



**KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS**

Tadas Telksnys

**ANTROS EILĖS SOLITONINIŲ SPRENDINIŲ EGZISTAVIMAS
MULTIPLIKATYVIUOJU SĄRYŠIU SUSIETOSE RIKATI
LYGTYSE**

Baigiamasis magistro projektas

Vadovas

prof. habil. dr. Minvydas Ragulskis

KAUNAS, 2015

KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS

**ANTROS EILĖS SOLITONINIŲ SPRENDINIŲ EGZISTAVIMAS
MULTIPLIKATYVIUOJU SĄRYŠIU SUSIETOSE RIKATI
LYGTYSE**

Baigiamasis magistro projektas
Taikomoji matematika (kodas 621G1003)

Vadovas

prof. habil. dr. M. Ragulskis

Recenzentas

prof. dr. J. Rimas

Projektą atliko

Tadas Telksnys

KAUNAS, 2015

Turinys

LENTELIŲ SAŖAŠAS	5
PAVEIKSLŲ SAŖAŠAS	6
SUMMARY	7
IŽANGA	8
1 LITERATŪROS APŽVALGA	9
1.1 Solitonai	9
1.2 Tikslių diferencialinių lygčių sprendinių paieškos metodų apžvalga	12
1.2.1 Eksponentinių funkcijų metodas	13
1.2.2 Hiperbolinio tangento metodas	14
1.2.3 (G'/G) skleidinio metodas	14
1.2.4 Kiti metodai	15
2 MEDŽIAGOS IR TYRIMO METODAI	15
2.1 Operatorinio metodo taikymas paprastosioms diferencialinėms lygtims	15
2.1.1 Formalių eilučių algebra ir tiesiniai operatoriai	15
2.1.2 Funkcijų klasės	16
2.1.3 Apibendrintasis diferencialinis operatorius	18
2.1.4 Paprastųjų diferencialinių lygčių sprendinių operatorinės išraiškos	18
2.1.5 Tiesiškai rekurentinės sekos	19
2.1.6 Tiesiškai rekurentinių sekų panaudojimas solitoninių sprendinių konstravimui	21
2.1.7 Nesusietos Rikati lygties sprendimas	22
2.2 Simbolinių pertvarkymų programinės įrangos paketai	24
3 TYRIMO REZULTATAI	25
3.1 Atvirkštinio balansavimo metodas	25
3.1.1 Antros eilės solitonai Rikati lygčių sistemoje	25
3.1.2 Formalūs sąryšiai tarp solitoninio sprendinio ir Rikati sistemos parametrų	28
3.1.3 Solitoninių sprendinių fazinės trajektorijos	29
3.2 Operatorinis metodas	30
3.2.1 Diferencialinių lygčių sistemos sprendinių operatorinės išraiškos	30
3.2.2 Specialiosios Hankelio determinantų išraiškos	31
3.2.3 Antros eilės solitoninio sprendinio koeficientų sekų kanoninių išraiškų konstravimas	34
3.2.4 Uždara antros eilės solitoninio sprendinio forma	35
3.3 Solitoninių sprendinių dinamika	37
3.3.1 Pusiausvyros taškai ir stabilumas	37

3.3.2	Trys pusiausvyros taškai	39
3.3.3	Du pusiausvyros taškai	40
4	PROGRAMINĖ REALIZACIJA	42
4.1	Darbe naudojamų funkcijų aprašymai	42
4.2	Darbui parašytų darbalapių (programų) aprašymai	43
5	REZULTATŲ APTARIMAS	44
5.1	Tyrimo apžvalga	44
5.2	Analizinės sprendinio formos pranašumai	44
	IŠVADOS	45
	REKOMENDACIJOS	45
	PADĖKA	46
	LITERATŪRA	47
	PRIEDAI	51
	Priedas A Žymėjimų sąrašas	51
	Priedas B Apibendrintųjų diferencialinių operatorių lentelė	52
	Priedas C Programų išeities tekstai	53
C.1	Programa P11_1. Pav. 1.1 ir 1.2 braižymas	53
C.2	Programa P11_1. Pav. 1.3 ir 1.4 braižymas	53
C.3	Programa P137. (2.1.7 poskyris)	54
C.4	Programa P211. (3.1.1 poskyris)	55
C.5	Programa P213. (3.1.3 poskyris)	56
C.6	Programa P222. (3.2.2 poskyris)	57
C.7	Programa P223. (3.2.3 poskyris)	58
C.8	Programa P231. (3.3.1 poskyris)	59
C.9	Programa P232. (3.3.2 poskyris)	60
C.10	Programa P233. (3.3.3 poskyris)	62

LENTELIŲ SĄRAŠAS

B.1	Diferencialinių lygčių sprendinių operatorinės išraiškos	52
-----	--	----

PAVEIKSLŲ SĄRAŠAS

1.1	KdV ir sine-Gordono lygties solitonai.	10
1.2	Šriodingerio lygties solitonas	11
1.3	KdV ir sine-Gordono lygties solitonų priklausomybė nuo banginio kintamojo.	12
1.4	Šriodingerio lygties solitono priklausomybė nuo banginio kintamojo.	12
3.1	3.3.2 poskyrio (3.94) sistemos fazinis portretas	38
3.2	Sistemos (3.94) sprendinių evoliucija kintamojo x atžvilgiu	40
3.3	3.3.3 poskyrio (3.98) sistemos fazinis portretas	41
3.4	Sistemos (3.98) sprendinių evoliucija kintamojo x atžvilgiu	42

Telksnys T. Existence of second order solitary solutions to Riccati differential equations coupled with a multiplicative term. *Master's project in applied mathematics* / supervisor d. habil. prof. M. Ragulskis; Kaunas university of technology, Faculty of mathematics and natural sciences, Department of mathematical modeling.

Kaunas, 2015. 64 p.

SUMMARY

Necessary and sufficient conditions for the existence of second order solitary solutions to Riccati equations coupled with multiplicative terms are derived in this thesis. Necessary conditions are obtained using the inverse balancing method, which generates the system parameters when the solution parameters are known. Sufficient conditions are derived using the generalized differential operator method. By utilizing the theory of linear recurring sequences the functional relationship between the solution parameters and initial conditions is obtained. It is demonstrated that when the parameters of the Riccati system satisfy the existence conditions for the second order solitary solution, all nonspecial solutions of the system are second order solitary solutions. The phase trajectories of these solutions are conic sections. The dynamical properties of the Riccati system are illustrated using computational examples. The computer algebra system Maple is used to derive the symbolic results of the thesis.

IŽANGA

Biologijoje dažnai sutinkamos ekosistemos, aprašomos susietųjų diferencialinių lygčių sistemomis. Priešingai nei klasikinėje dinaminų sistemų sinchronizacijos teorijoje, šias sistemas sieja ne difuzinis ($\alpha(y - z)$) ar harmoninis ($\alpha \sin(y - z)$) ryšys, bet multiplikatyvusis sąryšis αyz . Multiplikatyviuoju sąryšiu pasižymi biologinės sistemos, kuriose populiacijos dydis ar tankis ir jos augimo tempas yra susiję. Ši sąsaja vadinama Ali¹ efektu [1], [2], [3].

Ali efektu pasižymi ir [4] nagrinėjama paprastųjų diferencialinių lygčių sistema:

$$\begin{aligned} y''_x + \alpha y'_x &= a_0 + a_1 y + a_2 y^2 + a_3 y^3 + a_4 yz; \\ z''_x + \alpha z'_x &= b_0 + b_1 z + b_2 z^2 + b_3 z^3 + b_4 yz, \end{aligned} \quad (0.1)$$

čia $\alpha, a_k, b_k \in \mathbb{R}, k = 0, \dots, 4, a_3, b_3 \neq 0$. Ši sistema aprašo aukos-grobuonies dinamiką, kuomet aukos ir grobuonies natūralūs augimo tempai yra lygūs, o biomasės susikūrimo tempas atitinka jos asimiliacijos tempą [5].

[4] parodoma, kad (0.1) tenkina funkcijos:

$$y(x) = \sigma \frac{e^{\eta(x-c)} - y_1}{e^{\eta(x-c)} - x_1}, \quad z(x) = \gamma \frac{e^{\eta(x-c)} - z_1}{e^{\eta(x-c)} - x_1}; \quad (0.2)$$

su $\eta, \sigma, \gamma, c, y_1, z_1, x_1 \in \mathbb{R}$. Svarbu pastebėti, kad (0.2) funkcijos, vadinamos solitonais, nėra (0.1) bendrieji sprendiniai, o tenkina sistemą tik prie tam tikrų pradinių sąlygų.

Kita vertus, [6] parodyta, kad (0.2) yra bendrasis Rikati² lygties su pastoviais koeficientais

$$y'_x = a_0 + a_1 y + a_2 y^2, \quad a_0, a_1, a_2 \in \mathbb{R}; \quad (0.3)$$

sprendinys. Tai motyvuoja panagrinėti multiplikatyviuoju sąryšiu susietų Rikati lygčių sistemą:

$$\begin{aligned} y'_x &= a_0 + a_1 y + a_2 y^2 + a_3 yz; \\ z'_x &= b_0 + b_1 z + b_2 z^2 + b_3 yz. \end{aligned} \quad (0.4)$$

Šio darbo tikslas - parodyti, kad (0.4) turi sudėtingesnius sprendinius nei (0.2) sprendinius, kurie gali būti išreikšti tokia forma:

$$y(x) = \sigma \frac{(e^{\eta(x-c)} - y_1)(e^{\eta(x-c)} - y_2)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - x_2)}; \quad z(x) = \gamma \frac{(e^{\eta(x-c)} - z_1)(e^{\eta(x-c)} - z_2)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - x_2)}. \quad (0.5)$$

Šių sprendinių egzistavimo įrodymas gali įnešti naują kokybę į biologinių sistemų nagrinėjimą.

Ieškodami būtinų ir pakankamų (0.5) egzistavimo (0.4) sistemoje sąlygų naudosimės atvirktinio balansavimo bei operatorinio skaičiavimo metodais [7]. Šie metodai sėkmingai taikomi analizinių paprastųjų diferencialinių lygčių sprendinių paieška - naudojantis jais yra parodyta, kad (0.2)

¹Warder Clyde Allee (1885 - 1955), JAV zoologas ir ekologas

²Jacopo Francesco Riccati (1676-1754), italų matematikas

tipo sprendiniai egzistuoja KdV [8], Rikati [6], Heslio³ [9] ir Liuvilio⁴ lygtyse [10].

1. LITERATŪROS APŽVALGA

Šioje dalyje apžvelgsime tikslių diferencialinių lygčių sprendinių apskaičiavimo metodus, pritatysime solitono sąvoką. Pristatysime pagrindinį įrankį, šiame darbe naudojamą tiksliais uždaros formos diferencialinių lygčių sprendiniams rasti -- operatorinį skaičiavimą, bei jo veikimą pademonstruosime išsprendami Rikati diferencialinę lygtį. Pagrįsime programinės įrangos paketo Maple pasirinkimą simboliniams skaičiavimams atlikti.

1.1. Solitonai

Fizikoje solitonu⁵ vadinama netiesinė banga, kuri keliaudama išlaiko savo formą net ir sąveikaudama su kitomis bangomis. Tai - viena iš pagrindinių sąvokų, parodanti galimą fizikinių reiškinių dualumą, kadangi solitonas turi ir banginių, ir dalelės savybių. Fizikai bendrame termino solitonas vartojime neretai atsisako reikalavimo išlaikyti formą po susidūrimo, tačiau iš matematinės pusės dauguma svarbių solitono savybių yra susiję būtent su šiuo reikalavimu [11].

Istoriškai terminas solitonas yra sietinas su Kortevego⁶-de Vrizo⁷ (KdV) lygtimi:

$$\frac{\partial u}{\partial t} + 6u \frac{\partial u}{\partial x} + \frac{\partial^3 u}{\partial x^3} = 0. \quad (1.1)$$

Ši lygtis aprašo silpnai netiesines vandens bangas negiliam kanale ar baseine. Vėliau buvo pastebėta, kad (1.1) gaunama ir nagrinėjant sistemas su stipriais netiesiškumo efektais, tačiau sprendiniai vistiek išlaiko solitonų savybes [12].

Funkcinė solitonų forma gali būti įvairi, tačiau dažniausiai minimi KdV, sine-Gordono⁸ bei Šriodingerio⁹ lygčių generuojami solitonai [11]. Trumpai aptarsime kiekvieną iš jų.

KdV lygties (1.1) solitoninį sprendinį galima gauti pritaikius keitinį

$$u = \psi(y), \quad y = x - ct, \quad (1.2)$$

čia c nurodo pastovų bangos sklidimo greitį. Kintamasis y vadinamas banginiu kintamuoju ir yra dažnai sutinkamas solitoninių sprendinių paieškos metoduose. Įrašę (1.2) į (1.1) gauname:

$$\psi''' + \psi\psi' - c\psi' = 0, \quad (1.3)$$

³Andrew Huxley (1917 - 2012), anglų biofizikas ir fiziologas, Karališkosios draugijos narys, Nobelio premijos laureatas.

