakampės schemos. Pirmos eilės polinomo interpoliacija suteikia numanomą trapezoidinę (1.2.12) schemą [35 p. 5].

$$y_n = T_{m-1}[y; a](t_n) + \frac{h^\alpha}{\Gamma(1 + \alpha)} \sum_{k=0}^{n-1} b_{n-k-1} f(t_k, y_k); \quad (1.2.10)$$

$$y_n = T_{m-1}[y; a](t_n) + \frac{h^\alpha}{\Gamma(1 + \alpha)} \sum_{k=0}^n b_{n-k} f(t_k, y_k); \quad (1.2.11)$$

$$y_n = T_{m-1}[y; a](t_n) + \frac{h^\alpha}{\Gamma(2 + \alpha)} \left[\tilde{c}_n f_0 + \sum_{k=1}^n c_{n-k} f(t_k, y_k) \right]; \quad (1.2.12)$$

čia

$$b_k = (k + 1)^\alpha - k^\alpha;$$

$$\tilde{c}_k = (k - 1)^\alpha - k^\alpha;$$

$$c_k = \begin{cases} 1, & n = 0; \\ (n - 1)^{\alpha+1} - 2n^{\alpha+1} + (n + 1)^{\alpha+1}, & n \geq 1; \end{cases} \quad (1.2.13)$$

Numanomos schemos kaip (1.2.10) įprastai yra efektyvesnės už tikslias schemas, bet kaip ir anksčiau, $y_n = f(t_n, y_n) + \text{const}$ lygties sprendimas nėra visada įmanomas. Prediktoriaus-korektoriaus (PC) metodas sujungia abi schemas – pirmiausia tikslia schema apskaičiuojamas spėjimas y_n^p , tuomet jis įstatomas į numanomos schemos lygtį, sugeneruojant koreguotą vertę y_n .

Bendru atveju, visos schemos y_n vertės gaunamos konvoliuciniu tipu:

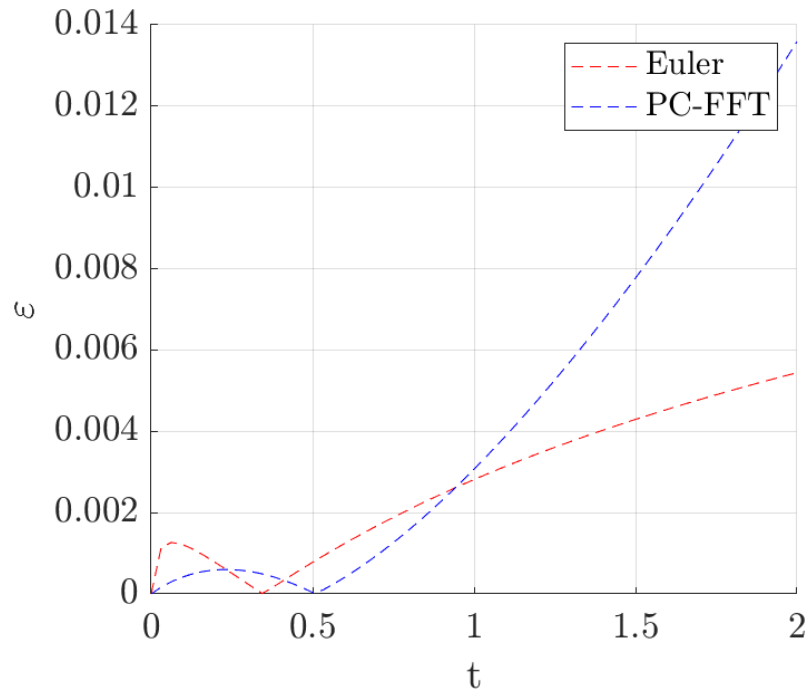
$$y_n = w_n + \sum_{k=0}^n c_{n-k} f_k, \quad f_k = f(t_k, y_k). \quad (1.2.14)$$

T.y. naujai vertei reikia susumuoti visas praeitas vertes. Dėl to, skaičiavimo sudėtingumas yra $\mathcal{O}(n^2)$, tai yra daug prasčiau nei klasikinių diferencialinių schemų $\mathcal{O}(n)$ sudėtingumas. Tačiau, pasitelkiant greitą Furjė transformaciją (FFT), galima tai sumažinti iki $\mathcal{O}(n \log(n)^2)$ [35 p. 10]

Sprendimo pavyzdžiui paimama lygtis:

$$\begin{cases} D^{0.5} y = 0.75\sqrt{\pi}x - 0.9375\sqrt{\pi}[\sqrt{x}y + x^3] \\ y(0) = 0 \end{cases} \quad (1.2.15)$$

Jos tikslus sprendinys yra $y(t) = t^{1.5} - t^{2.5}$.



4 pav. Euler ir PC-FFT metodų absoliučios paklaidos

Abu metodai naudojo tą patį $h = 2^{-5}$ žingsnio ilgį. Nors atrodo, kad mūsų išvestas metodas yra efektyvesnis didesnėms tolimesnėms vertėms, skaičiavimo laikas pasidarys labiau aktualus tiriant sistemas atliekant šimtus tūkstančių iteracijų.

1 lentelė. Metodų maksimalios absoliučios paklaidos $[0; 2]$ intervale skirtingiems žingsnio dydžiams

Žingsnis	Euler	PC-FFT
2^{-1}	3,3567	3,0453
2^{-2}	0,7554	0,8897
2^{-3}	0,0710	0,1948
2^{-4}	0,0190	0,0482
2^{-5}	0,0054	0,0136
2^{-6}	0,0016	0,0042
2^{-7}	0,0005	0,0014
2^{-8}	0,0002	0,0005

Visiems išbandytiems žingsniams padarytos išvados nepasikeitė.

1.2.3. Diferencialinių lygčių sistemų sprendimas

Jei dvimatė klasikinė diferencialinių lygčių sistemą atrodo taip:

$$\begin{cases} x'(t) = f(x(t), y(t)) \\ y'(t) = g(x(t), y(t)) \\ x(t_0) = x_0, \quad y(t_0) = y_0 \end{cases} ; \quad (1.2.16)$$

Tai tuomet trupmeninės eilės apibendrinimas būtų toks:

$$\begin{cases} D^\alpha x(t) = f(x(t), y(t)) \\ D^\beta y(t) = g(x(t), y(t)) \\ x(t_0) = x_0, \quad y(t_0) = y_0 \end{cases}. \quad (1.2.17)$$

Dažnai suvienodinamos eilės $\alpha = \beta$, ir apribojama $0 < \alpha, \beta < 1$.

Apibendrinti (1.2.6) metodą dvejoms dimensijoms nėra sudėtinga:

$$\begin{aligned} x_n &= C_\alpha^{-1} f(2x_{k-1} - x_{k-2}, 2y_{k-1} - y_{k-2}) - \sum_{k=1}^n x_{n-k} \varphi_{n,k}^\alpha ; \\ y_n &= C_\beta^{-1} g(2x_{k-1} - x_{k-2}, 2y_{k-1} - y_{k-2}) - \sum_{k=1}^n y_{n-k} \varphi_{n,k}^\beta, \quad n = 1, 2, 3, \dots \end{aligned} \quad (1.2.18)$$

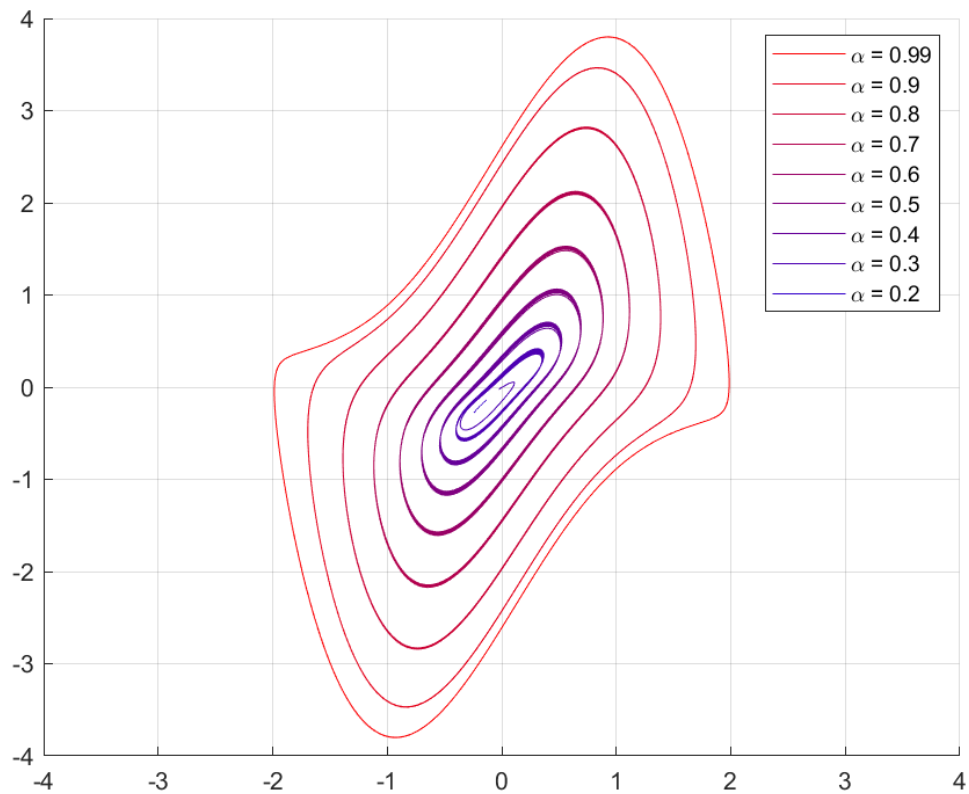
PC-FFT metodas naudojamame pakete leidžia naudoti tik vienodas išvestinių eiles.

Pavyzdžiui, paimamas trupmeninis Van der Pol (toliau trumpinamas kaip VDP) osciliatorius:

$$\begin{cases} D^\alpha x = y \\ D^\beta y = \mu(1 - x^2)y - x \\ x(0) = 0,25; \quad y(0) = 0,25 \end{cases} ; \quad (1.2.19)$$

su $\alpha = \beta$ ir $\mu = 2$, ir žiūrime kaip atrodo fazių grafikas kintant α vertei.

Kad geriau matytume ribinius ciklus, ignoruojame didelę dalį pradinių iteracijų.



5 pav. VDP ribiniai ciklai su skirtingomis α vertėmis

Matoma, kai α artėja vienetui, tai sistema beveik sutampa su klasikine. Taip pat α mažėjant link nulio yra įdomus sistemos elgesys – trajektorijos vietoj periodinės orbitos artėja prie kažkokio fiksuoto taško. T.y. įvyksta superkritinė Hopf bifurkacija.

2 lentelė. Stabilus trupmeninio VDP ribinio ciklo periodai

Eilė α	Periodas T
0,99	7,22
0,95	6,22
0,90	5,45
0,85	4,95
0,80	4,70
0,75	4,59
0,70	4,55
0,65	4,60
0,60	4,69
0,55	4,82
0,50	5,04

Skaičiavimai parodė, kad mažinant išvestinės eilę mažėjo ir ribinio ciklo periodas, kol pasiektas lokalus minimumas ties $\alpha \sim 0,7$. Toliau mažinant eilę, periodas pradėjo didėti.

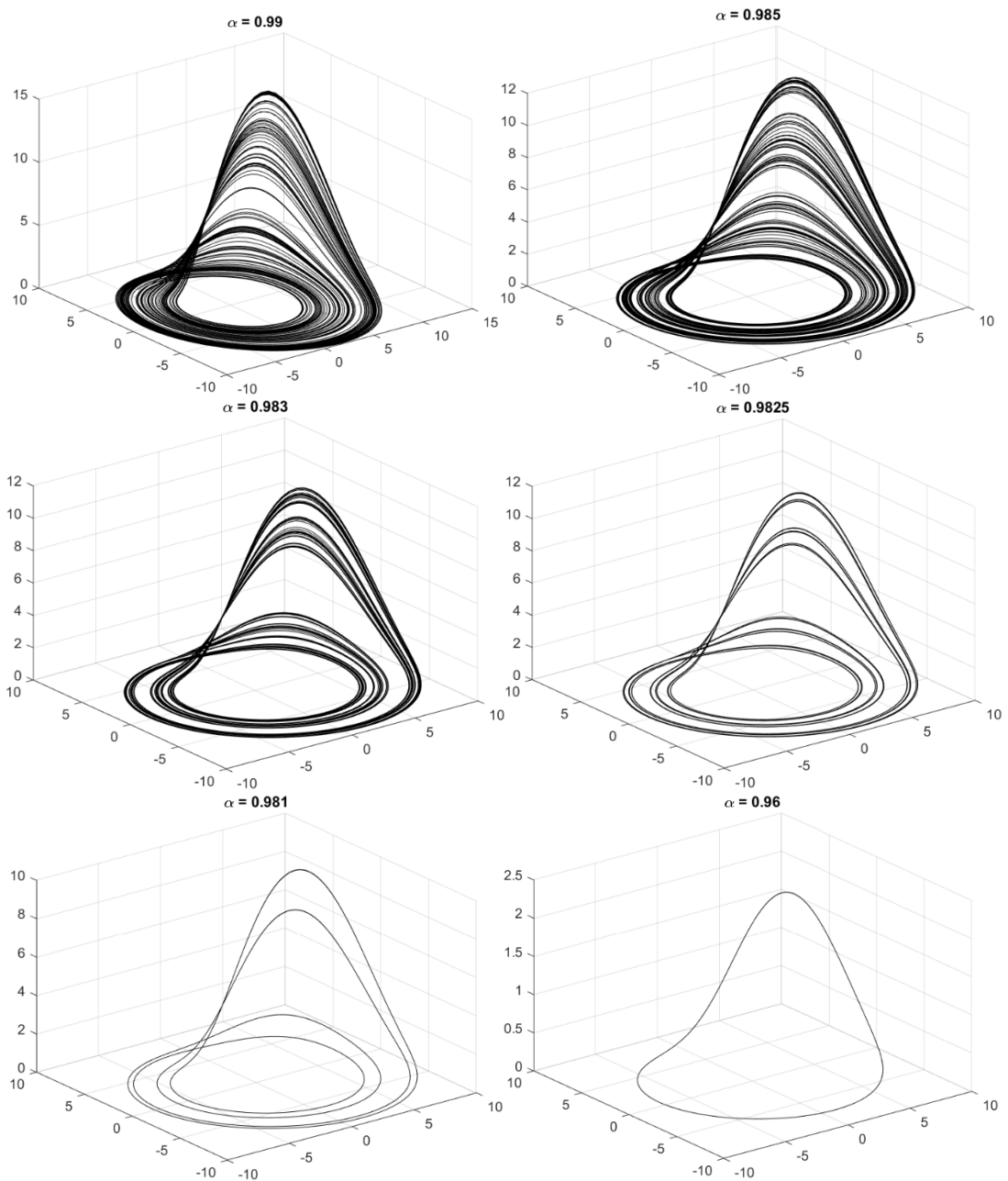
1.3. Chaosas trupmeninėse sistemose

Žinoma, kad pereinant nuo klasikinių iki trupmeninių sistemų, vis tiek galima rasti chaoso pavyzdžių [36, 37, 38, 39, 40].

Paimama Rossler sistema [41, 42] ir konvertuojama į trupmeninę:

$$\begin{cases} D^\alpha x = -y - z \\ D^\alpha y = x + ay \\ D^\alpha z = b + z(x - c) \\ x(0) = y(0) = z(0) = 0 \end{cases} ; \quad (1.3.1)$$

su $a = 0,2$; $b = 0,2$; $c = 5,7$. Klasikinė sistema turi chaotinę atraktorių [43] – visos trajektorijos sukongcentruotos tarp aiškių ribų, bet niekada idealiai nepasikartoja. Pateikiame vaizdus kelioms α vertėms, bet ne mažesnėms už 0,94, nes tuomet trajektorijas labai ilgą laiką svyruoja vienoje xy plokštumoje ir tik daug vėliau pakankamai „išsisiūbuoja“ į kitas z vertes. Braižant grafikus ignoruojame didelę dalį pradinių taškų, kad geriau matytume pseudo-periodines orbitas.



6 pav. Sprendinio trajektorijos su įvairiomis α vertėmis

Iš skaitinių rezultatų manoma, kad α sumažinus iki $\sim 0,983$ sistemoje dingsta chaosas – artimos trajektorijos susiglaudžia ir susidaro tikros periodinės orbitos.

Projektui naudojama $\alpha = 0,99$, kad grįžtamojo ryšio pritaikymo pavyzdys vis dar turėtų chaoso. Verta paminėti, kad skaitiniu atžvilgiu nesvarbu ar sistemoje egzistuoja chaosas, ar tiesiog didelis skaičius periodų.

1.4. Grįžtamojo ryšio metodas

1992 m. lietuvių matematikas Kęstutis Pyragas pristatė metodą chaotinių sistemų stabilizavimui, minimaliai jas pakoreguojant jų pačių sugeneruojamomis vertėmis [44], bet pats metodas gali būti pritaikytas ir kitokio tipo nestabilumams.

Paimame nestabilią sistemą – ji gali turėti nestabilų ribinį ciklą, arba būti chaotiška. Chaosui egzistuoti reikia bent 3 dimensijų [45 p. 210], tai tariame, kad mūsų diferencialinių lygčių sistema atrodo taip:

$$\begin{cases} x'(t) = f(x(t), y(t), z(t)) \\ y'(t) = g(x(t), y(t), z(t)) \\ z'(t) = k(x(t), y(t), z(t)) \\ x(0) = x_0, y(0) = y_0, z(0) = z_0 \end{cases} \quad (1.4.1)$$

Grįžtamo ryšio (arba Pyrago) metodas modifikuoja vieną iš šių lygčių pridedant priklausomybę nuo laiko. Sakykime, kad pasirinkta antra lygtis ir pakeičiame jos išraišką į:

$$y'(t) = g(x(t), y(t), z(t)) + F(t); \quad (1.4.2)$$

čia $F(t)$ – pasirenkama funkcija iš dviejų variantų:

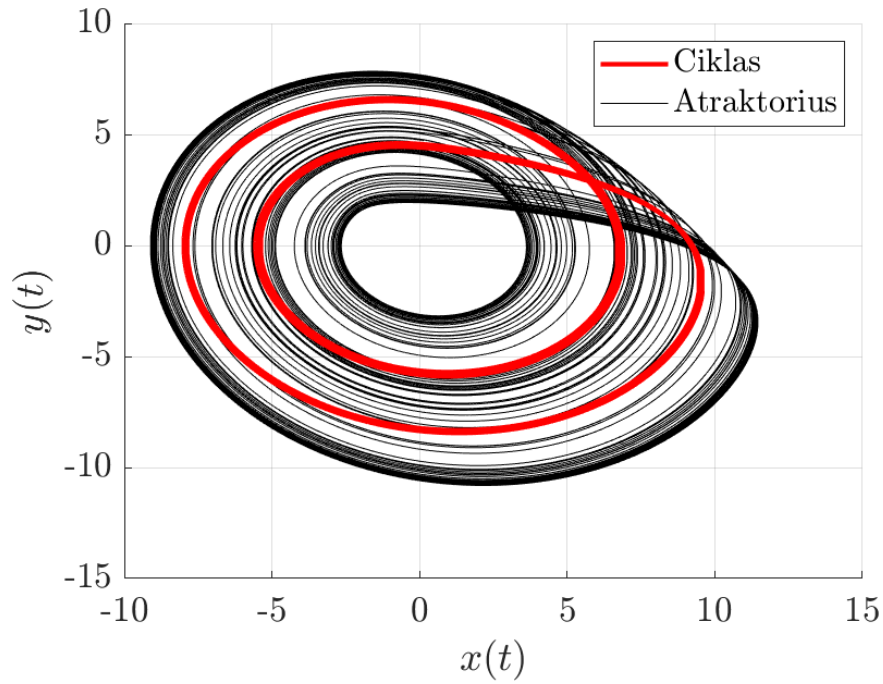
$$F(t) = K[y_i(t) - y(t)] = K \cdot D(t) \text{ (išorinės jėgos tipas);} \quad (1.4.3)$$

$$F(t) = K[y(t - \tau) - y(t)] = K \cdot D(t) \text{ (laiko poslinkio tipas);} \quad (1.4.4)$$

$K > 0, \tau > 0$ – laisvai pasirenkamos skalės ir poslinkio konstantos. y_i – idealaus ciklo y koordinatė.

Idėja tokia – tipinėje chaotinė sistemoje egzistuoja daug labai panašių trajektorijų. Išskiriama viena iš jų kaip $y_i(t)$. Jei nemodifikuotos trajektorijos y koordinatė yra žemiau idealaus ciklo, tai $F(t)$ vertė bus teigiama, ir pagal (1.4.2), modifikuota $y'(t)$ vertė bus didesnė nei įprastai. Jei koordinatė yra aukščiau ciklo - $y'(t)$ bus mažesnė. Iteraciniame skaičiavimo modelyje, tai reikštu sekančiu žingsniu trajektorija priartės prie norimo ciklo. Aišku, reikia atsižvelgti kad K vertė nebūtų per didelė, nes tuomet vienu žingsniu galima smarkiai peršokti ciklą.

Logika yra tokia pati laiko poslinkio tipui, tik dabar tariama, kad norimas ciklas prasidėjo prieš τ laiko vienetų.



7 pav. Rossler sistemos trajektorijos vienoje plokštumoje

Išorinės jėgos tipui pavaizduoti, naudojama nestabili VDP sistema:

$$\begin{cases} x' = -y \\ y' = \mu(x^2 - 1)y + x. \\ x(0) = x_0, y(0) = 0 \end{cases} \quad (1.4.5)$$

Šiam pavyzdžiui imama $\mu = 2$, bei du variantai $x_0 \in \{1,8; 2,2\}$. Jei kalbama apie nemodifikuotą VDP sistemą, tai pirmam variantui, trajektorija artės prie pusiausvyros taško $(0,0)$, o antram variantui, trajektorija lėks toli nuo ciklo.