⁴Joseph Liouville (1809 - 1882), prancūzų matematikas, Karališkosios švedų mokslo akademijos narys.

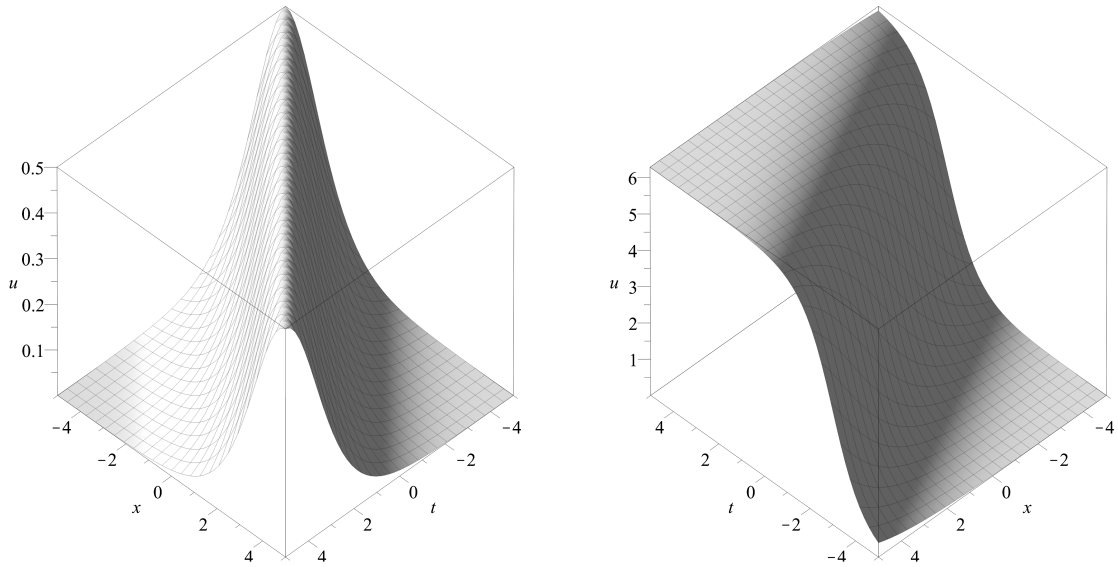
⁵angl. *soliton* arba *solitary wave*.

⁶Diederick Korteweg (1848 - 1941), olandų matematikas, Karališkosios Nyderlandų mokslo akademijos narys.

⁷Gustav de Vries (1866 - 1934), olandų matematikas.

⁸Walter Gordon (1893 - 1939), vokiečių fizikas-teoretikas.

⁹Erwin Rudolf Josef Alexander Schrödinger (1887 - 1961), austrų fizikas, Nobelio premijos laureatas.



(a) Solitonas, apibrėžtas (1.5) su parametru reikšmėmis $c = 1, x_0 = 0$.

(b) Solitonas, apibrėžtas (1.7) su parametru reikšmėmis $a = b = \lambda = k = \theta_0 = 1, \mu = 2$.

1.1 Pav.: KdV ir sine-Gordono lygties solitonai.

kur $' \sim \frac{d}{dy}$. Suintegravę (1.3) ir nustatę integravimo konstantas gauname:

$$-\frac{c}{2}\psi^2 + \frac{1}{2}(\psi')^2 + \psi^3 = 0. \quad (1.4)$$

Atskyrę kintamuosius ir suintegravę turėsime:

$$u(t, x) = \frac{c}{2} \operatorname{sech}^2 \left(\frac{c}{2} (x - ct - x_0) \right), \quad (1.5)$$

kur x_0 - konstanta. Taigi, KdV tipo solitonai išreiškiami eksponentinių funkcijų sumų santykiais [13], [8]. Sprendinys (1.5) pavaizduotas 1.1 (a) pav.

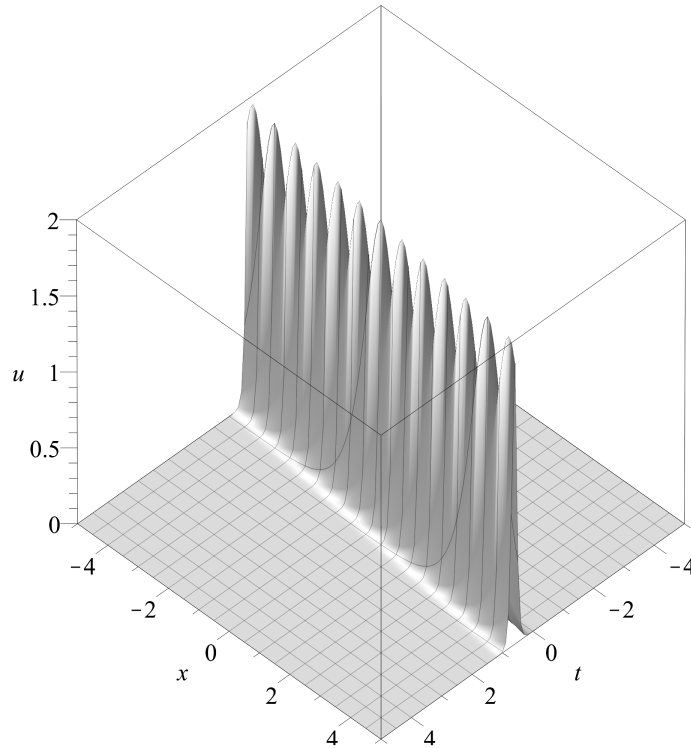
Sine-Gordono lygtis

$$\frac{\partial^2 u}{\partial t^2} - a \frac{\partial^2 u}{\partial x^2} = b \sin(\lambda u); \quad (1.6)$$

turi solitoninius sprendinius:

$$u(t, x) = \begin{cases} \frac{4}{\lambda} \arctan \left(\exp \left(\pm \frac{b\lambda(kx + \mu t + \theta_0)}{\sqrt{b\lambda(\mu^2 - ak^2)}} \right) \right), & b\lambda(\mu^2 - ak^2) > 0; \\ \frac{4}{\lambda} \arctan \left(\exp \left(\pm \frac{b\lambda(kx + \mu t + \theta_0)}{\sqrt{b\lambda(ak^2 - \mu^2)}} \right) \right) - \frac{\pi}{\lambda}, & b\lambda(\mu^2 - ak^2) < 0. \end{cases} \quad (1.7)$$

Čia λ - banginis skaičius, o μ, θ_0 - pasirenkamos konstantos [14]. Sprendinys (1.7) pavaizduotas 1.1 (b) pav.



1.2 Pav.: Solitonas, apibrėžtas (1.9) su parametru reikšmėmis $\alpha = 4, q = 1, V = 8$.

Netiesinė Šriodingerio lygtis (kartais dar vadinama kubine Šriodingerio lygtimi)

$$i \frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + qu|u|^2 = 0, \quad (1.8)$$

čia $u(t, x) = \Re u(t, x) + i \Im u(t, x)$, turi solitoninius sprendinius:

$$u(t, x) = \sqrt{\frac{\alpha}{q}} \operatorname{sech}(\sqrt{\alpha}(x - Vt)); \alpha, q > 0, \quad (1.9)$$

kur V - bangos sklidimo greitis, $\alpha, q > 0$ - konstantos [11]. Sprendinys (1.9) pavaizduotas 1.2 pav.

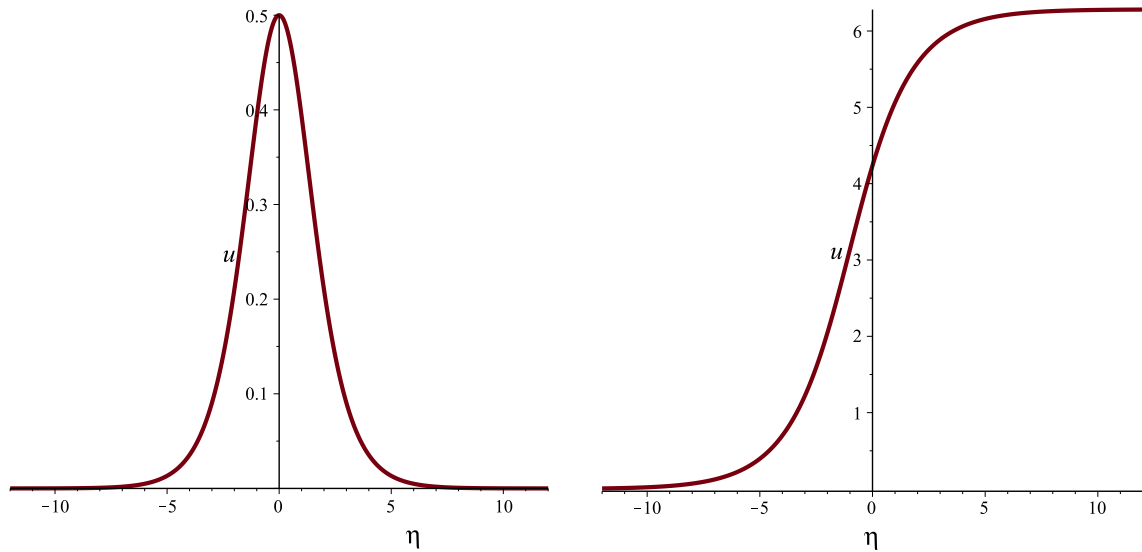
Pažymėtina, kad lygtys (1.1), (1.6), (1.8) turi ir daugiau solitoninių sprendinių - visi jie dar nesurasti, tačiau kai kuriems lygčių variantams yra įrodyta, kad skirtingų solitonų yra be galo daug [15], [11], [16]. Taip pat ir lygčių, kurių bent dalis sprendinių yra solitonai yra be galo daug.

Šiame darbe ieškosime sprendinių, kurių forma - modikuota KdV solitono forma. Jie išreiškiami eksponentinių funkcijų sumų santykiu:

$$y_n(x) = \frac{\prod_{k=0}^n (e^{\eta(x-c)} - y_k)}{\prod_{k=0}^n (e^{\eta(x-c)} - x_k)}. \quad (1.10)$$

Funkciją $y_n(x)$ vadinsime n -tos eilės solitonu. Reikia pastebėti, kad (1.10) yra vieno kintamojo funkcija, tuo tarpu nagrinėti (1.1), (1.6), (1.8) lygčių sprendiniai priklauso nuo kintamųjų t ir x . Nepaisant to, (1.10) lieka korektiška vadinti solitoniniu sprendiniu, kadangi minėti sprendiniai iš

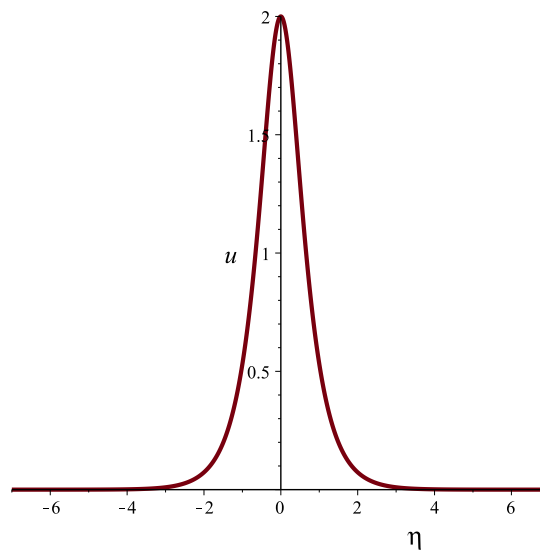
tikrųjų priklauso tik nuo vieno kintamojo - tam tikro kintamųjų t, x tiesinio darinio, vadinamo banginiu kintamuoju (pav. 1.3, 1.4).



(a) Solitono, apibrėžto (1.5) su parametru reikšmėmis $c = 1, x_0 = 0$ priklausomybė nuo banginio kintamojo $\eta := x - ct$.

(b) Solitono, apibrėžto (1.7) su parametru reikšmėmis $a = b = \lambda = k = \theta_0 = 1, \mu = 2$ priklausomybė nuo banginio kintamojo $\eta := kx + \mu t$.

1.3 Pav.: KdV ir sine-Gordono lygties solitonų priklausomybė nuo banginio kintamojo.



1.4 Pav.: Solitono, apibrėžto (1.9) su parametru reikšmėmis $\alpha = 4, q = 1, V = 8$ priklausomybė nuo kintamojo $\eta := x - Vt$.

1.2. Tikslių diferencialinių lygčių sprendinių paieškos metodų apžvalga

Šiame poskyryje apžvelgsime šiuo metu sparčiai populiarėjančius metodus tiksliams diferencialinių lygčių sprendiniams gauti. Visus pateiktus metodus vienijantis bruožas - simboliųjų pertvarkymų paketų (Maple, Wolfram Mathematica, MuPad, Axiom ir kt.) naudojimas didelės apimties netiesinėms algebrinėms lygtims spręsti ar algebrinės išraiškos pertvarkyti.