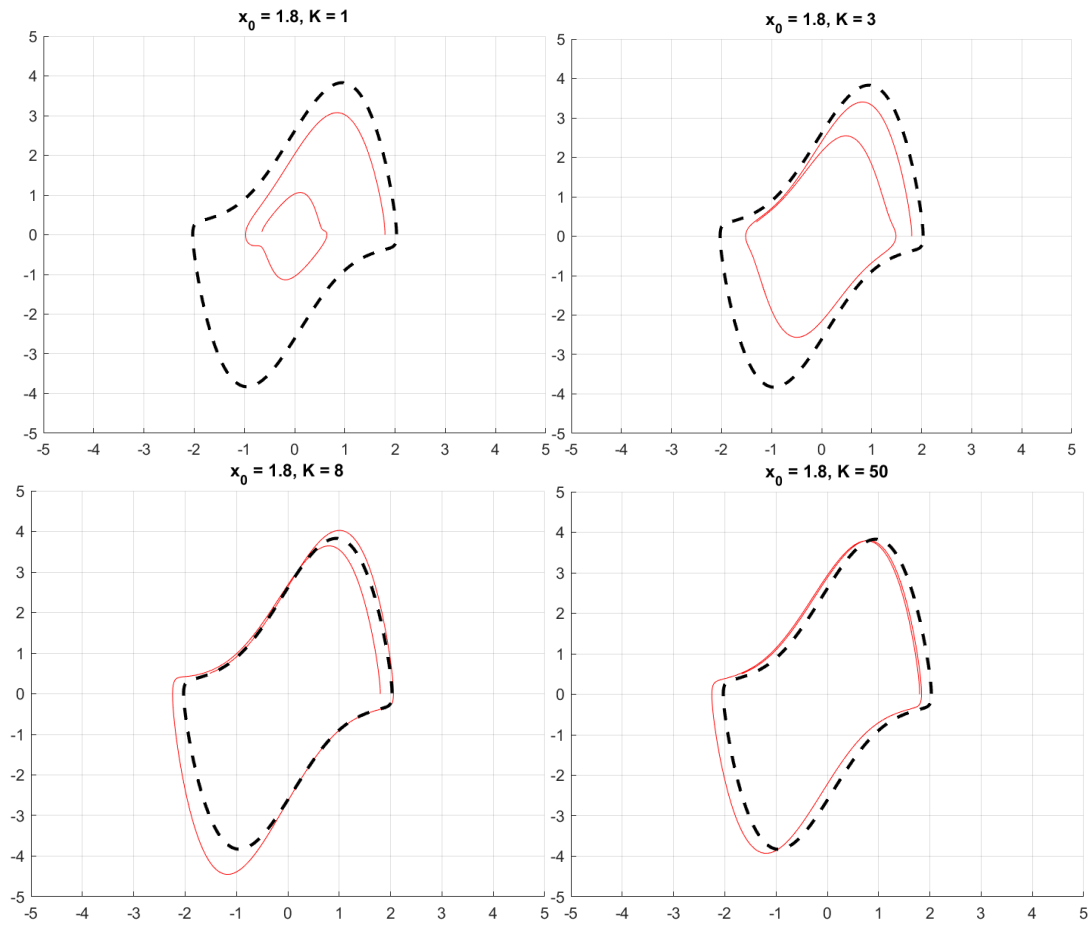
Modifikuojama y' lygtis:

$$y'(t) = \mu(x^2 - 1)y + x + F(t). \quad (1.4.6)$$

Grįžtamasis signalas F bus $F(t) = K[y_i(t) - y(t)]$, kur $y_i(t)$ – ribinio ciklo tikros y koordinatės. Tikslioms ciklo vėrtėms sugeneruoti ir (1.4.5) sistemos sukūrimui naudotas vienas triukas – laikas invertuojamas, ir peršokama nuo stabilios sistemos, į nestabilią ir atvirkščiai. Ribinio ciklo vėrtės yra invariacinės tokia transformacijai (nors orientacija – pasikečia).

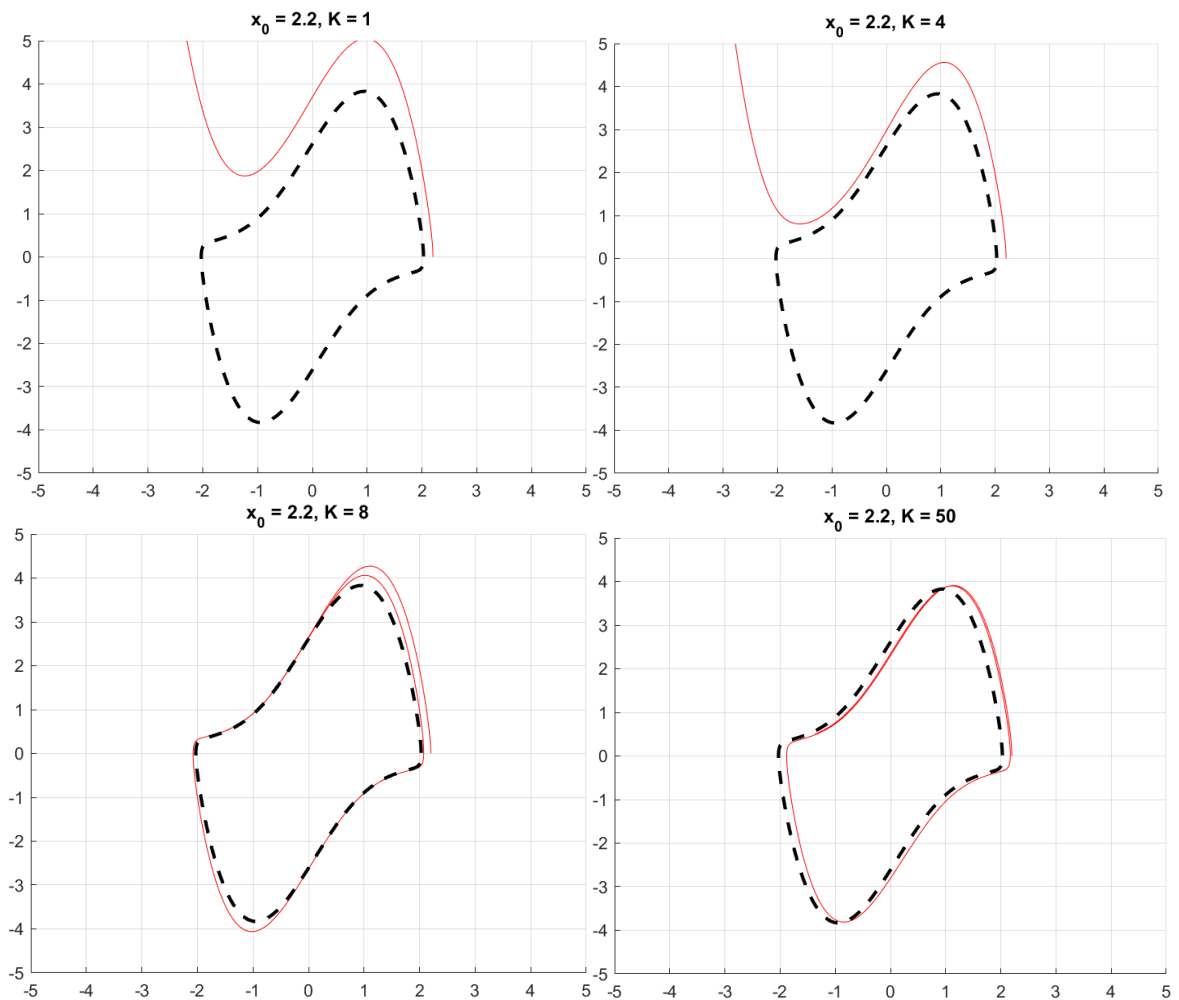
$$\begin{cases} x(t) = f(x(t), y(t)) \\ y(t) = g(x(t), y(t)); \\ x(0) = x_0, y(0) = y_0 \end{cases} \Leftrightarrow \begin{cases} x(t) = -f(x(t), y(t)) \\ y(t) = -g(x(t), y(t)). \\ x(0) = x_0, y(0) = y_0 \end{cases} \quad (1.4.7)$$

Išbandomos kelios K vertės abejiems variantams.



8 pav. Pirmo variantu trajektorija su skirtingomis K vertėmis

Jei pradinė koordinatė yra viduje ciklo, neatlikus jokių modifikacijų, trajektorija visada artės prie pusiausvyros taško $(0,0)$. Pakankamai padidinus K , trajektorija prisiglaudžia prie ribinio ciklo, bet toli į išorę nenulekia, visada liekanti arti kreivės.



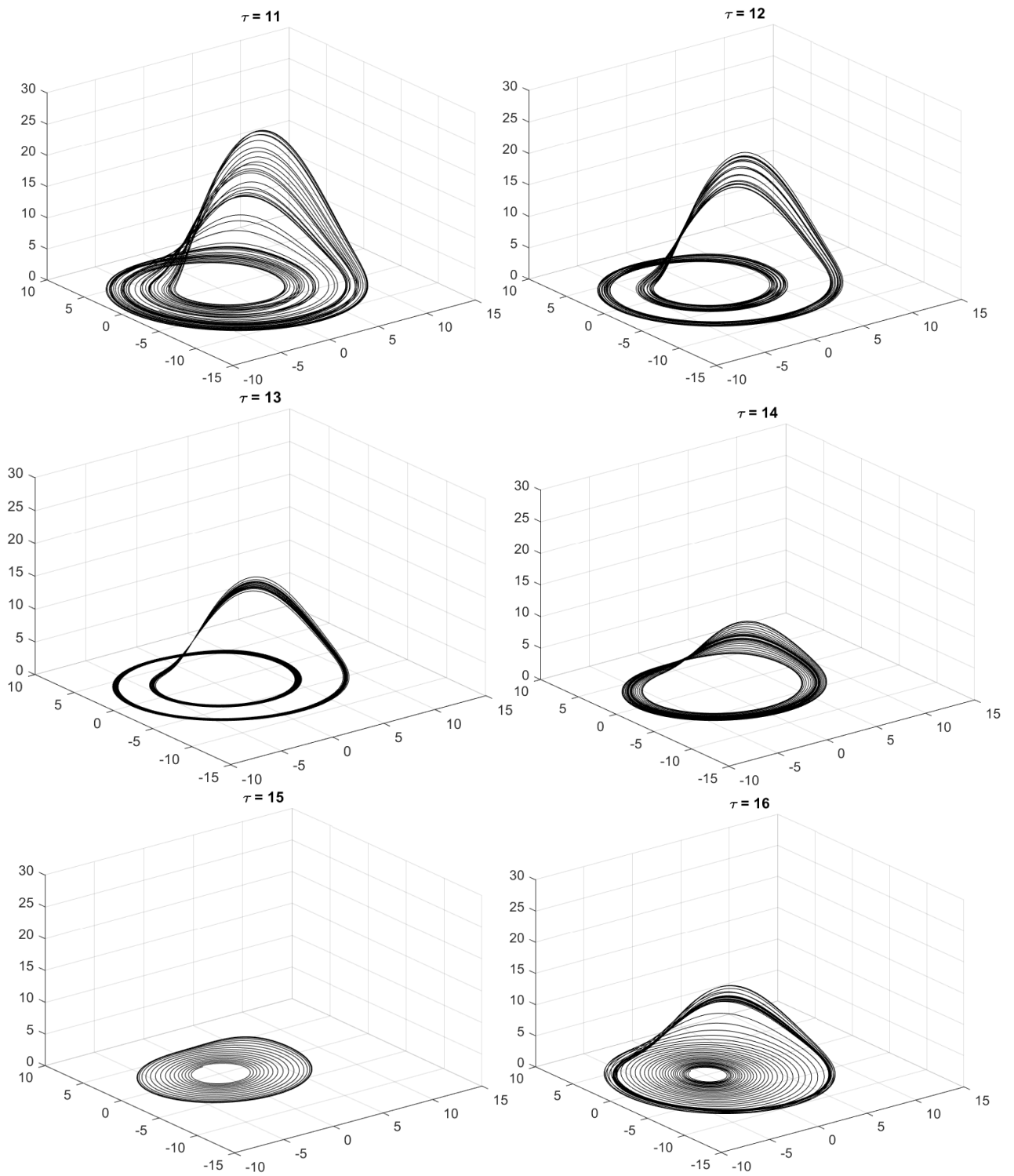
9 pav. Antro variantu trajektorija su skirtingomis K vertėms

Jei pradinė koordinatė yra viduje ciklo, nepridedant grįžtamojo ryšio, trajektorija visada be galo tols į begalybę. Pakankamai padidinus K , trajektorija prisiglaudžia prie ribinio ciklo, su mažais protrūkiais į ciklo vidų.

Nepriklausomai, ar pradinis taškas yra ciklo viduje, ar išorėje, su pakankamai didelę K verte galima trajektoriją priglausti prie ciklo kreivės. Tačiau, jei K per didelis, tai trajektorija užsifiksuos ties paslinkta ciklo kopija. Jei K viršija $1/\Delta t$, kur Δt – iteracijos žingsnio ilgis, tai gali susidaryti per didelės paklaidos, ir jokių naudingų rezultatų nebus.

Metodas yra gan paprastas, bet yra aiškus trūkumas – reikia iš anksto žinoti kaip atrodo sistemos nestabilus ribinis ciklas.

Toliau bandomas laiko poslinkio tipo grįžtamasis ryšys. Toliau pateikiami Rossler atraktoriaus grafikai su skirtingomis τ vertėmis ir fiksuota $K = 0,3$ verte.



10 pav. Rossler atraktorius kai $\tau \in [11; 16]$

Trajektorijos pradeda stabilizuotis ties $\tau \sim 12$, bei pilnai susitraukia ir vėl išsiskleidžia ties $\tau \sim 15$.

2. Duomenys ir tyrimų metodai

Dėl projekto prigimties, visi duomenys – savarankiškai sugeneruoti. Priede pateiktas kodas duomenų atkūrimui.

2.1. Naudojama programinė įranga

Visiems skaičiavimams naudota Matlab R2024a programa. Kai kuriems rezultatams panaudota Garrappa [35 p. 15] implementacija Matlab kalba [46], nes originalus straipsnio kodas – prarastas. Visais kitais atvejais naudoti savarankiškai parašyti algoritmai.

3. Tyrimų rezultatai ir jų aptarimas

Apžvelgiama kaip grįžtamasis ryšys paveikė visas tirtas sistemas. Tarpiniai ir galutiniai rezultatai pateikiami grafiškai. Pamenamos susidurtos problemos ar pastebėtos savybės.

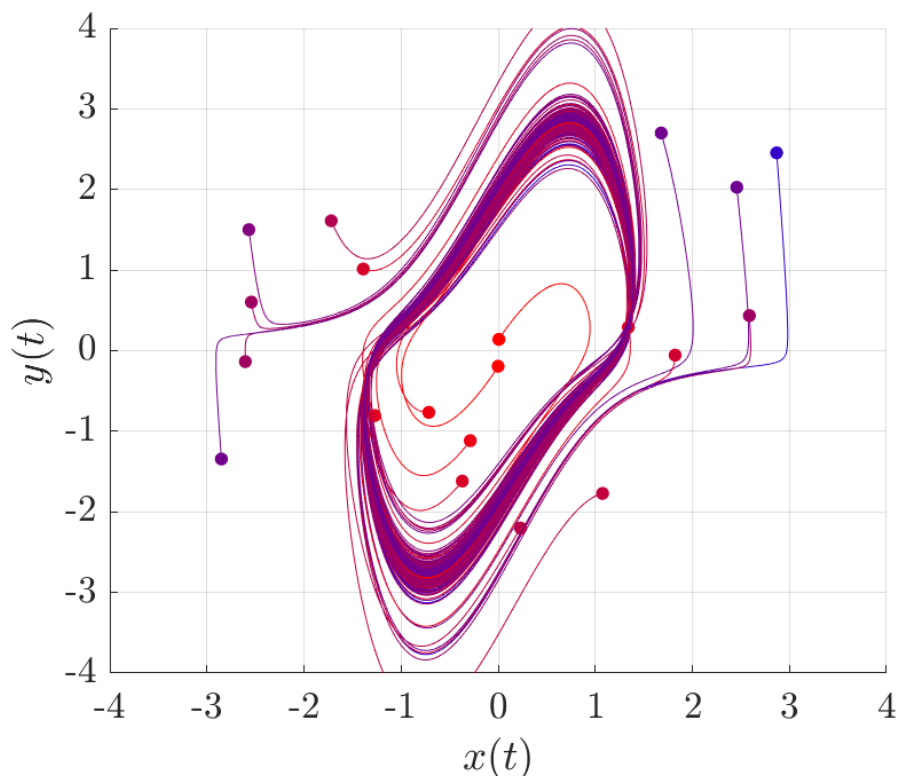
3.1. Trupmeninė Van der Pol sistema

Turime stabilią, trupmeninę VDP sistemą:

$$\begin{cases} D^{0.8}x = y \\ D^{0.8}y = \mu y(1 - x^2) - x. \\ x(0) = x_0, y(0) = y_0 \end{cases} \quad (3.1.1)$$

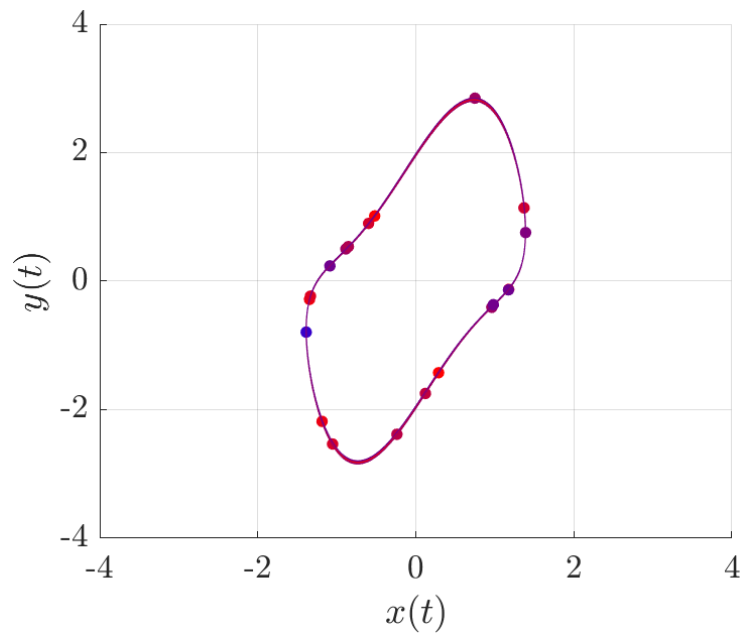
Laiko invertavimo triukas, kuris naudotas klasikinių sistemų stabilius ribinius ciklus paversti nestabiliais čia pilnu pajėgumu neveikia. Naudojamos trupmeninės išvestinės neigiamam laikui yra neapibrėžtos, bei sudaromos trajektorijos nėra tikri ribiniai ciklai, o labai lėtai gėstančios pseudo-periodinės trajektorijos. Tuoju galite įsitikinti atliekant tokį eksperimentą:

Grafiškai randame ribinį ciklą.

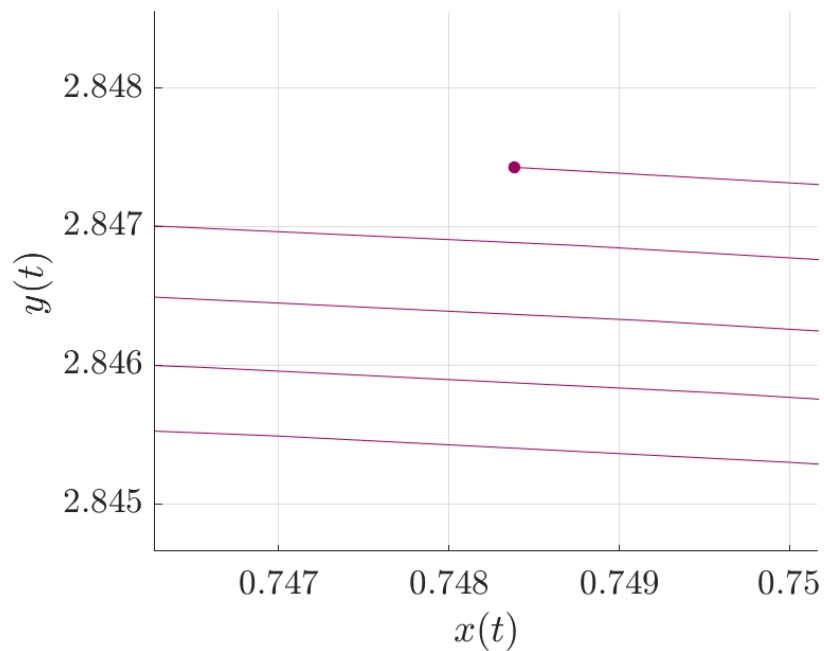


11 pav. Stabilios trupmeninės VDP sistemos fazinis vaizdas

Tuomet pavaizduojama tik dalis paskutinių iteracijų, ir priartinamas vaizdas



12 pav. Paskutinis 10% sistemos iteracijų

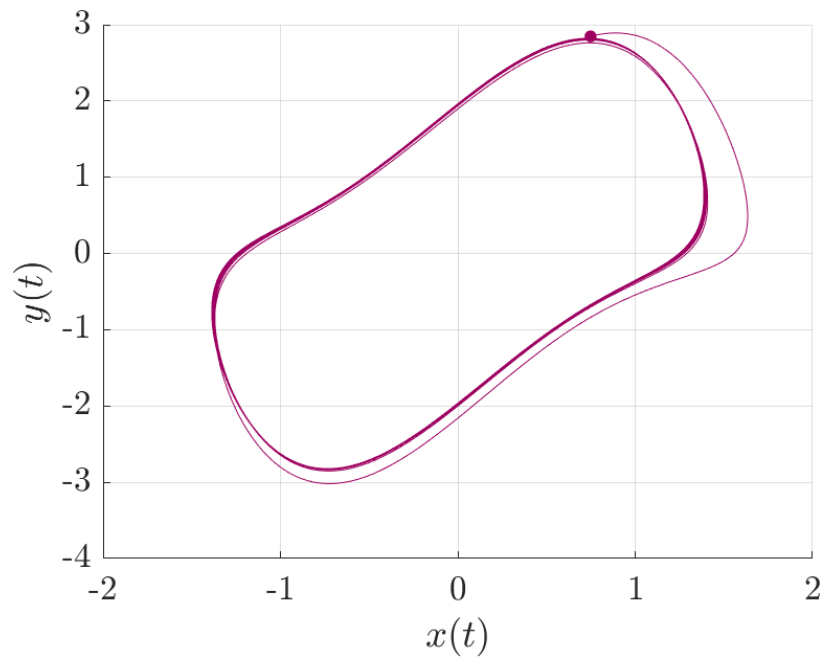


13 pav. Paskutinių sistemos iteracijų priartintas vaizdas

Matome, kad tai nėra tikra periodinė orbita – ciklas labai lėta nyksta. Toliau, kai kalbama apie ciklus trupmeninėse sistemose, turima omenyje pseudo-periodinius ciklus.