Atkreipsime dėmesį, kad daugumoje publikacijų šie metodai taikomi dalinių išvestinių diferencialinėms lygtims, tačiau pačiame pirmame žingsnyje pereinama prie paprastosios diferencialinės lygties, taigi gali būti pradėti taikyti ir turint tik paprastąją diferencialinę lygtį.

1.2.1. Eksponentinių funkcijų metodas¹⁰

Tai -- vienas dažniausiai publikacijose sutinkamų metodų. Tarkime, duota dalinių išvestinių diferencialinė lygtis

$$P\left(u, \frac{\partial u}{\partial t}, \frac{\partial u}{\partial x}, \frac{\partial^2 u}{\partial t^2}, \frac{\partial^2 u}{\partial x^2}, \frac{\partial^2 u}{\partial t \partial x}, \dots\right) = 0. \quad (1.11)$$

Įvedamas banginis kintamasis

$$u(t, x) = U(\eta), \quad \eta = kx + wt, \quad k, w \in \mathbb{R}/\{0\}. \quad (1.12)$$

Įstačius (1.12) į (1.11) gaunama paprastoji diferencialinė lygtis

$$Q\left(U, U'_\eta, U''_\eta, \dots\right) = 0. \quad (1.13)$$

Tariama, kad šios lygties sprendinys turi formą:

$$U(\eta) = \frac{a_c \exp(c\eta) + \dots + a_{-d} \exp(-d\eta)}{b_p \exp(p\eta) + \dots + b_{-q} \exp(-q\eta)}, \quad (1.14)$$

čia c, d, p, q -- neneigiami sveiki skaičiai, o $a_c, \dots, a_{-d}, b_p, \dots, b_{-q}$ -- realios konstantos.

Kitame žingsnyje įvykdomas balansavimas: įstačius funkciją (1.14) į gautą paprastąją diferencialinę lygtį (1.13) parenkami parametrai c, d, p, q tokiu būdu, kad naudojamas keitinys turėtų prasmę, t.y. subalansuojant aukščiausio ir mažiausio eksponentės rodiklio prie aukščiausio ir mažiausio laipsnio netiesinių ir tiesinių lygties narių. Nepaisant to, kad [17] parodyta, kad šis žingsnis visais atvejais veda prie $p = c, d = q$, šio žingsnio dauguma autorių neatsisakė.

Suradus parametrus p, c, d, q grįžtama prie tos pačios paprastosios diferencialinės lygties, kuri, pritaikius keitinį, tampa algebrine. Surinkus koeficientus prie vienodų $\exp(k\eta)$ laipsnių sprendžiama algebrinių lygčių sistema, gaunama prilyginus koeficientus nuliui. Sistemos sprendinius įstačius į (1.14) gaunamas diferencialinės lygties sprendinys.

Nepaisant itin plataus šio metodo taikymo [18], [19], [20] pastarasis susilaukė itin daug kritikos dėl neteisingų ar trivialių sprendinių pateikimo bei pradinių sąlygų, prie kurių gauti sprendiniai galioja, nenagrinėjimo [21], [6], [8], [22], [23], [17], [24], [25].

¹⁰ angl. *Exp-function method*

1.2.2. Hiperbolinio tangento metodas¹¹

Šis metodas taikomas panašiai kaip ir eksponentinių funkcijų metodas. Kuomet duota (1.11), įvedamas (1.12) keitinys, tačiau vietoje hipotezės (1.14) priimama, kad

$$u(t, x) = U(\eta) = \sum_{k=0}^M a_k (\tanh \eta)^k, \quad (1.15)$$

arba

$$u(t, x) = U(\eta) = \sum_{k=0}^M a_k (\tanh \eta)^k + \sum_{k=1}^M b_k (\tanh \eta)^{-k}. \quad (1.16)$$

Kuomet naudojama (1.15) lygybė, metodas vadinamas hiperbolinio tangento, kuomet (1.16) - išplėstiniu hiperbolinio tangento metodu. Toliau skaičiavimai tęsiami analogiškai, kaip ir eksponentinių funkcijų metode - sudaromos netiesinės algebrinės lygtys ir ieškoma jų sprendinių [26], [27], [28].

Natūralu, kad šis metodas, itin primenantis eksponentinių funkcijų metodą taip susilaukė kritikos dėl nekorektiško taikymo [23]. Be to, buvo parodyta, kad hiperbolinio tangento metodas ekvivalentus kitame poskyryje aptariamam (G'/G) skleidinio metodui [29].

1.2.3. (G'/G) skleidinio metodas¹²

(G'/G) skleidinio metodas yra praeitame poskyryje minėto metodo hiperbolinio tangento metodo apibendrinimas pirmąkart publikuotas [30]. Šiame metode, kaip ir anksčiau aptartuose, į (1.11) įvedamas banginis kintamasis (1.12) ir gaunama paprastoji diferencialinė lygtis (1.13).

Įvedama funkcija $G(\eta)$ ir tariama, kad (1.13) sprendinį galima išreikšti $G(\eta)$ logaritminės išvestinės laipsnių pagalba:

$$U(\eta) = \sum_{k=0}^N a_k \left(\frac{G'}{G} \right)^k; \quad a_k \in \mathbb{R}, \quad (1.17)$$

o funkcija $G(\eta)$ tenkina tiesinę antros eilės diferencialinę lygtį

$$G'' - \lambda G' - \mu G = 0. \quad (1.18)$$

Įstačius funkciją $U(\eta) = a_N \eta^N$ surandamas toks parametras N , su kuriuo lygtis ir sprendinys būtų subalansuoti (šis žingsnis ekvivalentus eksponentinių funkcijų metodo balansavimo žingsniui). Radus N , funkcija (1.17) įstatoma į (1.13) ir suprastinama atsižvelgus į (1.18). Kaip ir anksčiau aptartuose metoduose, surinkus koeficientus prie skirtingų G'/G laipsnių gaunama algebrinė lygčių sistema parametrams $a_0, \dots, a_N, \lambda, \mu, k, w$ rasti [30].

Kaip parodyta [29], kadangi (1.18) sprendinys prie tam tikrų λ, μ gali būti išreikštas kaip $G(\eta) = C_1 e^\eta + C_2 e^{-\eta}$, gauname, kad specialiai parinkus konstantas C_1, C_2 funkcija G'/G tampa lygi $\tanh(\eta)$.

¹¹angl. *tanh-function method*

¹²angl. *(G'/G) expansion method*

Šis metodas plačiai taikomas įvairiems banginiams dalinių išvestinių diferencialinių lygčių sprendiniams rasti [31], [32], [33].

1.2.4. Kiti metodai

Literatūroje galima rasti daug metodų, panašių į tris aukščiau aptartus. Verta paminėti dažniau sutinkamus: paprasčiausios lygties metodą¹³ [34], sinusų-kosinusų metodą¹⁴ [35], Kudrešovo arba Q -metodą¹⁵ [36].

Tarp klasikinių metodų, kurie naudojami solitoniniams ir kitiems tiksliams dalinių išvestinių sprendiniams rasti galima paminėti Hirotos metodą¹⁶ [37], Baklundo transformacijas¹⁷ [38] bei atvirkštinio išsklaidymo metodą¹⁸ [39].

Didžiausias šių metodų minusas yra tai, kad jais naudojantis negalima automatiškai generuoti sprendinio struktūros: ją reikia atspėti, net nežinant ar lygtis tokį sprendinį gali turėti. Kad išvengtume šios problemos, šiame darbe naudosime dviejų metodų kombinaciją: atvirkštinio balansavimo, leidžiančio pasakyti ar diferencialinė lygtis ar lygčių sistema gali turėti postuluojamą sprendinį ir operatorinio metodo, leidžiančio surasti norimo sprendinio analizinę išraišką bei jo egzistavimo sąlygas pradinių sąlygų erdvėje.

2. MEDŽIAGOS IR TYRIMO METODAI

2.1. Operatorinio metodo taikymas paprastosioms diferencialinėms lygtims

Šiame poskyryje apžvelgsime operatorinio metodo taikymą paprastosioms diferencialinėms lygtims spręsti.

2.1.1. Formalių eilučių algebra ir tiesiniai operatoriai

Apibrėžimas 2.1 *Eilutę*

$$f = \sum_{k=0}^{+\infty} a_k x^k, \quad a_k \in \mathbb{C}, \quad k = 0, 1, \dots \quad (2.1)$$

vadinsime formalia eilute, o aibę $\mathbb{F} = \left\{ f \mid f \text{ tenkina (2.1)} \right\}$ -- formalių eilučių aibe.

Apibrėžimas 2.2 *Aibė \mathbb{F} , kartu su įprastinėmis eilučių sumos, eilučių daugybos ir daugybos iš skaliaro operacijomis vadinama (formalių) eilučių algebra.*

Tegul $f = \sum_{k=0}^{+\infty} a_k x^k$, $g = \sum_{k=0}^{+\infty} b_k x^k$. Operacijos atliekamos paelemenčiui:

¹³ angl. *Simplest equation method*

¹⁴ angl. *sine-cosine function method*

¹⁵ angl. *Q-function method*

¹⁶ angl. *Hirota's bilinear method*

¹⁷ angl. *Bäcklund transformations*

¹⁸ angl. *Inverse scattering method*

$$1. f + g = g + f = \sum_{k=0}^{+\infty} (a_k + b_k) x^k.$$

$$2. f \cdot g = g \cdot f = \sum_{k=0}^{+\infty} \left(\sum_{l=0}^k a_l b_{l-k} \right) x^k.$$

$$3. c \cdot f = f \cdot c = \sum_{k=0}^{+\infty} (ca_k) x^k.$$

Neutralusis šios algebros elementas yra eilutė su koeficientais $a_0 = 1, a_k = 0, k > 1$, anuliuojantis elementas -- eilutė su visais nulinais koeficientais.

Apibrėžimas 2.3 Operatorius \mathbf{A} , veikiantis aibėje \mathbb{F} ir tenkinantis lygybę

$$\mathbf{A}(\alpha f + \beta g) = \alpha \mathbf{A}f + \beta \mathbf{A}g, \quad (2.2)$$

kur $\alpha, \beta \in \mathbb{C}, f, g \in \mathbb{F}$, vadinamas tiesiniu operatoriumi.

Nesunku suvokti, kad tiesinio operatoriaus veikimą pilnai nusako jo poveikis kiekvienam elementui $x^k, k = 0, 1, \dots$ (šių elementų (begalinė) aibė sudaro algebros \mathbb{F} bazę), nes:

$$\mathbf{A}f = \mathbf{A} \sum_{k=0}^{+\infty} a_k x^k = \sum_{k=0}^{+\infty} a_k \mathbf{A}x^k. \quad (2.3)$$

Diferencialinių lygčių sprendimui svarbiausi elementarieji operatoriai:

1. Vienetinis operatorius $\mathbf{I} : \mathbf{I}f = f$.
2. Nulinis operatorius $\mathbf{0} : \mathbf{0}f = 0$.
3. Diferencijavimo pagal x operatorius $\mathbf{D}_x : \mathbf{D}_x x^k = \begin{cases} 0, & k = 0 \\ kx^{k-1}, & k > 0 \end{cases}$.
4. Integravimo pagal x operatorius $\mathbf{L}_x : \mathbf{L}_x x^k = \frac{x^{k+1}}{k+1}$.

Iš šių operatorių sukonstruoti sudėtingesni operatoriai leidžia išreikšti paprastosios diferencialinės lygties sprendinį begaline eilute.

2.1.2. Funkcijų klasės

Nagrinėsime dvi funkcijų klases. Pirmajai klasei priklauso funkcijos $p_j = p_j(c, s_0, s_1, \dots, s_{n-1})$, nusakančios atitiktį

$$p_j : I_c \times I_{s_0} \times \dots \times I_{s_{n-1}} \rightarrow \mathbb{R}, \quad (2.4)$$

kur $I_c, I_{s_k} \subset \mathbb{R}, k = 1, \dots, n-1$ yra netuščios aibės, sudarytos iš intervalų arba jų sąjungų. Šiai klasei priklausančios funkcijos yra diferencijuojamos norimą skaičių kartų kintamųjų c, s_0, \dots, s_{n-1} atžvilgiu. Pirmąją funkcijų klasę žymėsime $\Phi_{c, s_0, \dots, s_{n-1}}$.

Antrosios klasės funkcijos konstruojamos tokiu algoritmu:

1. Suformuojama kintamojo $x \in \mathbb{R}$ laipsninė eilutė, kurios koeficientai yra $\Phi_{c,s_0,\dots,s_{n-1}}$ elementai, pavyzdžiui:

$$f_0(x, c, s_0, \dots, s_{n-1}) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} p_j(c, s_0, \dots, s_{n-1}). \quad (2.5)$$

Eilutės f_0 konvergavimo spindulys $|x-c| < R_f \leq +\infty$ priklauso nuo f_0 išraiškos ir p_j savybių.