Nors turimos trajektorijos nėra tikros orbitos, skaitiniu atžvilgiu jos mažai skiriasi nuo tikrų, ir aptarti metodai vis tiek jiems gali būti taikomi.

Taip pat galima paimti vieną trajektoriją ir ilgam laiko intervalui integruoti. Šiame pavyzdyje integruota 5000 laiko vienetų.

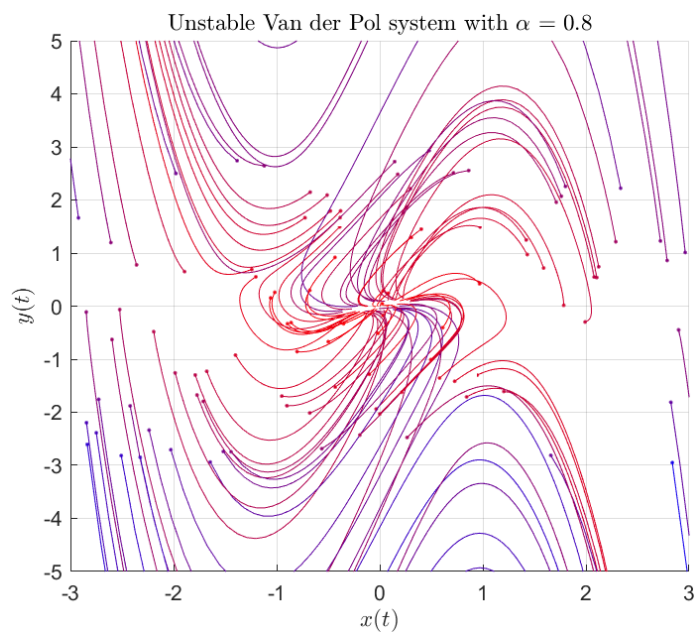


14 pav. Stabilios trupmeninės VDP sistemos pseudo-periodinis ribinis ciklas

Pereinama prie nestabilios VDP sistemos.

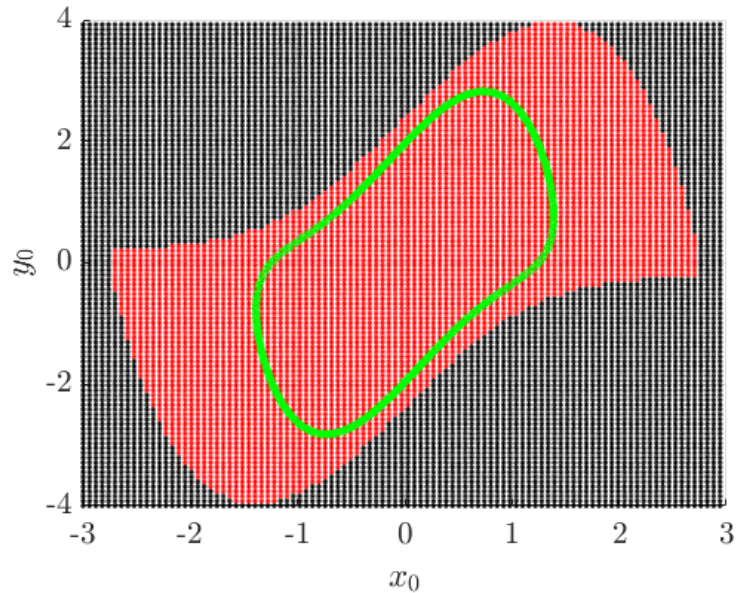
$$\begin{cases} D^{0.8}x = -y \\ D^{0.8}y = \mu y(x^2 - 1) + x. \\ x(0) = x_0, y(0) = y_0 \end{cases} \quad (3.1.2)$$

Nestabilius ciklus yra smarkiai sunkiau eksperimentiškai nustatyti, nei stabilius. Vietoj to, sugeneruojama daug trajektorijų su skirtingomis pradinėmis sąlygomis ir nubraižomas fazinis grafikas.



15 pav. Nestabilios trupmeninės VDP sistemos fazinis vaizdas

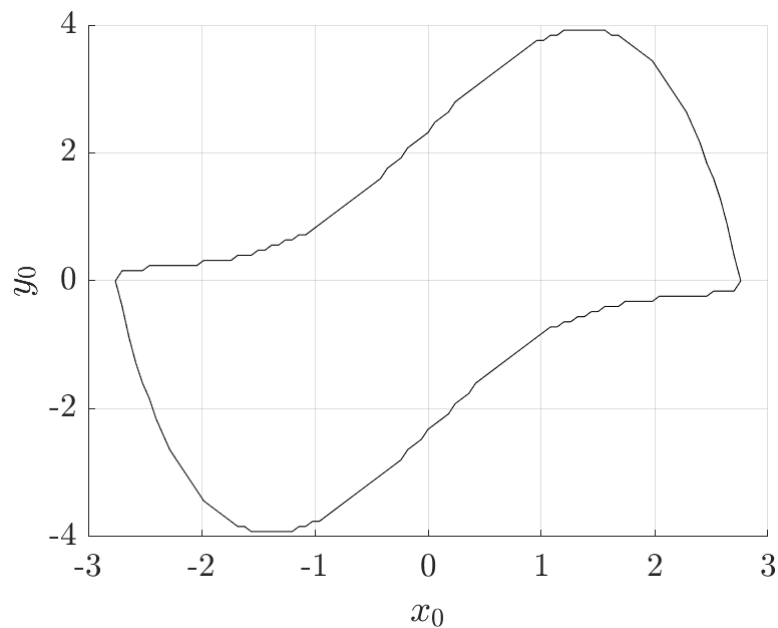
Iš šio vaizdo galima numatyti ribinę formą, kurios viduje visos trajektorijos artėja link pusiausvyros taško $(0,0)$, o visos išorėje atitolsta. Kadangi žinoma jų paskirties vieta, galima eksperimentiškai rasti vidinius taškus: sugeneruojama daug trajektorių, ir nuspalvinamas jų pradinis taškas raudonai jei po ilgo laiko galutinė trajektorijos vertė artima $(0,0)$. Palyginimui uždedama stabilaus ciklo kreivė žalia spalva.



16 pav. Nestabilios trupmeninės VDP sistemos pradinių taškų dinamika

Aišku, kad stabilaus ir nestabilaus ciklo kreivės nėra identiškos, ir reikia atskirai išskirti nestabilios kreivės taškus jei norima panaudoti 1.4 skyriuje naudotą grįžtamo ryšio metodą.

Išskiriant tik ribines vertes, kreivė atrodo taip:

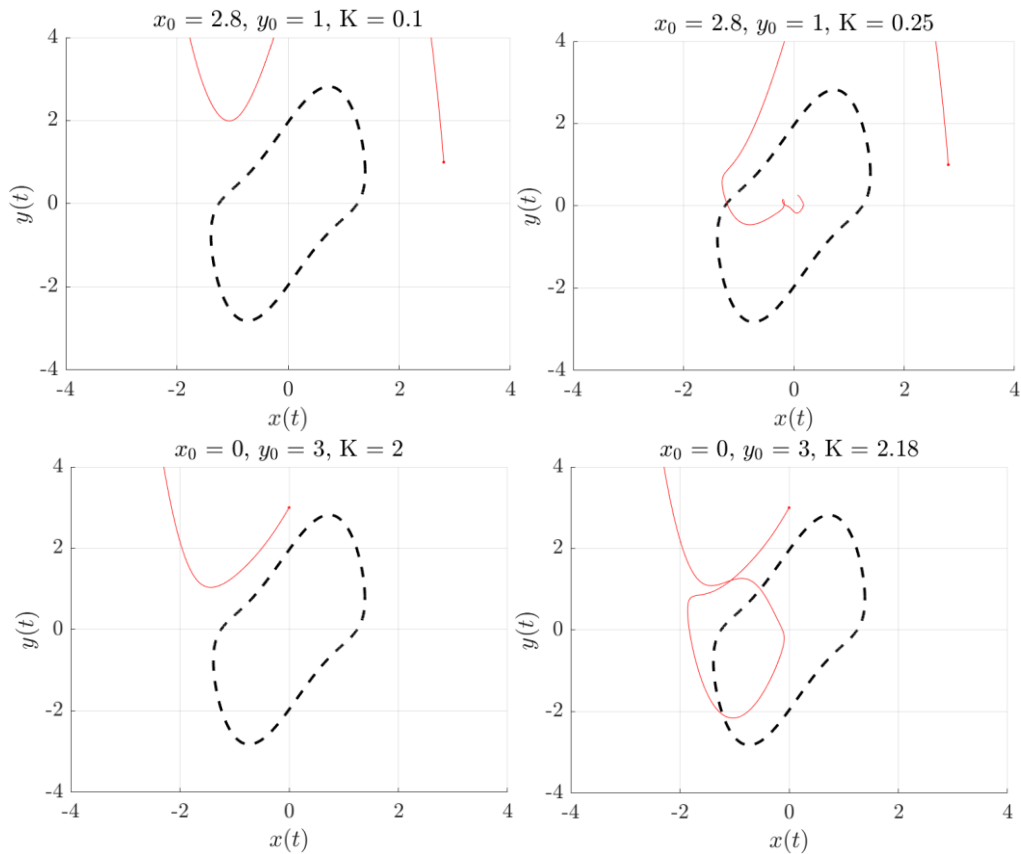


17 pav. Išskirta nestabilaus ribinio ciklo kreivė

Toks ciklo radimo metodas yra labai neefektyvus ir jam reikia labai ilgų skaičiavimų, tad negalima smarkiai pagerinti tikslumo.

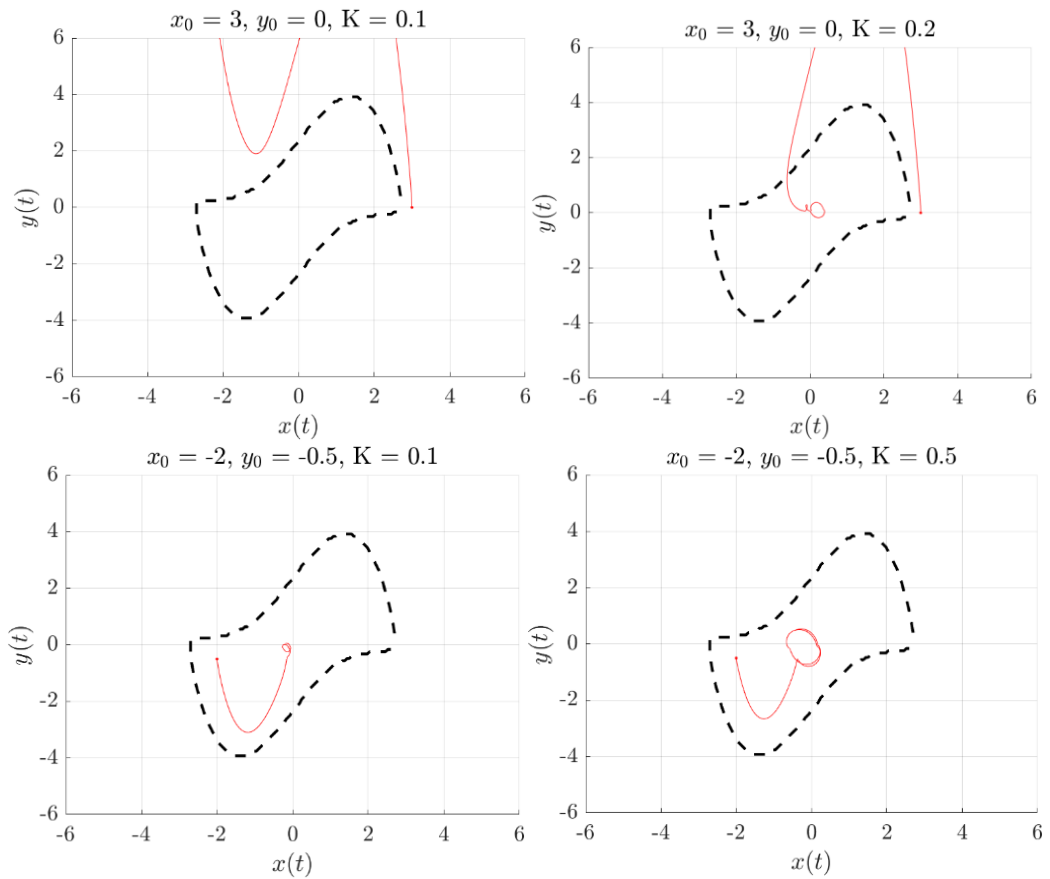
Verta paminėti esminį skirtumą tarp klasikinių ir trupmeninių sistemų ribinių ciklų – klasikinės sistemos trajektorijos prasidedančios viduje ciklo visada jame ir liks, o trupmeninės sistemos trajektorijos gali išeiti ir vėliau sugrįžti.

Pritaikomas grįžtamasis ryšys abejiems turimiems ciklams. Kaip ir anksčiau, išbandomos keletas skirtingų pradinėjų sąlygų ir K verčių.



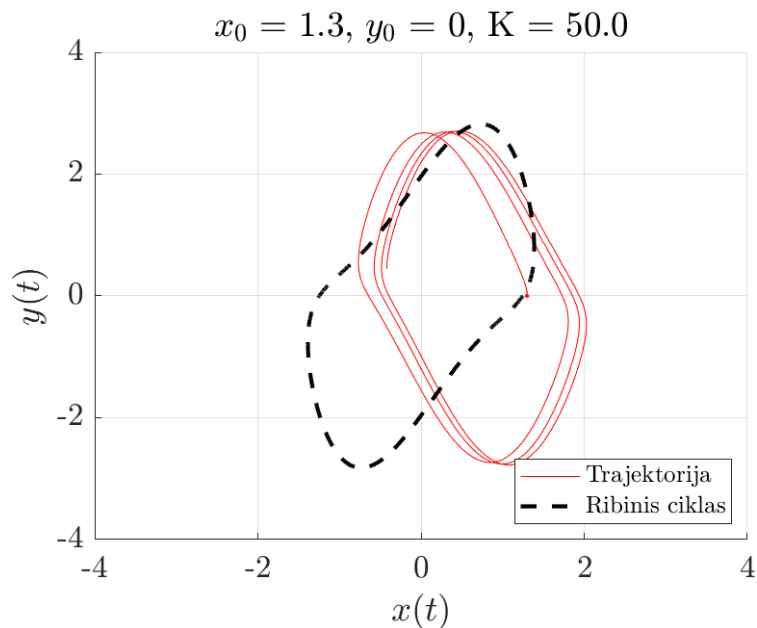
18 pav. Grįžtamo ryšio grafikai naudojant stabilų ribinį ciklą

Nustatant pakankamai didelę K vertę galima sustabdyti trajektorijos divergavimą, bet tik laikinai.



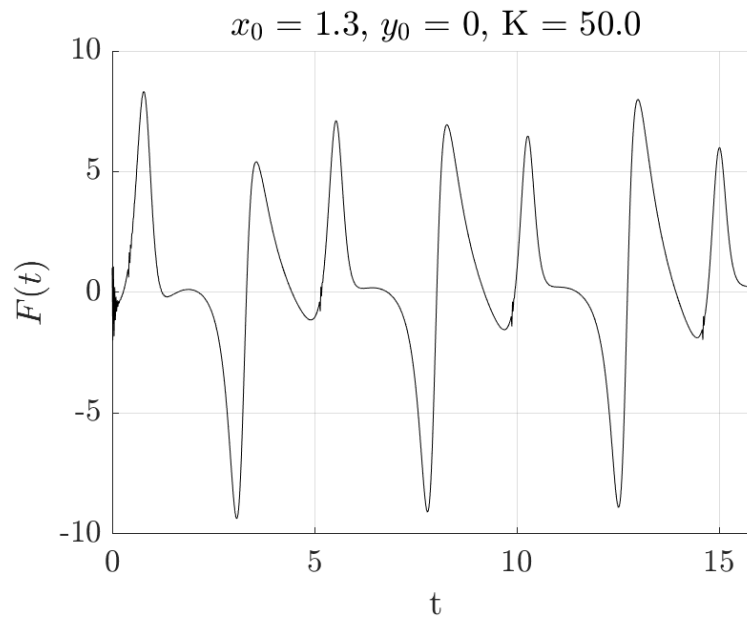
19 pav. Grįžtamo ryšio grafikai naudojant nestabilų ribinį ciklą

Rezultatai – ne tokie, kokių buvo tikimasi. Visais išbandytais variantais ir parametrais trajektorija arba nuklysta toli nuo ciklo, arba slenka link pusiausvyros taško. Yra ir įdomesnių atvejų – trajektorija gali priartėti prie pusiausvyros, bet tada apsisukti ir atitolti. Uždedant aukštą K vertę galima gauti periodines orbitas, bet neatitinkančias turimų ciklų.



20 pav. Sukurta pseudo-periodinė trajektorija

Geriausiu atveju trajektorijos laikinai priartinamos prie ciklo, bet neužsistovi. Dar pridėdamas grįžtamojo ryšio funkcijos $F(t) = K[y_i(t) - y(t)]$ grafikas.



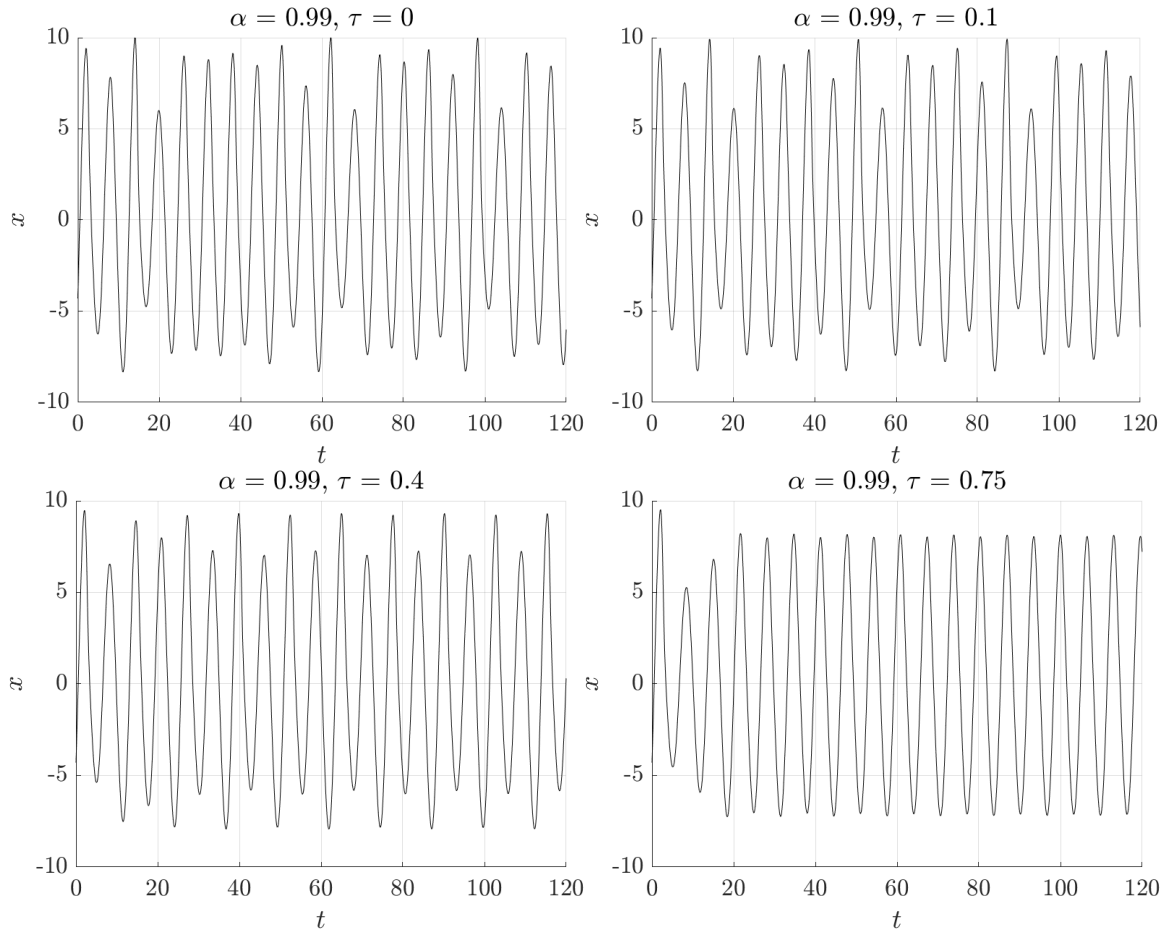
21 pav. Grįžtamojo ryšio funkcijos grafikas

Tai, $F(t)$ – irgi tampa pseudo-periodinė.

Gali būti, kad dėl trupmeninių išvestinių prigimties, pilnai trajektorijos kontrolei neužtenka modifikuoti tik paskutinį elementą, kai visos praeitos vertės turi įtaką. Ties tuo tenka ir užbaigti šio specifinio tipo grįžtamo ryšio analizę.