2. Jeigu įmanoma f_0 pratęsiama į platesnę apibrėžimo sritį ir gaunama funkcija $f(x, c, s_0, \dots, s_{n-1})$, priklausanti antrajai funkcijų klasei, kurią žymėsime $\Phi_{x,c,s_0,\dots,s_{n-1}}$.

Galima pastebėti, kad $\Phi_{c,s_0,\dots,s_{n-1}} \subset \Phi_{x,c,s_0,\dots,s_{n-1}}$.

Pavyzdys 2.1 Tarkime, duota funkcijų seka:

$$p_j = p_j(c, s_0, s_1) := j! \left(\frac{s_1}{s_0(1-c)} \right)^j, \quad j = 0, 1, \dots \quad (2.6)$$

Kintamųjų c, s_0, s_1 apibrėžimo sritys lygios:

$$I_c = (-\infty, 1) \cup (1, +\infty), \quad I_{s_0} = (-\infty, 0) \cup (0, +\infty), \quad I_{s_1} = (-\infty, +\infty). \quad (2.7)$$

Be to, egzistuoja norimos eilės p_j išvestinės, taigi $p_j \in \Phi_{c,s_0,\dots,s_{n-1}}$.

Suformuokime laipsninę eilutę

$$f_0(x, c, s_0, s_1) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \left(j! \left(\frac{s_1}{s_0(1-c)} \right)^j \right) = \sum_{j=0}^{+\infty} \left(\frac{s_1(x-c)}{s_0(1-c)} \right)^j. \quad (2.8)$$

Pagal nykstamosios geometrinės progresijos sumos formulę, (2.8) konverguoja, kai $\left| \frac{s_1(x-c)}{s_0(1-c)} \right| < 1$ ir turi sumą:

$$\frac{s_0(c-1)}{s_0(c-1) - s_1(c-x)}, \quad s_0(c-1) \neq s_1(c-x). \quad (2.9)$$

Pažymėję

$$f(x, c, s_0, s_1) := \frac{s_0(c-1)}{s_0(c-1) - s_1(c-x)}; \quad s_0(c-1) \neq s_1(c-x), \quad (2.10)$$

gauname antrajai funkcijų klasei priklausančią funkciją, t.y. pratęsiame f_0 į funkciją f .

Tokiu būdu operatoriniame skaičiavime pateikiama diferencialinės lygties sprendinio struktūra, kuri nebūtinai turi būti konverguojanti (Koši prasme) eilutė, kad tenkintų duotąją lygtį. ■

Standartinės funkcijos, tokios kaip e^x , $\sin x$, $\cos x$ ir t.t. interpretuojamos kaip laipsninės eilutės, taigi taip pat priklauso antrajai funkcijų klasei.

$$e^x := \sum_{j=0}^{+\infty} \frac{x^j}{j!}, \quad \sin x := \sum_{j=0}^{+\infty} (-1)^j \frac{x^{2j+1}}{(2j+1)!}, \quad \cos x := \sum_{j=0}^{+\infty} (-1)^j \frac{x^{2j}}{(2j)!}. \quad (2.11)$$

2.1.3. Apibendrintasis diferencialinis operatorius

Tegul $P := P(c, s), Q := Q(c, s) \in \Phi_{c,s}$.

Pastaba 2.1 Bet kokį operatorių \mathbf{A} pakeltą nuliniu laipsniu, laikysime vienetiniu operatoriumi \mathbf{I} , t.y.

$$\mathbf{A}^0 = \mathbf{I}, \quad \mathbf{I}f = f, \quad \forall f. \quad (2.12)$$

Apibrėžimas 2.4 Operatorių

$$\mathbf{D} := P(c, s)\mathbf{D}_c + Q(c, s)\mathbf{D}_s \quad (2.13)$$

vadinsime apibendrintuoju diferencialiniu operatoriumi. Trumpumo dėlei naudosime žymėjimą $\mathbf{D} = P\mathbf{D}_c + Q\mathbf{D}_s$, kuomet funkcijų P, Q kintamųjų skaičius bus aiškus iš konteksto.

Apibendrintojo diferencialinio operatoriaus pagrindinės savybės ($f, g, h \in \Phi_{c,s}$):

1. $\mathbf{D}(af + bg) = a\mathbf{D}f + b\mathbf{D}g$;
2. $\mathbf{D}(fg) = (\mathbf{D}f)g + f\mathbf{D}g$;
3. $\mathbf{D}\frac{f}{g} = \frac{(\mathbf{D}f)g - f\mathbf{D}g}{g^2}$;
4. $\mathbf{D}h(f(c, s), g(c, s)) = \frac{\partial h}{\partial f}\mathbf{D}f + \frac{\partial h}{\partial g}\mathbf{D}g$.

Kaip matome, galioja įprastos išvestinių apskaičiavimo taisyklės, pakeičiant įprastą diferencijavimą apibendrintuoju.

Įrodysime antrą savybę:

$$\begin{aligned} \mathbf{D}(fg) &= (P\mathbf{D}_c + Q\mathbf{D}_s)(fg) = P((\mathbf{D}_c f)g + f(\mathbf{D}_c g)) + Q((\mathbf{D}_s f)g + f(\mathbf{D}_s g)) \\ &= (P(\mathbf{D}_c f)g + Q(\mathbf{D}_s f)g) + (fP(\mathbf{D}_c g) + fQ(\mathbf{D}_s g)) = (\mathbf{D}f)g + f(\mathbf{D}g). \end{aligned} \quad (2.14)$$

Analogiškais pertvarkymais galima įrodyti ir kitas savybes.

Pavyzdys 2.2 Pailiustruosime ketvirtą savybę. Tarkime $h(f, g) := e^{f+g}$, $f := c^2$, $g := s^2$, o apibendrintas diferencialinis operatorius apibrėžtas (2.13) lygybe. Tuomet

$$\begin{aligned} \mathbf{D}h(f, g) &= \mathbf{D}e^{c^2+s^2} = P\mathbf{D}_c e^{c^2+s^2} + Q\mathbf{D}_s e^{c^2+s^2} = Pe^{c^2+s^2} \cdot 2c + Qe^{c^2+s^2} \cdot 2s \\ &= e^{c^2+s^2} (P \cdot 2c + Q \cdot 2s) = e^{f+g} (P\mathbf{D}_c c^2 + Q\mathbf{D}_s s^2). \blacksquare \end{aligned} \quad (2.15)$$

2.1.4. Paprastųjų diferencialinių lygčių sprendinių operatorinės išraiškos

Teorema 2.1 Tarkime, duota paprastoji diferencialinė lygtis išreikštineje formoje:

$$y'_x = P(x, y), \quad (2.16)$$

kur funkcija $P = P(c, s)$ priklauso pirmajai funkcijų klasei $\Phi_{c,s}$. Tarkime, kad (2.16) lygties bendrasis sprendinys $y = y(x, c, s)$, priklausantis nuo kintamojo x ir parametų c, s tenkina pradinę

sąlygą $y(c, c, s) = s$. Diferencialinei lygčiai (2.16) apibrėžiamas apibendrintasis diferencialinis operatorius:

$$\mathbf{D} := \mathbf{D}_c + P(c, s)\mathbf{D}_s, \quad (2.17)$$

tuomet jos sprendinys užrašomas:

$$y = y(x, c, s) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s. \quad (2.18)$$

Teoremos įrodymą galima rasti [7].

Patogu įvesti pažymėjimą $p_j := \mathbf{D}^j s$, tuomet $p_{j+1} = \mathbf{D}p_j, j = 0, 1, \dots$, tuomet (2.18) galima užrašyti:

$$y = y(x, c, s) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} p_j. \quad (2.19)$$

Šis užrašas leidžia automatiškai generuoti tikslų (2.16) sprendinį išreikštą begaline eilute. Svarbu pastebėti, kad operatorinio skaičiavimo teorijoje nekalbama apie konvergavimą, taigi (2.19) eilutė gali ir nekonverguoti su visomis parametru c, s reikšmėmis. Tuomet pasinaudojama jau minėtu pratęsimu nepaisant to, kad „sumuojama“ nekonverguojanti eilutė, gauti uždaros formos¹⁹ (išreikšti standartinėmis funkcijomis) sprendiniai galioja ir išplėstai parametru aibei.

Svarbu pažymėti, kad analogiškai (2.16) galima spręsti ir aukštesnės eilės išreikštines paprasčiausias diferencialines lygtis bei jų sistemas. Apibendrintieji diferencialiniai operatoriai dažniausiai pasitaikantiems uždaviniams pateikti B.1 lent.

2.1.5. Tiesiškai rekurentinės sekos

Apibrėžimas 2.5 Eilės m tiesiškai rekurentine seka²⁰ vadinama skaičių arba funkcijų seka $(x_j; j \in \mathbb{Z}_0)$, tenkinanti rekurentinį sąryšį:

$$x_{j+m} = \alpha_{m-1}x_{j+m-1} + \dots + \alpha_0x_j; \quad (2.20)$$

su konstantų rinkiniu $\alpha_k \in \mathbb{C}, k = 0, \dots, m$.

Tiesiškai rekurentinės sekos turi svarbią savybę: jeigu seka $(x_j; j \in \mathbb{Z}_0)$ tenkina (2.20), tai sukonstravę Hankelio²¹ determinantų seką

$$d_0 := 1, d_n := \det(p_{k+l-2})_{1 \leq k, l \leq n}, n = 1, 2, \dots; \quad (2.21)$$

gauname, kad

$$d_{m-1} \neq 0, d_{m+k} = 0, k = 0, 1, 2, \dots \quad (2.22)$$

¹⁹ angl. *closed form*

²⁰ angl. *linear recurring sequence*

²¹ *Hermann Hankel* (1839-1873), vokiečių matematikas

Simboliškai žymėsime

$$\text{order}(x_j; j \in \mathbb{Z}_0) = m. \quad (2.23)$$

Kai žinoma sekos eilė m galima užrašyti sekos $(x_j; j \in \mathbb{Z}_0)$ charakteringą lygtį²²:

$$\begin{vmatrix} x_0 & x_1 & \dots & x_m \\ x_1 & x_2 & \dots & x_{m+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m-1} & x_m & \dots & x_{2m-1} \\ 1 & \rho & \dots & \rho^m \end{vmatrix} = 0. \quad (2.24)$$

Apibrėžimas 2.6 Lygties (2.24) šaknis ρ_1, \dots, ρ_l vadinsime sekos $(x_j; j \in \mathbb{Z}_0)$ charakteringosios šaknimis. Šaknies ρ_k kartotinumą žymėsime n_k .

Irodyta, kad bet kokiame komutatyviame žiede tiesiškai rekurentinės sekos elementus galima išreikšti charakteringųjų lygčių šaknų laipsnių ir binominių koeficientų tiesiniu dariniu [40]. Jeigu nagrinėjamas tik kompleksinių skaičių laukas, tuomet tokia išraiška yra būtina ir pakankama, kad seka būtų tiesiškai rekurentinė [7]:

Teorema 2.2 Seka $(x_j; j \in \mathbb{Z}_0)$ yra m -tos eilės tiesiškai rekurentinė seka:

$$\text{order}(x_j; j \in \mathbb{Z}_0) = n_1 + \dots + n_l = m; \quad (2.25)$$

tada ir tik tada, kai

$$x_j = \sum_{k=1}^l \sum_{r=0}^{n_k-1} \lambda_{kr} \binom{j}{r} \rho_k^{j-r}, \quad (2.26)$$

čia λ_{kr} - konstantos, randamos iš tiesinės lygčių sistemos

$$\sum_{k=1}^l \sum_{r=0}^{n_k-1} \lambda_{kr} \binom{j}{r} \rho_k^{j-r} = x_j, \quad j = 0, \dots, m-1. \quad (2.27)$$

Pastebėsime, kad jeigu visos (2.24) šaknys skirtingos, tai (2.26) užrašomas paprasčiau:

$$x_j = \sum_{k=1}^m \lambda_k \rho_k^j. \quad (2.28)$$

Taip pat svarbu pažymėti, kad (2.26) laikomasi tokių susitarimų:

$$0^0 := 1, \quad \binom{j}{r} \rho_k^{j-r} = 0, \text{ jei bent vienas iš daugiklių lygus nuliui.} \quad (2.29)$$

²²angl. *characteristic equation* arba *auxiliary equation*

2.1.6. Tiesiškai rekurentinių sekų panaudojimas solitoninių sprendinių konstravimui

Tegul duota diferencialinė lygtis

$$y'_x = P(x, y), \quad y(c) = s. \quad (2.30)$$

Įvedame kintamojo pakeitimą:

$$\hat{x} := e^{\eta x}, \quad x = \frac{1}{\eta} \ln \hat{x}; \quad \eta \in \mathbb{R} / \{0\}. \quad (2.31)$$

Tuomet

$$y = y(x) = y\left(\frac{1}{\eta} \ln \hat{x}\right) =: \hat{y}(\hat{x}) = \hat{y}(e^{\eta x}) =: \hat{y}, \quad (2.32)$$

ir

$$y'_x = \eta \hat{x} \hat{y}'_{\hat{x}}. \quad (2.33)$$

Taigi (2.30) galima perrašyti:

$$\eta \hat{x} \hat{y}'_{\hat{x}} = \hat{P}(\hat{x}, \hat{y}), \quad (2.34)$$

čia $\hat{P}(\hat{x}, \hat{y}) := P\left(\frac{1}{\eta} \ln \hat{x}, \hat{y}\right)$. Pradinės (2.34) sąlygos yra:

$$\hat{c} := e^{\eta c}; \quad \hat{y}(\hat{c}, \hat{c}, s) = s. \quad (2.35)$$

Tarp (2.30) ir (2.34) lygčių sprendinių egzistuoja sąryšis:

$$y(x, c, s) = \hat{y}(\hat{x}, \hat{c}, s). \quad (2.36)$$

Apibrėžimas 2.7 Lygtį (2.34) vadinsime lygties (2.30) vaizdu, o sprendinį $\hat{y}(\hat{x}, \hat{c}, s)$ - sprendinio $y(x, c, s)$ vaizdu.

Pasinaudoję 2.1 teorema, galime užrašyti (2.34) sprendinį eilute:

$$\hat{y}(\hat{x}, \hat{c}, s) = \sum_{j=0}^{+\infty} \frac{(\hat{x} - \hat{c})^j}{j!} p_j, \quad (2.37)$$

čia $p_j := \hat{\mathbf{D}}^j s$, o $\hat{\mathbf{D}} := \mathbf{D}_{\hat{c}} + \frac{\hat{P}(\hat{c}, s)}{\eta \hat{c}} \mathbf{D}_s$ - apibendrintasis diferencialinis operatorius, atitinkantis (2.34).

Teorema 2.3 Tarkime, kad seka $\hat{p}_j := \frac{p_j}{j!}, j = 0, 1, \dots$ yra m -tos eilės tiesiškai rekurentinė seka:

$$\text{order}(\hat{p}_j; j \in \mathbb{Z}_0) = m, \quad (2.38)$$

be to, visų charakteristinių šios sekos šaknų ρ_1, \dots, ρ_m kartotinumai $n_k = 1, k = 1, \dots, m$. Nemažindami bendrumo, sakykime, kad nulinė šaknis yra pirmoji: $\rho_1 = 0$. Jeigu tenkinamos sąlygos:

1. $\hat{\mathbf{D}} \rho_k = \rho_k^2, k = 1, \dots, m;$

$$2. \widehat{D}\lambda_k = \lambda_k \rho_k, k = 1, \dots, m,$$

tai (2.37) užrašoma:

$$\widehat{y}(\widehat{x}, \widehat{c}, s) = \lambda_1 + \sum_{k=2}^m \frac{\lambda_k}{1 - (\widehat{x} - \widehat{c})\rho_k}, \quad (2.39)$$

ir, be to, galioja sąlyga $\lambda_1 + \dots + \lambda_m = s$ [10].

Reikia pastebėti, kad išraiška (2.39) elementariais pertvarkiais gali būti suvesta į racionaliąją funkciją:

$$\widehat{y}(\widehat{x}, \widehat{c}, s) = \frac{A_{m-1} \left(\frac{\widehat{x}}{\widehat{c}}\right)^{m-1} + \dots + A_1 \frac{\widehat{x}}{\widehat{c}} + A_0}{B_{m-1} \left(\frac{\widehat{x}}{\widehat{c}}\right)^{m-1} + \dots + B_1 \frac{\widehat{x}}{\widehat{c}} + B_0} = \sigma \frac{\left(\frac{\widehat{x}}{\widehat{c}} - \widehat{y}_1\right) \dots \left(\frac{\widehat{x}}{\widehat{c}} - \widehat{y}_{m-1}\right)}{\left(\frac{\widehat{x}}{\widehat{c}} - \widehat{x}_1\right) \dots \left(\frac{\widehat{x}}{\widehat{c}} - \widehat{x}_{m-1}\right)}, \quad (2.40)$$

čia $A_k, B_k, \widehat{x}_k, \widehat{y}_k, k = 0, \dots, m-1$ priklauso nuo \widehat{c}, η, s . Sugrįžę prie kintamojo x iš (2.40) gauname solitoninį sprendinį:

$$y(x, c, s) = \sigma \frac{\left(e^{\eta(x-c)} - y_1\right) \dots \left(e^{\eta(x-c)} - y_{m-1}\right)}{\left(e^{\eta(x-c)} - x_1\right) \dots \left(e^{\eta(x-c)} - x_{m-1}\right)}. \quad (2.41)$$

Šiame poskyryje aptartus sprendimo būdus pademonstruosime išsprendę nesusietą Rikati lygtį.

2.1.7. Nesusietos Rikati lygties sprendimas

Tarkime, duota Rikati lygtis su pastoviais koeficientais [41]:

$$y'_x = c_0 + c_1 y + c_2 y^2, \quad (2.42)$$

čia $c_0, c_1, c_2 \in \mathbb{R}; c_2 \neq 0$.

Lygtį (2.42) galima perrašyti kanoninėje formoje:

$$y'_x = \beta (y - u_1) (y - u_2), \quad (2.43)$$

kur $\beta \in \mathbb{R}$ ir u_1, u_2 yra realios arba kompleksinės jungtinės konstantos.

Yra žinoma, kad (2.44) bendrasis sprendinys $y = y(x, c, s)$ lygus [41]:

$$y(x, c, s) = \frac{u_2(s - u_1)e^{\beta(u_1 - u_2)(x-c)} - u_1(s - u_2)}{(s - u_1)e^{\beta(u_1 - u_2)(x-c)} - (s - u_2)}, \quad (2.44)$$

čia x - nepriklausomas kintamasis; s - y pradinė reikšmė taške $x = c$:

$$y(c, c, s) = s; \quad (2.45)$$

$\beta = c_2$ ir $u_1 \neq u_2$ yra nesutampančios polinomo $c_2 y^2 + c_1 y + c_0 = 0$ šaknys. (Jeigu šaknys sutampa, lygtį galima suintegruoti atskyrus kintamuosius).

Parodysime, kaip gaunamas (2.44) naudojantis apibendrintojo diferencialinio operatoriaus metodu. Pirmiausia pakeičiame nepriklausomą kintamąjį pagal (2.31):

$$\eta \widehat{xy}'_x = \beta (\widehat{y} - u_1) (\widehat{y} - u_2), \quad (2.46)$$

kur $\widehat{y} = \widehat{y}(\widehat{x}, \widehat{c}, s)$ tenkina pradinę sąlygą $\widehat{y}(\widehat{c}, \widehat{c}, s) = s$.

Lygtį (2.46) atitinkantis apibendrintas diferencialinis operatorius lygus:

$$\widehat{\mathbf{D}} = \mathbf{D}_{\widehat{c}} + \frac{\beta (s - u_1) (s - u_2)}{\eta \widehat{c}} \mathbf{D}_s. \quad (2.47)$$

Tegul $\widehat{p}_j = \widehat{\mathbf{D}}^j s$, $j = 0, 1, \dots$, tuomet:

$$\begin{aligned} \widehat{p}_0 &= s, & \widehat{p}_1 &= \frac{\beta (s - u_1) (s - u_2)}{\eta \widehat{c}}, \\ \widehat{p}_2 &= \frac{\beta (s - u_1) (s - u_2) (2\beta s - \beta u_1 - \beta u_2 - \eta)}{\eta^2 \widehat{c}^2}, \dots \end{aligned} \quad (2.48)$$

Sekos $\left(\frac{1}{j!} \widehat{p}_j; j \in \mathbb{Z}_0\right)$ Hankelio determinantų seka lygi:

$$\begin{aligned} d_1 &= s, & d_2 &= \frac{\beta (s - u_1) (s - u_2) (\beta s u_1 + \beta s u_2 - 2\beta u_1 u_2 - \eta s)}{2\eta^2 \widehat{c}^2}, \\ d_j &= \frac{\eta^2 - \beta^2 (u_1 - u_2)^2}{\eta^{2j} \widehat{c}^{2j}} f_j(\widehat{c}, s, \eta), \end{aligned} \quad (2.49)$$

kur f_j yra polinamai, priklausantys nuo \widehat{c}, s, η . Įstačius $\eta := \beta (u_1 - u_2)$ gauname:

$$\text{order} \left(\frac{1}{j!} \widehat{p}_j; j \in \mathbb{Z}_0 \right) = 2. \quad (2.50)$$

Charakteristinė lygtis (2.24) lygi (su reikšme $\eta = \beta (u_1 - u_2)$):

$$\begin{vmatrix} s & \widehat{\mathbf{D}}s & \frac{1}{2} \widehat{\mathbf{D}}^2 s \\ \widehat{\mathbf{D}}s & \frac{1}{2} \widehat{\mathbf{D}}^2 s & \frac{1}{6} \widehat{\mathbf{D}}^3 s \\ 1 & \rho & \rho^2 \end{vmatrix} = 0. \quad (2.51)$$

Gauname (2.51) šaknis:

$$\rho_1 = 0; \quad \rho_2 = \frac{s - u_1}{\widehat{c} (u_1 - u_2)}, \quad (2.52)$$

tuomet $\widehat{\mathbf{D}}^j s$ gali būti užrašyti tokioje formoje:

$$\widehat{\mathbf{D}}^j s = j! \left(\lambda_1 0^j + \lambda_2 \left(\frac{s - u_1}{\widehat{c} (u_1 - u_2)} \right)^j \right). \quad (2.53)$$

Koeficientai λ_1, λ_2 (randami iš (2.27)) lygūs:

$$\lambda_1 = u_2, \quad \lambda_2 = s - u_2. \quad (2.54)$$

Taigi,

$$\begin{aligned} \hat{y}(\hat{x}, \hat{c}, s) &= \sum_{j=0}^{+\infty} \frac{(\hat{x} - \hat{c})^j}{j!} \hat{\mathbf{D}}^j s = \sum_{j=0}^{+\infty} (\hat{x} - \hat{c})^j \left(u_2 \theta^j + (s - u_2) \left(\frac{s - u_1}{\hat{c}(u_1 - u_2)} \right)^j \right) \\ &= \frac{u_1 (s - u_2) \hat{c} - u_2 (s - u_1) \hat{x}}{(s - u_2) \hat{c} - (s - u_1) \hat{x}}. \end{aligned} \quad (2.55)$$

Pasinaudoję keitiniu (2.36) ir elementariais algebriniais pertvarkiais gauname (2.44). Sprendimą įgyvendinanti programa gali būti randama C.3 priede.

2.2. Simbolinių pertvarkymų programinės įrangos paketai

Simbolinių pertvarkymų paketai²³ tai - programinė įranga, skirta manipuluoti matematinėmis išraiškomis. Šie paketai esmingai skiriasi nuo įprastų programavimo kalbų ir apytikslio skaičiavimo paketų tuo, kad dirba ne su skaitinėmis, o simbolinėmis kintamųjų reikšmėmis bei sugeba jomis manipuluoti naudojant tikslius metodus [42].

Nors simbolinius pertvarkymus gebanti atlikti programinė įranga pradėta kurti XX a. septyntajame dešimtmetyje, smarkiai ji išpopuliarėjo tik pastaraisiais metais. Tai galima paaiškinti greitu pastarųjų laikų progresu kuriant simbolinio skaičiavimo algoritmus ir darant paketus vis „protingesniais“. Su simbolinių pertvarkymų paketų išpopuliarėjimu iškilo ir ištisa klasė tikslų sprendimo metodų diferencialinėms lygtims spręsti. Ankstesniais laikais šiuos metodus kurti nebuvo praktiška, kadangi žmogui beveik neįmanoma nesuklydus sutvarkyti itin didelės apimties matematinių išraiškų [43].

Didžiąją šio darbo dalį sudaro simboliniai skaičiavimai, taigi turėjome išsirinkti programinės įrangos paketą, kurį jiems naudosime. Vieni iš populiariausių ir daugiausia funkcionalumo turinčių paketų yra Maple, Wolfram Mathematica, Maxima, MuPad (neseniai integruotas į Matlab). Neabejotini funkcionalumo lyderiai yra Maple ir Mathematica - moksliniuose straipsniuose neretai galima rasti nuorodas į šių programų tekstus. Taigi, kaip vieną dažniausiai minimų ir plačiausiai naudojamų paketų, pasirinkome Maple.

Simbolinių pertvarkymų paketą naudosime pagrindinių teiginių įrodymui bei pavyzdžių konstravimui, taip pat ir vizualizacijai - grafikų braižymui bei animacijų generavimui.

²³angl. *computer algebra system*

3. TYRIMO REZULTATAI

3.1. Atvirkštinio balansavimo metodas

Atvirkštinio balansavimo metodas naudojamas surasti būtinoms sąlygoms, kurias turi tenkinti diferencialinė lygčių sistema, kad galėtų turėtų nurodytos formos sprendinį. Jo esmę trumpai galima apibūdinti taip: žinomos formos sprendinys statomas į diferencialinių lygčių sistemą ir iš gautų (jau algebrinių) lygčių apskaičiuojamos galimos sistemos parametrų reikšmės bei papildomos sistemai keliamos sąlygos, kad joje egzistuotų postuluojamas sprendinys.