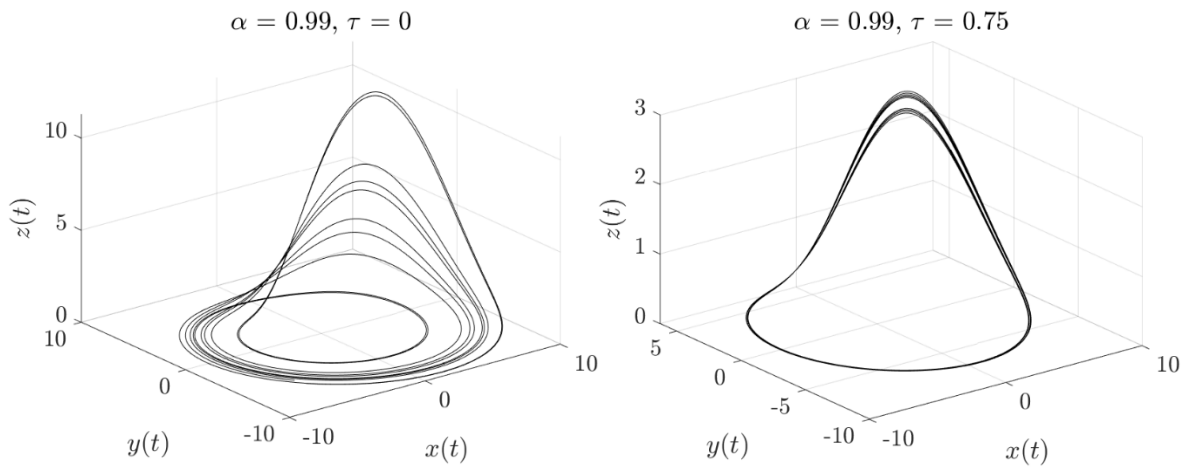
3.2. Trupmeninė Rossler sistema

Paimama (1.3.1) sistema su tais pačiais parametrais ir išvestinės eile $\alpha = 0,99$. Fiksuojama $K = 0,2$ ir didinama τ vertė, pradedant nuo nulio (nemodifikuotos sistemos). 3D vaizdai panašūs į 10 pav. ir nesuteikia pakankamai informacijos. Vietoj to, nubraižomos x koordinatės vertės ir atkreipiamas dėmesys į periodų skaičių.

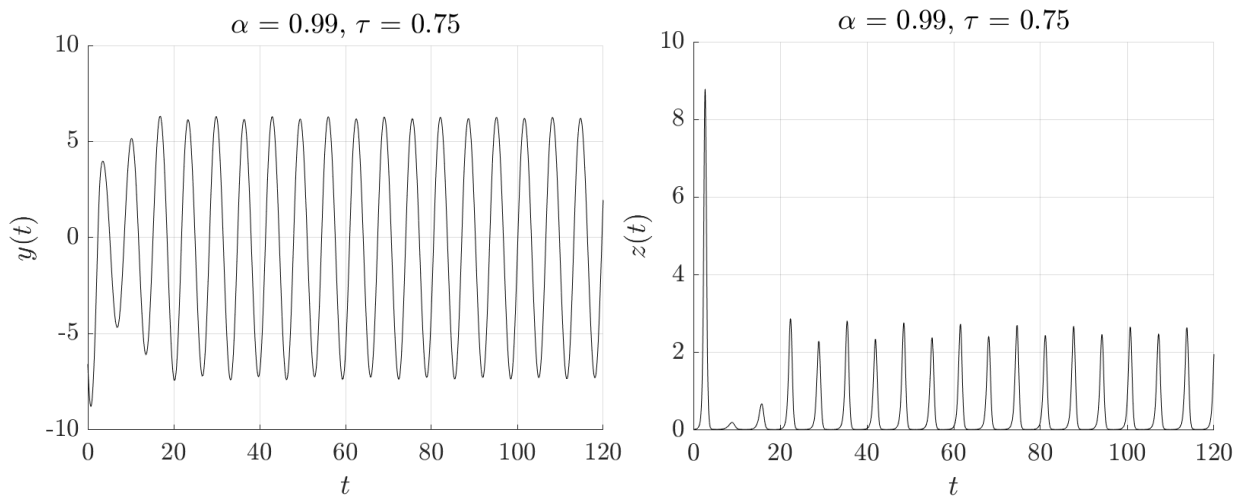


22 pav. Trajektorijų x koordinatės su keliomis τ vertėmis

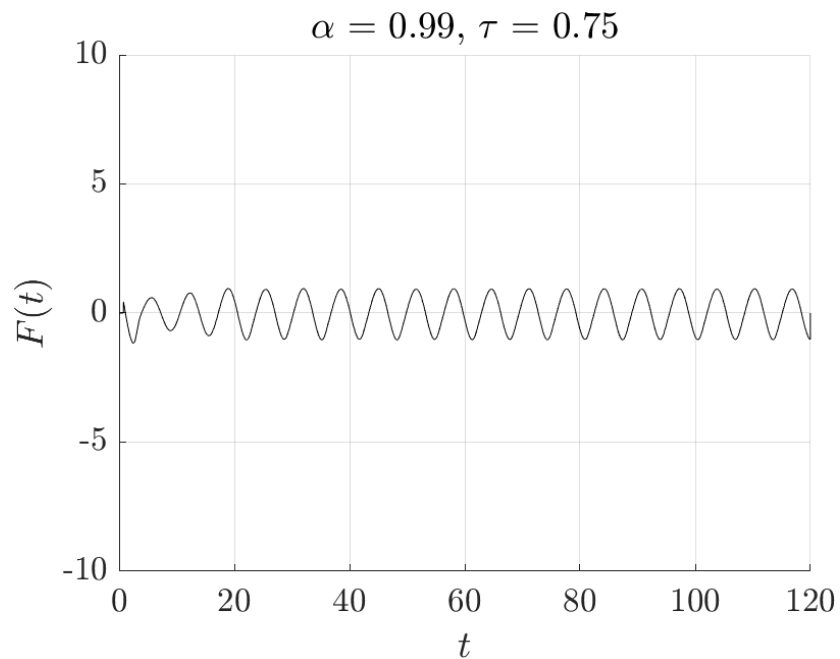
Kai $\tau = 0$, turime 1.3 skyriuje tirtą sistemą. Manoma, kad čia susidaro chaosas, arba bent labai aukšto periodų skaičiaus orbita. Padidinus τ iki 0,1 ir 0,4 periodų skaičius atitinkamai sumažėja iki 6 ir 2. Apie $\tau \sim 0,75$ du periodai susilygina ir tampa vienu.



23 pav. Originalios ir stabilizuotos trupmeninės Rossler sistemos 3D vaizdas.



24 pav. Stabilizuotos sistemos y ir z koordinatės



25 pav. Stabilizuotos sistemos grįžamojo ryšio funkcijos grafikas

Visos koordinatės tapo periodiškos, o grįžtamasis ryšis – reliatyviai smulkus. Galima teigti, jog sistema – sėkmingai stabilizuota.

Išvados

1. Klasikinėms sistemoms skirti metodai peržvelgti ir sėkmingai įgyvendinti. Grafiškai ir skaitiškai patvirtintas jų tikslumas.
2. Abu grįžtamojo ryšio metodo tipai pritaikyti trupmeninėms sistemoms, atkurti šaltiniuose pateikti pavyzdžiai.
3. Pavaizduotos klasikinių ir trupmeninių sistemų dinamikos, skirtumai tarp ribinių ciklų sampratos, išvestinės eilės įtaka.
4. Laiko poslinkio tipo grįžtamasis ryšys trupmeninių sistemų atžvilgiu veikė taip pat efektyviai, kaip ir klasikinių sistemų. Išorinės jėgos tipas suteikė mišrių rezultatų.

Literatūros sąrašas

1. Gottfried Wilhelm Leibniz (1665).
2. Joseph Liouville (1832). *Mémoire sur quelques questions de géométrie et de mécanique, et sur un nouveau genre de calcul pour résoudre ces questions.*
3. Oliver Heaviside (1892). *Electrical papers.*
4. Stephen W. Wheatcraft, Mark M. Meerschaert (2008). Fractional conservation of mass.
5. Abdon Atangana, P. D. Vermeulen (2014). Analytical Solutions of a Space-Time Fractional Derivative of Groundwater Flow Equation.
6. Lorenzo Santi, Gianni Royer-Carfagni (2024). Viscoelastic modeling via fractional calculus of the cold bending of laminated glass.
7. Isaac Y. Miranda-Valdez, Jesús G. Puente-Córdova, et al. (2024). Viscoelastic phenomena in methylcellulose aqueous systems: Application of fractional calculus.
8. Zhixin Shi, Jianqiu Zhou, et al. (2025). The viscoelastic stress wave propagation model based on fraction derivative constitutive.
9. Gustavo S. Matias, Fernando H. Lermen, et al. (2023). A model of distributed parameters for non-Fickian diffusion in grain drying based on the fractional calculus approach.
10. Ruohan Guo, Weixiang Shen (2022). Online state of charge and state of power co-estimation of lithium-ion batteries based on fractional-order calculus and model predictive control theory.
11. Sunday Oluwafemi Gbodogbe (2025). Harmonizing epidemic dynamics: A fractional calculus approach to optimal control strategies for cholera transmission.
12. Shaher Momani, Iqbal M. Batiha, et al. (2025). On finite-time stability of some COVID-19 models using fractional discrete calculus.
13. Dumitru Baleanu, Parisa Shekari, et al. (2023). Stability analysis and system properties of Nipah virus transmission: A fractional calculus case study.
14. Changdev P. Jadhav, Tanisha B. Dale, et al. (2024). Some problems of mesh R-L electrical circuits using fractional calculus.
15. Marcin Sowa, Lukasz Majka, Klaudia Wajda (2023). Excitation system voltage regulator modeling with the use of fractional calculus.
16. Mi Zhou, Hongyun Huang, et al. (2024). Fractional order calculus model-derived histogram metrics for assessing pathological complete response to neoadjuvant chemotherapy in locally advanced rectal cancer.
17. Muhammad Usman, Oluwale Daniel Makinde, et al. (2023). Applications of fractional calculus to thermodynamics analysis of hydromagnetic convection in a channel.
18. Qiao Zhang, Yuxin Sun, Jialing Yang (2021). Bio-heat transfer analysis based on fractional derivative and memory-dependent derivative heat conduction models.
19. Noor Ahmad Mohammad, Ayesha Rashed Saif Rashed Alsalmi, et al. (2025). Application of fractional derivative in Buongiorno's model for enhanced fluid flow and heat transfer analysis over a permeable cylinder.
20. Ali Haider, M.S. Anwar, et al. (2025). Numerical simulations of heat and mass transfer in Sutterby fluid within porous media using Caputo fractional derivative.
21. Z. Zaabouli, L. Afraites, A. Laghrib (2025). A novel time-fractional decomposition model for image denoising integrating Caputo derivative.

22. Sugandha Arora, Trilok Mathur, et al. (2022). Applications of fractional calculus in computer vision: A survey.
23. Marwan Al-Raeei (2021). Applying fractional quantum mechanics to systems with electrical screening effects.
24. Mohammad (2021). Fractional paradigms in quantum chemistry.
25. Yihong Wang, Changpin Li (2006). Does the fractional Brusselator with efficient dimension less than 1 have a limit cycle?
26. Guoqi Zhang, Zhiqiang Wu (2020). Approximate limit cycles of coupled nonlinear oscillators with fractional derivatives.
27. Celaleddin Yeroglu, Nusret Tan (2010). Limit Cycle Prediction for Fractional Order Systems with Static Nonlinearities.
28. Bernhard Riemann (1896). Versuch einer allgemeinen Auffassung der integration und differentiation.
29. Michele Caputo (1967). Linear model of dissipation whose Q is almost frequency independent. II.
30. B. R. Sontakke, A. S. Shaikh (2015). Properties of Caputo Operator and Its Applications to Linear Fractional Differential Equations.
31. Beata Sikora (2023). Remarks on the Caputo fractional derivative.
32. Gavriil Shchedrin, Nathanael C. Smith, et al. (2018). Exact results for a fractional derivative of elementary functions.
33. Min Chai, Changpin Li (2019). Numerical Approaches to Fractional Integrals and Derivatives: A Review.
34. Yuri Dimitrov, Slavi Georgiev, Venelin Todorov (2023). Approximation of Caputo Fractional Derivative and Numerical Solutions of Fractional Differential Equations.
35. Roberto Garrappa (2017). Numerical Solution of Fractional Differential Equations: A Survey and a Software Tutorial.
36. Mudasir Younis, Haroon Ahmad, et al. (2025). Unveiling fractional-order dynamics: A new method for analyzing Rossler Chaos.
37. Muhammad Farman, Kottakkaran Sooppy Nisar, et al. (2025). Chaos and proportional integral derivative (PID) control on cancer dynamics with fraction fractional operator.
38. Mark Edelman (2024). Periodic Points, Stability, Bifurcations, and Transition to Chaos in Generalized Fractional Maps.
39. Manisha Krishna Naik, Chandrali Baishya, P. Veerasha (2023). A chaos control strategy for the fractional 3D Lotka-Volterra like attractor.
40. Chandrali Baishya, R. N. Premakumari, et al. (2023). Chaos control of fractional order nonlinear Block equation by utilizing sliding mode controller.
41. Otto Rossler (1976). An Equation for Continuous Chaos.
42. Otto Rossler (1979). An Equation for Hyperchaos.
43. Eric W. Weisstein (2021). Attractor.
44. Kęstutis Pyragas (1992). Continuous control of chaos by self-controlling feedback.
45. Steven Henry Strogatz (1994). Nonlinear Dynamics and Chaos.
46. Matlab Central. Predictor-corrector for FDEs.

Priedai

1 priedas. Programos kodas

```
% -----
% Martas Ulozevicius
% KTU Magistro projektas 2025
% -----
clear
clc

%% -----
% Bendri duomenys
% -----
vdp_stable = @(t, y, par) [ ...
    y(2); ...
    par(1)*y(2)*(1 - y(1)^2) - y(1)];

vdp_unstable = @(t, y, par) [ ...
    -y(2); ...
    par(1)*y(2)*(y(1)^2 - 1) + y(1)];

mu = 2;
param = mu;

%% -----
% VDP stabilaus ribinio ciklo su mu = 2, alpha = 0.8 generavimas
% -----
t0 = 0;           % Pradinis laikas
tn = 64;          % Galutinis laikas
y0 = [1.25; 0];   % Pradines koordinates
h = 2^(-9);       % Zingsnio dydis
alpha = 0.8;      % Ivestiniu eile

[t, y] = fde12(alpha, vdp_stable, t0, tn, y0, h, param);

% Rankiniu budu nustatytas vienas pilnas periodas tarp
% 28569 ir 30987 indeksu
vdpx = y(1,28569:30987);
vdpy = y(2,28569:30987);

% Invertuojama orientacija
vdpx = flip(vdpx);
vdpy = flip(vdpy);

vdpinfo.mu = mu;
vdpinfo.stepsize = h;
vdpinfo.alpha = alpha;
vdpinfo.x0 = y0(1);
vdpinfo.y0 = y0(2);
save('vdpvals.mat','vdpinfo','vdpx','vdpy')

figure(1)
plot(vdpx, vdpy, '--k')
```

```

%% -----
% VDP nestabilaus ribinio ciklo su  $\mu = 2$ ,  $\alpha = 0.8$  generavimas
% -----
t0 = 0;
tn = 3;
h = 2^(-5);
alpha = 0.8;
tlen = (tn - t0)/h + 1;

xlim = [-3 3]; % Skaiciavimo srities x ribos
ylim = [-4 4]; % Skaiciavimo srities y ribos
N = 100;       % Tasku skaicius vienai koordinatei, bendras sk. bus N^2

xh = (xlim(2) - xlim(1)) / N;
yh = (ylim(2) - ylim(1)) / N;
xpts = xlim(1):xh:xlim(2);
ypts = ylim(1):yh:ylim(2);
xlen = length(xpts);
ylen = length(ypts);

epsilon = 0.5;
c = zeros(xlen, ylen);

% Jei trajektorija arti (0,0), tai to pradinio tasko c verte yra 1
for i = 1:N
    for j = 1:N
        y0 = [xpts(i); ypts(j)];
        [~, y] = fde12(alpha, vdp_unstable, t0, tn, y0, h, param);
        r = y(1,tlen)^2 + y(2,tlen)^2;
        if r < epsilon
            c(i, j) = 1;
        end
    end % j
end % i

cc = flip(transpose(c));
yvals_top = zeros(1, ylen); % Virsutines kreives vertes
yvals_bot = zeros(1, ylen); % Apatines kreives vertes

for j = 1:ylen
    for i = 1:xlen
        if cc(i,j) == 1
            yvals_top(j) = ypts(xlen - i + 1);
            break;
        end
    end
end

for j = 1:ylen
    for i = 1:xlen
        if cc(i,ylen - j + 1) == 1
            yvals_bot(j) = ypts(i);
            break;
        end
    end
end

figure(2)
hold on
grid on
plot(xpts, yvals_top, 'k')
plot(xpts, yvals_bot, 'k')

```

```

%% -----
% VDP sistemos griztamasis rysis (isorine jega)
% -----
h = 1/64;
TotalTime = 16;
N = TotalTime/h;

vdplen = length(vdpx);

K = 20;

x = zeros(1, N);
y = zeros(1, N);
X = zeros(1, N);
Y = zeros(1, N);

% Pradines koordinates
x(1) = 0;
y(1) = 3;

alpha = 0.8;
Ca = h^(-alpha) / gamma(2 - alpha);
Cainv = 1/Ca;
repeater = round(h/vdpinfo.stepsize); % Kiek kartu laiko singsnis didesnis uz duomenu

[~, pti] = min((vdpx - x(1)).^2 + (vdpy - y(1)).^2); % Artimiausias taskas ties duomenu
kreive
X(1) = vdpx(pti);
Y(1) = vdpy(pti);

Y(2) = vdpy(mod(pti + repeater - 1, vdplen) + 1);
x(2) = Cainv * f(x(1), y(1)) + x(1);
y(2) = Cainv * ( g(x(1), y(1)) + K*(Y(1) - y(1)) ) + y(1);

for k = 3:N
    Y(k) = vdpy(mod(pti + repeater*k - 1, vdplen) + 1);
    x(k) = Cainv*f(2*x(k-1) - x(k-2), 2*y(k-1) - y(k-2));
    y(k) = Cainv*( g(2*x(k-1) - x(k-2), 2*y(k-1) - y(k-2)) + K*(Y(k-1) - y(k-1)) );
    for i = 2:k-1
        theta_val = theta(k - i, alpha);
        x(k) = x(k) - x(i)*theta_val;
        y(k) = y(k) - y(i)*theta_val;
    end
    lambda_val = lambda(k - 1, alpha);
    x(k) = x(k) - x(1)*lambda_val;
    y(k) = y(k) - y(1)*lambda_val;
end

figure(3)
hold on
grid on
axis([-4 4 -4 4])
plot(x, y, 'r')
plot(vdpx, vdpy, '--k', 'LineWidth', 2)

```

```

%% -----
% Rossler sistemos griztamasis rysis (laiko poslinkis)
% -----
a = 0.2;
b = 0.2;
c = 5.7;
alpha = 0.99;

K = 0.2;
tau = 0.75;

dt = 1/64;
TotalTime = 120;
N = TotalTime/dt;
shift = ceil(tau/dt); % Laiko poslinkis konvertuotas indeksams

x = zeros(1, N);
y = zeros(1, N);
z = zeros(1, N);

% Pradines koordinates
x(1) = -4.295;
y(1) = -6.583;
z(1) = 0.0192;

Ca = dt^(-alpha) / gamma(2 - alpha);
Cainv = 1/Ca;

x(2) = Cainv * fx(x(1), y(1), z(1)) + x(1);
y(2) = Cainv * fy(x(1), y(1), z(1)) + y(1);
z(2) = Cainv * fz(x(1), y(1), z(1)) + z(1);

for k = 3:N
    x(k) = Cainv* fx(2*x(k-1) - x(k-2), 2*y(k-1) - y(k-2), 2*z(k-1) - z(k-2));
    y(k) = Cainv*( fy(2*x(k-1) - x(k-2), 2*y(k-1) - y(k-2), 2*z(k-1) - z(k-2)) + F(y,
K, i, shift));
    z(k) = Cainv* fz(2*x(k-1) - x(k-2), 2*y(k-1) - y(k-2), 2*z(k-1) - z(k-2));
    for i = 2:k-1
        theta_val = theta(k - i, alpha);
        x(k) = x(k) - x(i)*theta_val;
        y(k) = y(k) - y(i)*theta_val;
        z(k) = z(k) - z(i)*theta_val;
    end
    lambda_val = lambda(k - 1, alpha);
    x(k) = x(k) - x(1)*lambda_val;
    y(k) = y(k) - y(1)*lambda_val;
    z(k) = z(k) - z(1)*lambda_val;
end

len = length(x);
ign = N/2;

figure(4)
hold on
grid on
plot3(x(ign:len), y(ign:len), z(ign:len), 'k')
view(3)

```

```

%% -----
% Pagalbines funkcijos
% -----
function res = f(~, y)
    res = -y;
end

function res = g(x, y)
    res = 2*y*(x*x - 1) + x;
end

function res = theta(k, a)
    res = (k - 1)^(1 - a) - 2*k^(1 - a) + (k + 1)^(1 - a);
end

function res = lambda(n, a)
    res = (n - 1)^(1 - a) - n^(1 - a);
end

function res = F(y, K, i, shift)
    if i <= shift
        res = 0;
    else
        y1 = y(i - shift);
        y2 = y(i);
        res = K*(y1 - y2);
    end
end

function res = fx(~, y, z)
    res = -y - z;
end

function res = fy(x, y, ~)
    res = x + 0.2*y;
end

function res = fz(x, ~, z)
    res = 0.2 + z*(x - 5.7);
end

```