3.1.1. Antros eilės solitonai Rikati lygčių sistemoje

Kaip jau minėta įvade, spręsimė dviejų Rikati lygčių, susietų multiplikatyviuoju sąryšiu, sistema:

$$\begin{aligned} y'_x &= a_0 + a_1y + a_2y^2 + a_3yz; \\ z'_x &= b_0 + b_1z + b_2z^2 + b_3yz. \end{aligned} \quad (3.1)$$

Atvirkštinio balansavimo metodu rasime sąlygas, kurias turi tenkinti (3.1), kad turėtų sprendinį:

$$y(x) = \sigma \frac{(e^{\eta(x-c)} - y_1)(e^{\eta(x-c)} - y_2)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - x_2)}; \quad z(x) = \gamma \frac{(e^{\eta(x-c)} - z_1)(e^{\eta(x-c)} - z_2)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - x_2)}. \quad (3.2)$$

Toliau pertvarkant gautas lygtis patogiu naudoti pažymėjimus:

$$X(x) := (x - x_1)(x - x_2); \quad Y(x) := (x - y_1)(x - y_2); \quad Z(x) := (x - z_1)(x - z_2), \quad (3.3)$$

kur $x_1, x_2, y_1, y_2, z_1, z_2$ - bendru atveju kompleksiniai parametrai. Pastebėsime, kad

$$\begin{aligned} X(x_1) = X(x_2) = Y(y_1) = Y(y_2) = Z(z_1) = Z(z_2) = 0; \\ X(y_1)X(y_2) = Y(x_1)Y(x_2), \dots, Y(z_1)Y(z_2) = Z(y_1)Z(y_2). \end{aligned} \quad (3.4)$$

Tuomet (3.2) galima užrašyti:

$$y_0(x) = \sigma \frac{Y(e^{\eta(x-c)})}{X(e^{\eta(x-c)})}; \quad z_0(x) = \gamma \frac{Z(e^{\eta(x-c)})}{X(e^{\eta(x-c)})}. \quad (3.5)$$

Pasinaudoję 2.1.6 poskyryje aprašytu keitiniu, gauname (3.5) funkcijų vaizdą:

$$\hat{y}_0 = \hat{y}_0(\hat{x}) = \sigma \frac{Y\left(\frac{\hat{x}}{c}\right)}{X\left(\frac{\hat{x}}{c}\right)}; \quad \hat{z}_0 = \hat{z}_0(\hat{x}) = \gamma \frac{Z\left(\frac{\hat{x}}{c}\right)}{X\left(\frac{\hat{x}}{c}\right)}. \quad (3.6)$$

Tuo tarpu transformuota sistema (3.1) užrašoma:

$$\begin{aligned}\eta\widehat{x}(\widehat{y}_0)'_{\widehat{x}} &= a_0 + a_1\widehat{y}_0 + a_2\widehat{y}_0^2 + a_3\widehat{y}_0\widehat{z}_0; \\ \eta\widehat{x}(\widehat{z}_0)'_{\widehat{x}} &= b_0 + b_1\widehat{z}_0 + b_2\widehat{z}_0^2 + b_3\widehat{y}_0\widehat{z}_0.\end{aligned}\quad (3.7)$$

Pastebėsime, kad (3.5) tenkina (3.1) tada ir tik tada, kai (3.6) tenkina (3.7). Tarkime, kad (3.6) tenkina (3.7). Įstatę išraišką (3.6) į (3.7) ir suprastinę, gauname du polinomus:

$$\begin{aligned}&\eta\sigma\frac{\widehat{x}}{\widehat{c}}\left(\left(2\frac{\widehat{x}}{\widehat{c}} - (y_1 + y_2)\right)X\left(\frac{\widehat{x}}{\widehat{c}}\right) - \left(2\frac{\widehat{x}}{\widehat{c}} - (x_1 + x_2)\right)Y\left(\frac{\widehat{x}}{\widehat{c}}\right)\right) \\ &= a_0X^2\left(\frac{\widehat{x}}{\widehat{c}}\right) + a_1\sigma Y\left(\frac{\widehat{x}}{\widehat{c}}\right)X\left(\frac{\widehat{x}}{\widehat{c}}\right) + a_2\sigma^2Y^2\left(\frac{\widehat{x}}{\widehat{c}}\right) + a_3\sigma\gamma Y\left(\frac{\widehat{x}}{\widehat{c}}\right)Z\left(\frac{\widehat{x}}{\widehat{c}}\right); \\ &\eta\gamma\frac{\widehat{x}}{\widehat{c}}\left(\left(2\frac{\widehat{x}}{\widehat{c}} - (z_1 + z_2)\right)X\left(\frac{\widehat{x}}{\widehat{c}}\right) - \left(2\frac{\widehat{x}}{\widehat{c}} - (x_1 + x_2)\right)Z\left(\frac{\widehat{x}}{\widehat{c}}\right)\right) \\ &= b_0X^2\left(\frac{\widehat{x}}{\widehat{c}}\right) + b_1\gamma Z\left(\frac{\widehat{x}}{\widehat{c}}\right)X\left(\frac{\widehat{x}}{\widehat{c}}\right) + b_2\gamma^2Z^2\left(\frac{\widehat{x}}{\widehat{c}}\right) + b_3\gamma\sigma Z\left(\frac{\widehat{x}}{\widehat{c}}\right)Y\left(\frac{\widehat{x}}{\widehat{c}}\right).\end{aligned}\quad (3.8)$$

Paeiliui parinę $\widehat{x} = \widehat{c}x_1$; $\widehat{x} = \widehat{c}x_2$; $\widehat{x} = \widehat{c}y_1$; $\widehat{x} = \widehat{c}y_2$; $\widehat{x} = \widehat{c}z_1$; $\widehat{x} = \widehat{c}z_2$ ir, galiausiai, $\widehat{x} = 0$ ir įstatę (3.4) į (3.8) gauname tiesinių lygčių sistemą parametrų $a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3, x_1, x_2$ atžvilgiu:

$$\begin{aligned}(i) \quad &\sigma Y(x_1)a_2 + \gamma Z(x_1)a_3 = \eta x_1(x_2 - x_1); \\ (ii) \quad &\sigma Y(x_2)a_2 + \gamma Z(x_2)a_3 = \eta x_2(x_1 - x_2); \\ (iii) \quad &\gamma Z(x_1)b_2 + \sigma Y(x_1)b_3 = \eta x_1(x_2 - x_1); \\ (iv) \quad &\gamma Z(x_2)b_2 + \sigma Y(x_2)b_3 = \eta x_2(x_1 - x_2); \\ (v) \quad &a_0X^2(y_1) = \eta\sigma y_1(y_1 - y_2)X(y_1); \\ (vi) \quad &a_0X^2(y_2) = \eta\sigma y_2(y_2 - y_1)X(y_2); \\ (vii) \quad &b_0X^2(z_1) = \eta\gamma z_1(z_1 - z_2)X(z_1); \\ (viii) \quad &b_0X^2(z_2) = \eta\gamma z_2(z_2 - z_1)X(z_2); \\ (ix) \quad &(x_1x_2)^2 a_0 + \sigma x_1x_2y_1y_2a_1 + \sigma^2(y_1y_2)^2 a_2 + \sigma\gamma y_1y_2z_1z_2a_3 = 0; \\ (x) \quad &(x_1x_2)^2 b_0 + \gamma x_1x_2z_1z_2b_1 + \gamma^2(z_1z_2)^2 b_2 + \gamma\sigma y_1y_2z_1z_2b_3 = 0.\end{aligned}\quad (3.9)$$

Lygtys (i)-(iv) turi sprendinius:

$$a_3 = b_2 = \Xi_2\frac{\eta}{\gamma}; \quad b_3 = a_2 = \Theta_2\frac{\eta}{\sigma}, \quad (3.10)$$

kur

$$\begin{aligned}\Xi_2 &= \frac{(x_1 - x_2)(x_2Y(x_1) + x_1Y(x_2))}{Y(x_1)Z(x_2) - Y(x_2)Z(x_1)}; \\ \Theta_2 &= \frac{(x_2 - x_1)(x_1Z(x_2) + x_2Z(x_1))}{Y(x_1)Z(x_2) - Y(x_2)Z(x_1)}.\end{aligned}\quad (3.11)$$

Pažymėsime, kad šiame darbe nenagrinėjami išsigimę atvejai, kuomet sprendinių išraiškų vardikliuose esančios išraiškos tampa lygios nuliui. Tokių situacijų skaičius yra didžiulis, taigi

visas jas išnagrinėti būtų itin sudėtinga.

Lygtys (v)-(viii) iš (3.9) sistemos duoda tokius apribojimus parametrms y_1, y_2, z_1, z_2 :

$$\frac{X(y_1)}{X(y_2)} = -\frac{y_1}{y_2}; \quad \frac{X(z_1)}{X(z_2)} = -\frac{z_1}{z_2}. \quad (3.12)$$

Lygtis (3.12) gali būti perrašyta tokia forma:

$$\begin{aligned} 2y_1y_2(x_1 + x_2) - x_1x_2(y_1 + y_2) &= y_1y_2(y_1 + y_2); \\ 2z_1z_2(x_1 + x_2) - x_1x_2(z_1 + z_2) &= z_1z_2(z_1 + z_2). \end{aligned} \quad (3.13)$$

Pasinaudojus pažymėjimais:

$$x_1 + x_2 = -\frac{\Delta_1}{\Delta_0}; \quad x_1x_2 = \frac{\Delta_2}{\Delta_0}, \quad (3.14)$$

kur

$$\begin{aligned} \Delta_0 &:= 2(y_1y_2(z_1 + z_2) - z_1z_2(y_1 + y_2)); \\ \Delta_1 &:= (y_1 + y_2)(z_1 + z_2)(y_1y_2 - z_1z_2); \\ \Delta_2 &:= 2y_1y_2z_1z_2((y_1 + y_2) - (z_1 + z_2)), \end{aligned} \quad (3.15)$$

galima gauti, kad:

$$x_{1,2} = \frac{1}{2\Delta_0} \left(\Delta_1 \pm \sqrt{\Delta_1^2 - 4\Delta_0\Delta_2} \right) \in \mathbb{C}; \quad (3.16)$$

$$a_0 = \Theta_0\eta\sigma; \quad b_0 = \Xi_0\eta\gamma, \quad (3.17)$$

kur

$$\Theta_0 = \frac{y_1(y_1 - y_2)}{X(y_1)}; \quad \Xi_0 = \frac{z_1(z_1 - z_2)}{X(z_1)}. \quad (3.18)$$

Lieka apskaičiuoti a_1, b_1 pasinaudojant (3.10), (3.16), (3.17) ir (ix),(x) iš (3.9):

$$a_1 = -(\Theta_0 + \Theta_2 + \Xi_2)\eta; \quad b_1 = -(\Xi_0 + \Xi_2 + \Theta_2)\eta. \quad (3.19)$$

Atliktų skaičiavimų kontekste prasmingas toks apibrėžimas:

Apibrėžimas 3.1 *Solitoninių sprendinių pora (3.5) vadinama suderinta, jeigu jos parametrai x_1 ir x_2 tenkina (3.12). Rikati lygčių sistema (3.1) vadinama suderinta, jeigu tenkinami apribojimai (3.10).*

Analogiškai (3.6) ir (3.7) vadinami suderintais solitoninių sprendinių ir lygčių sistemos vaizdas, jeigu galuoja (3.10) ir (3.12).

Išvada 3.1 *Jei (3.5) tenkina (3.1), tai (3.5) ir (3.1) būtinai turi būti suderinti.*

Išvada 3.2 *Jei (3.5) tenkina (3.1), tai sistema turi tokią formą:*

$$\begin{aligned} (y_0)'_x &= \Theta_0\eta\sigma - (\Theta_0 + \Theta_2 + \Xi_2)\eta y_0 + \Theta_2\frac{\eta}{\sigma}y_0^2 + \Xi_2\frac{\eta}{\gamma}y_0z_0; \\ (z_0)'_x &= \Xi_0\eta\gamma - (\Xi_0 + \Xi_2 + \Theta_2)\eta z_0 + \Xi_2\frac{\eta}{\gamma}z_0^2 + \Theta_2\frac{\eta}{\sigma}y_0z_0. \end{aligned} \quad (3.20)$$

Tarkime, kad funkcijų (3.5) parametrai $(\eta, \sigma, y_1, y_2, \gamma, z_1, z_2)$ - fiksuoti. Tuomet pagalbiniai parametrai $(x_1, x_2, \Theta_0, \Theta_2, \Xi_0, \Xi_2)$ gali būti apskaičiuoti aprašytu algoritmu. Tokiu pat būdu galima rasti sistemos (3.1) parametrus $(a_0, a_1, a_2, b_0, b_1, b_2)$.

Šiame skyriuje atlikti skaičiavimai realizuoti Maple. Programos tekstas - C.4 priede.

3.1.2. Formalūs sąryšiai tarp solitoninio sprendinio ir Rikati sistemos parametrų

Pasinaudojant 3.1.1 poskyrio rezultatais, perėjimas nuo sprendinio parametrų prie sistemos parametrų gali būti aprašomas tokiomis atitiktimis:

1. Pagalbinių parametrų apskaičiavimas:

$$(y_1, y_2, z_1, z_2) \rightarrow (x_1, x_2, \Theta_0, \Theta_2, \Xi_0, \Xi_2). \quad (3.21)$$

2. Sistemos parametrų apskaičiavimas:

$$(\eta, \sigma, y_1, y_2, \gamma, z_1, z_2) \rightarrow (a_0, a_1, a_2, b_0, b_1, b_2). \quad (3.22)$$

Toliau sekančios išvados gaunamos iš 3.1.1 skyrelio rezultatų.

Išvada 3.3 Iš (3.11), (3.12) ir (3.17) išplaukia, kad:

$$(\psi y_1, \psi y_2, \psi z_1, \psi z_2) \rightarrow (\psi x_1, \psi x_2, \Theta_0, \Theta_2, \Xi_0, \Xi_2); \quad \psi \in \mathbb{R} \setminus \{0\}, \quad (3.23)$$

Išvada 3.4 Iš (3.11), (3.12) ir (3.20) gaunama:

$$(\phi \eta, \alpha \sigma, \psi y_1, \psi y_2, \beta \gamma, \psi z_1, \psi z_2) \rightarrow \left(\phi \alpha a_0, \phi a_1, \frac{\phi}{\alpha} a_2, \phi \beta b_0, \phi b_1, \frac{\phi}{\beta} b_2 \right); \quad (3.24)$$

$\phi, \alpha, \beta, \psi \in \mathbb{R} \setminus \{0\}$.

Išvada 3.5 Iš (3.23) ir (3.24), kartu su

$$\begin{aligned} y_0(x) &= \sigma \frac{y_1 y_2}{x_1 x_2} \frac{\left(e^{-\eta(x-c)} - \frac{1}{y_1} \right) \left(e^{-\eta(x-c)} - \frac{1}{y_2} \right)}{\left(e^{-\eta(x-c)} - \frac{1}{x_1} \right) \left(e^{-\eta(x-c)} - \frac{1}{x_2} \right)}, \\ z_0(x) &= \gamma \frac{z_1 z_2}{x_1 x_2} \frac{\left(e^{-\eta(x-c)} - \frac{1}{z_1} \right) \left(e^{-\eta(x-c)} - \frac{1}{z_2} \right)}{\left(e^{-\eta(x-c)} - \frac{1}{x_1} \right) \left(e^{-\eta(x-c)} - \frac{1}{x_2} \right)}, \end{aligned} \quad (3.25)$$

išplaukia:

$$\left(-\eta, \sigma \frac{y_1 y_2}{x_1 x_2}, \frac{1}{y_1}, \frac{1}{y_2}, \gamma \frac{z_1 z_2}{x_1 x_2}, \frac{1}{z_1}, \frac{1}{z_2} \right) \rightarrow (a_0, a_1, a_2, b_0, b_1, b_2). \quad (3.26)$$

3.1.3. Solitoninių sprendinių fazinės trajektorijos

Teorema 3.1 Solitoninių sprendinių trajektorija fazinėje plokštumoje (y_0, z_0) tenkina antros eilės kreivės lygtį:

$$\widehat{A}y_0^2 + 2\widehat{B}y_0z_0 + \widehat{C}z_0^2 + \widehat{D}y_0 + \widehat{E}z_0 = 1. \quad (3.27)$$

Irodymas. Įstatę (3.6) į (3.27) ir padauginę iš $\left(X\left(\frac{\widehat{x}}{\widehat{c}}\right)\right)^2$ gauname daugianarių lygybę:

$$\begin{aligned} \sigma^2 Y^2\left(\frac{\widehat{x}}{\widehat{c}}\right) \widehat{A} + 2\sigma\gamma Y\left(\frac{\widehat{x}}{\widehat{c}}\right) Z\left(\frac{\widehat{x}}{\widehat{c}}\right) \widehat{B} + \gamma^2 Z^2\left(\frac{\widehat{x}}{\widehat{c}}\right) \widehat{C} + \\ \sigma Y\left(\frac{\widehat{x}}{\widehat{c}}\right) X\left(\frac{\widehat{x}}{\widehat{c}}\right) \widehat{D} + \gamma Z\left(\frac{\widehat{x}}{\widehat{c}}\right) X\left(\frac{\widehat{x}}{\widehat{c}}\right) \widehat{E} - X^2\left(\frac{\widehat{x}}{\widehat{c}}\right) = 0. \end{aligned} \quad (3.28)$$

Paėmę $\widehat{x} = \widehat{c}x_1; \widehat{x} = \widehat{c}x_2; \widehat{x} = \widehat{c}y_1; \widehat{x} = \widehat{c}y_2; \widehat{x} = \widehat{c}z_1; \widehat{x} = \widehat{c}z_2$ ir įstatę į (3.28) gauname tiesinę lygčių sistemą kintamųjų $\widehat{A}, \widehat{B}, \widehat{C}, \widehat{D}, \widehat{E}$ atžvilgiu:

$$\begin{aligned} \text{(i)} \quad & \gamma^2 Z^2(y_1) \widehat{C} + \gamma Z(y_1) X(y_1) \widehat{E} = X^2(y_1); \\ \text{(ii)} \quad & \gamma^2 Z^2(y_2) \widehat{C} + \gamma Z(y_2) X(y_2) \widehat{E} = X^2(y_2); \\ \text{(iii)} \quad & \sigma^2 Y^2(z_1) \widehat{A} + \sigma Y(z_1) X(z_1) \widehat{D} = X^2(z_1); \\ \text{(iv)} \quad & \sigma^2 Y^2(z_2) \widehat{A} + \sigma Y(z_2) X(z_2) \widehat{D} = X^2(z_2); \\ \text{(v)} \quad & \sigma^2 Y^2(x_1) \widehat{A} + 2\sigma\gamma Y(x_1) Z(x_1) \widehat{B} + \gamma^2 Z^2(x_1) \widehat{C} = 0; \\ \text{(vi)} \quad & \sigma^2 Y^2(x_2) \widehat{A} + 2\sigma\gamma Y(x_2) Z(x_2) \widehat{B} + \gamma^2 Z^2(x_2) \widehat{C} = 0; \end{aligned} \quad (3.29)$$

Pastebėsime, kad iš 6 lygčių viena išreiškiama kitų tiesiniu dariniu, taigi lygčių ir kintamųjų skaičius sutampa. Neišsigimusiu atveju (kai antros eilės kreivė nevirsta tiese), sistemos (3.29) matricos determinantas nenulinis, taigi egzistuoja vienintelis sprendinys:

$$\begin{aligned} \widehat{A} = -\frac{X(z_1)X(z_2)}{\sigma^2 Y(z_1)Y(z_2)}; \quad \widehat{C} = -\frac{X(y_1)X(y_2)}{\gamma^2 Z(y_1)Z(y_2)}; \quad \widehat{B} = \frac{Y(x_1)Z(x_2) + Y(x_2)Z(x_1)}{\sigma\gamma(Y(z_1)Y(z_2) + Z(y_1)Z(y_2))}; \\ \widehat{D} = \frac{Y(z_1)X(z_2) + Y(z_2)X(z_1)}{\sigma Y(z_1)Y(z_2)}; \quad \widehat{E} = \frac{Z(y_1)X(y_2) + Z(y_2)X(y_1)}{\gamma Z(y_1)Z(y_2)} \blacksquare \end{aligned} \quad (3.30)$$

Antros eilės kreivės (3.27) diskriminantas išreiškiamas lygybe [44]:

$$\widehat{A}\widehat{C} - \widehat{B}^2 = -\frac{(Z(x_2)Y(x_1) - Y(x_2)Z(x_1))^2}{4(Y(z_1)Y(z_2))^2}. \quad (3.31)$$

Taigi, antros eilės kreivės rūšis (parabolė, hiperbolė, elipsė) priklauso nuo reiškinio

$$\tau = -(Z(x_2)Y(x_1) - Y(x_2)Z(x_1))^2, \quad (3.32)$$

ženklo, kadangi (3.31) vardiklis yra teigiamas, bet skaitiklis bendru atveju yra realaus arba menamo skaičiaus kvadratas. Tai galima patikrinti pasinaudojus simbolinių pertvarkymų paketu, parodant,

kad τ ženklas lygus τ_0 ženkliui, kuris apibrėžiamas:

$$\begin{aligned} \tau_0 := 4\Delta_0\Delta_2 - \Delta_1^2 = 16y_1y_2z_1z_2 \left((y_1 + y_2) - (z_1 + z_2) \right) & \left(y_1y_2(z_1 + z_2) \right. \\ & \left. - z_1z_2(y_1 + y_2) \right) + (y_1 + y_2)^2(z_1 + z_2)^2(y_1y_2 - z_1z_2)^2. \end{aligned} \quad (3.33)$$

Taigi, (3.27) yra elipsės fragmentas, jeigu $\tau_0 > 0$; parabolės fragmentas, jeigu $\tau_0 = 0$; hiperbolės fragmentas, jeigu $\tau_0 < 0$.

Šiame skyriuje atlikti skaičiavimai realizuoti Maple. Programos tekstas - C.5 priede.

3.2. Operatorinis metodas

Operatorinis metodas papildo atvirkštinio balansavimo metodą - jį galima naudoti pakankamoms solitoninių sprendinių (3.2) Rikati lygčių sistemoje su multiplikatyviuoju sąryšiu (3.1) rasti. Kitaip tariant, šiuo metodu bus sprendžiama diferencialinė lygtis, atitinkanti būtinas solitoninių sprendinių egzistavimo sąlygas. Šio proceso rezultate gausime sprendinių parametrų $x_1, x_2, \sigma, y_1, y_2, \gamma, z_1, z_2$ išraiškas.

Turime pažymėti, kad jeigu atvirkštinio balansavimo metode dar nėra neišvengiamas simbolių pertvarkymų paketo naudojimas, operatoriniame skaičiavime be jo išsiversti nebepavyksta. Todėl šiame skyriuje pateiksime nuorodas į programas, realizuojančius vienus ar kitus skaičiavimus. Plačiau pačių programų sandara aptariama 4 skyriuje.

3.2.1. Diferencialinių lygčių sistemos sprendinių operatorinės išraiškos

Šiame poskyryje apibendrinsime 2.1 skyrelio teiginius diferencialinių lygčių sistemoms.

Tarkime, duota dviejų pirmos eilės paprastųjų diferencialinių lygčių sistema:

$$\begin{aligned} y'_x &= P(x, y, z); \\ z'_x &= Q(x, y, z), \end{aligned} \quad (3.34)$$

kur funkcijos $P = P(c, s, t)$, $Q = Q(c, s, t) \in \Phi_{c,s,t}$.

Bendrasis (3.34) sprendinys yra funkcijų pora

$$y = y(x, c, s, t); \quad z = z(x, c, s, t), \quad (3.35)$$

tenkinanti pradines sąlygas taške $x = c$:

$$y(c, c, s, t) = s; \quad z(c, c, s, t) = t. \quad (3.36)$$

Šią sistemą atitinka apibendrintasis diferencialinis operatorius:

$$\mathbf{D} = \mathbf{D}_c + P(c, s, t)\mathbf{D}_s + Q(c, s, t)\mathbf{D}_t. \quad (3.37)$$

Tuomet operatorinė (3.35) išraiška yra:

$$y = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s; \quad z = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j t. \quad (3.38)$$

Pastebėsime, kad, jeigu $\left(\frac{1}{j!} \mathbf{D}^j s; j \in \mathbb{Z}_0\right)$ ir $\left(\frac{1}{j!} \mathbf{D}^j t; j \in \mathbb{Z}_0\right)$ yra tiesiškai rekurentinės sekos su charakteristinėmis šaknimis $\rho_0 = 0, \rho_k = \rho_k(c, s, t), k = 1, 2$, tai galioja lygybės:

$$\frac{1}{j!} \mathbf{D}^j s = \lambda_0 \theta^j + \lambda_1 \rho_1^j + \lambda_2 \rho_2^j; \quad \frac{1}{j!} \mathbf{D}^j t = \mu_0 \theta^j + \mu_1 \rho_1^j + \mu_2 \rho_2^j. \quad (3.39)$$

Analogiškai 2.3 teoremai, jeigu tenkinamos sąlygos $\mathbf{D}\rho_k = \rho_k^2, k = 1, 2$ and $\mathbf{D}\lambda_l = \lambda_l \rho_l; \mathbf{D}\mu_l = \mu_l \rho_l, l = 0, 1, 2$, tai sprendiniai užrašomi:

$$y = \lambda_0 + \frac{\lambda_1}{1 - \rho_1(x-c)} + \frac{\lambda_2}{1 - \rho_2(x-c)}; \quad (3.40)$$

$$z = \mu_0 + \frac{\mu_1}{1 - \rho_1(x-c)} + \frac{\mu_2}{1 - \rho_2(x-c)},$$

be to,

$$\lambda_0 + \lambda_1 + \lambda_2 = s; \quad \mu_0 + \mu_1 + \mu_2 = t. \quad (3.41)$$

3.2.2. Specialiosios Hankelio determinantų išraiškos

Suderintos Rikati lygčių, susietų multiplikatyviuoju sąryšiu, sistemos vaizdas yra:

$$\eta \widehat{x}(\widehat{y})'_x = a_0 + a_1 \widehat{y} + a_2 \widehat{y}^2 + b_2 \widehat{y} \widehat{z}; \quad (3.42)$$

$$\eta \widehat{x}(\widehat{z})'_x = b_0 + b_1 \widehat{z} + b_2 \widehat{z}^2 + a_2 \widehat{y} \widehat{z}.$$

Atkreipsime dėmesį, kad, pagal 3.1.1 poskyrio rezultatus, čia $b_3 = a_2$ ir $a_3 = b_2$.

Bendrasis (3.42) sprendinys žymimas:

$$\widehat{y} = \widehat{y}(\widehat{x}, \widehat{c}, s, t); \quad \widehat{z} = \widehat{z}(\widehat{x}, \widehat{c}, s, t). \quad (3.43)$$

Sprendinys tenkina pradines sąlygas

$$\widehat{y}(\widehat{c}, \widehat{c}, s, t) = s; \quad \widehat{z}(\widehat{c}, \widehat{c}, s, t) = t, \quad (3.44)$$

ir operatorinėje formoje yra išreiškiamas tokiu būdu:

$$\widehat{y} = \sum_{j=0}^{+\infty} \frac{(\widehat{x} - \widehat{c})^j}{j!} \mathbf{D}^j s; \quad \widehat{z} = \sum_{j=0}^{+\infty} \frac{(\widehat{x} - \widehat{c})^j}{j!} \mathbf{D}^j t, \quad (3.45)$$

čia \mathbf{D} - apibendrintasis diferencialinis sistemos (3.42) operatorius, apibrėžiamas (pagal (3.37)):

$$\mathbf{D} := \mathbf{D}_{\widehat{c}} + \frac{1}{\eta \widehat{c}} \left((a_0 + a_1 s + a_2 s^2 + b_2 s t) \mathbf{D}_s + (b_0 + b_1 t + b_2 t^2 + a_2 s t) \mathbf{D}_t \right). \quad (3.46)$$

Pradėdami spręsti sistemą pirmiausia apskaičiuojame sekos narius (Programa P222):

$$\widehat{p}_j := \frac{1}{j!} \mathbf{D}^j s; \quad \widehat{q}_j := \frac{1}{j!} \mathbf{D}^j t; \quad j = 0, 1, \dots, 5. \quad (3.47)$$

Pastebėsime, kad kadangi η operatoriaus (3.46) atžvilgiu yra konstanta, galioja lygybės:

$$\left(\widehat{p}_j \right) \Big|_{\eta=\eta_0} = \left(\mathbf{D}^j s \right) \Big|_{\eta=\eta_0} = \left((\mathbf{D}^j) s \right) \Big|_{\eta=\eta_0} = \left((\mathbf{D}^j) \Big|_{\eta=\eta_0} s \right), \quad (3.48)$$

ir

$$\left(\widehat{q}_j \right) \Big|_{\eta=\eta_0} = \left(\mathbf{D}^j t \right) \Big|_{\eta=\eta_0} = \left((\mathbf{D}^j) t \right) \Big|_{\eta=\eta_0} = \left((\mathbf{D}^j) \Big|_{\eta=\eta_0} t \right). \quad (3.49)$$

Apskaičiuojame Hankelio determinantus (Programa P222):

$$W_{\widehat{y}} := W_{\widehat{y}}(\widehat{c}, s, t, \eta) = \begin{vmatrix} \widehat{p}_1 & \widehat{p}_2 & \widehat{p}_3 \\ \widehat{p}_2 & \widehat{p}_3 & \widehat{p}_4 \\ \widehat{p}_3 & \widehat{p}_4 & \widehat{p}_5 \end{vmatrix}; \quad W_{\widehat{z}} := W_{\widehat{z}}(\widehat{c}, s, t, \eta) = \begin{vmatrix} \widehat{q}_1 & \widehat{q}_2 & \widehat{q}_3 \\ \widehat{q}_2 & \widehat{q}_3 & \widehat{q}_4 \\ \widehat{q}_3 & \widehat{q}_4 & \widehat{q}_5 \end{vmatrix}. \quad (3.50)$$

Jeigu (3.42) sprendiniai yra antros eilės solitonai (su visomis pradinėmis sąlygomis $s, t \in \mathbb{R}$), būtina, kad determinantai (3.50) būtų lygūs nuliui (pagal 2.1.6 poskyrį).

Kad egzistuotų antros eilės solitoniniai sprendiniai, turi egzistuoti toks $\eta_0 \in \mathbb{R}$, kad

$$W_{\widehat{y}}(\widehat{c}, s, t, \pm \eta_0) = 0; \quad W_{\widehat{z}}(\widehat{c}, s, t, \pm \eta_0) = 0, \quad (3.51)$$

su visomis $\widehat{c}, s, t \in \mathbb{R}$ reikšmėmis. Tai, kad tinka ir neigiama, ir teigiama η_0 reikšmė galima suprasti iš 3.1.2 poskyrio (3.26) lygybės.

Panagrinėkime determinantą . Nesunku įsitikinti (Programa P222), kad:

$$W_{\widehat{y}} = A_6(s, t) \eta^6 + A_4(s, t) \eta^4 + A_2(s, t) \eta^2 + A_0(s, t), \quad (3.52)$$

čia $A_k(s, t)$ - daugianariai, priklausantys nuo s ir t . Taigi, ieškome tokių (3.52) sprendinių η atžvilgiu, kurie nepriklauso nuo s ir t .

Panagrinėkime (3.52) struktūrą. Gerai žinoma Vieto teorema leidžia išskaidyti daugianarį su skaitiniais koeficientais į daugiklius. Kadangi (3.52) koeficientai nėra skaičiai, o daugianariai, išskaidymo būdų iškart padaugėja. Reikia pastebėti, kad skaidinyje gali dalyvauti ir ne daugianariai, o radikalai, bet skaidinį išskleidus jo koeficientai turi būti daugianariai. Trečios eilės daugianario bendriausias skaidinys yra:

$$\left(K\sqrt{L}\eta^2 - (K\sqrt{L} + \sqrt{E}) \right) \left(K\sqrt{L}\eta^2 - (K\sqrt{L} - \sqrt{E}) \right) (B\eta^2 - C) = 0, \quad (3.53)$$

čia B, C, E, K, L - daugianariai, priklausantys nuo s, t .

Pastebėsime, kad jeigu (3.52) turi šaknis $\pm\eta_0$, tai vienas iš jo skaidinio daugiklių privalo būti $(\eta^2 - \eta_0)$. Be to, nesunku apskaičiuoti (Programa P222), kad:

$$A_6 = \frac{(s^2 a_2 + (tb_2 + a_1)s + a_0)^3}{2160}, \quad (3.54)$$

taigi $A_6 = K^3$, kur $K = \left(\frac{1}{2160}\right)^{1/3} (s^2 a_2 + (tb_2 + a_1)s + a_0)$. Tuomet turime (3.53) atvejį, kai $K = L$, taigi ieškomas polinomo skaidinys turi tokią formą:

$$(K^3 \eta^4 + B_2 \eta^2 + B_0) (\eta^2 - \eta_0^2) = 0. \quad (3.55)$$

Prieštaros metodu nesunku parodyti, kad B_2, B_0 irgi yra polinamai, o ne radikalai. Sudauginę gauname:

$$K^3 \eta^6 + (-K^3 \eta_0^2 + B_2) \eta^4 + (-B_2 \eta_0^2 + B_0) \eta^2 - B_0 \eta_0^2 = A_6 \eta^6 + A_4 \eta^4 + A_2 \eta^2 + A_0 = 0. \quad (3.56)$$

Kita vertus, įrodoma, kad

$$A_4 = 9K \cdot A'_4(s, t, a_0, \dots, b_2), \quad (3.57)$$

kur A'_4 - daugianaris. Prie prielaidos, kad galioja (3.55) gauname, kad koeficientas $(-K^3 \eta_0^2 + B_2)$ turi dalintis iš K , taigi

$$-K^3 \eta_0^2 + B_2 = -K^3 \eta_0^2 + K \cdot B'_2. \quad (3.58)$$

Nesunku pastebėti iš K struktūros, kad jeigu tik a_2, a_1, b_2 nėra kartu lygūs nuliui, galima rasti tokias reikšmes s_0, t_0 , kad $K \Big|_{s=s_0, t=t_0} = 0$, pavyzdžiui, jeigu $b_2 \neq 0$, tai galima imti

$$t_0 = -\frac{a_2 s_0^2 + a_1 s_0 + a_0}{b_2 s_0}, s_0 \in \mathbb{R}. \quad (3.59)$$

Irašę šias reikšmes į (3.56) gausime:

$$B_0 \eta^2 - B_0 \eta_0^2 = 0 \Rightarrow \eta = \pm \eta_0. \quad (3.60)$$

Vadinasi, įstačius reikšmes (3.59) į (3.52) gausime, kad $A_6 = A_4 = 0$ nepriklausomai nuo to, ar egzistuoja solitoniniai sprendiniai ar ne. Tačiau likusio daugianario laisvasis narys, padalytas iš koeficiento prie η^2 yra pastovus ir lygus $-\eta_0^2$. Formaliai galima užrašyti šią sąlygą tokiu būdu:

$$\frac{A_0(s_0, t_0)}{A_2(s_0, t_0)} = -\eta_0^2 = \text{const}. \quad (3.61)$$

Kadangi A_0, A_2 žinomi, gauname (Programa P222):

$$\frac{\left(\frac{19}{10}a_2a_0(a_1 - \frac{23}{19}b_1) + (-\frac{4}{5}b_0b_2 - \frac{8}{15}a_1^2 + \frac{8}{15}a_1b_1 + \frac{1}{15}b_1^2)(a_1 - \frac{1}{2}b_1)\right)s_0 + \frac{3}{2}a_0(a_0a_2 + b_0b_2 - \frac{1}{3}a_1^2 + \frac{1}{3}a_1b_1 - \frac{1}{3}b_1^2)}{(a_1 - \frac{1}{2}b_1)s_0 + \frac{3}{2}a_0} = -\eta_0^2. \quad (3.62)$$

Pažymėję koeficientus trumpiau (3.62) užrašome:

$$\frac{\alpha_1 s_0 + \beta_1}{\alpha_2 s_0 + \beta_2} = -\eta_0^2. \quad (3.63)$$

Analogiškus veiksmus įvykdę su determinantu $W_{\bar{z}}$, gauname tokią sąlygą (Programa P222):

$$-\frac{4}{5} \frac{\left(-\frac{23}{4}(a_1 - \frac{19}{23}b_1)b_2 b_0 + (a_0 a_2 - \frac{1}{12}a_1^2 - \frac{2}{3}a_1 b_1 + \frac{2}{3}b_1^2)(a_1 - 2b_1)\right)t_0 + \frac{15}{4}b_0(a_0 a_2 + b_0 b_2 - \frac{1}{3}a_1^2 + \frac{1}{3}a_1 b_1 - \frac{1}{3}b_1^2)}{(a_1 - 2b_1)t_0 - 3b_0} = -\eta_0^2. \quad (3.64)$$

Pažymėję (3.64) koeficientus, galime užrašyti:

$$\frac{\alpha'_1 t_0 + \beta'_1}{\alpha'_2 t_0 + \beta'_2} = -\eta_0^2. \quad (3.65)$$

Iš (3.63) it (3.65) gauname:

$$\frac{\alpha_1 s_0 + \beta_1}{\alpha_2 s_0 + \beta_2} = \frac{\alpha'_1 t_0 + \beta'_1}{\alpha'_2 t_0 + \beta'_2} = -\eta_0^2, \quad (3.66)$$

taigi trupmenos turi būti lygios tarpusavyje ir nepriklausyti nuo s_0, t_0 . Ši sąlyga tenkinama, jeigu

$$R := 9a_0 a_1 a_2 + 9b_0 b_1 b_2 - 18a_0 a_2 b_1 - 18b_0 b_2 a_1 + 3a_1 b_1^2 + 3b_1 a_1^2 - 2a_1^3 - 2b_1^3 = 0, \quad (3.67)$$

nes

$$\alpha_1 \beta_2 - \alpha_2 \beta_1 = 135a_0 R; \quad \alpha'_1 \beta'_2 - \alpha'_2 \beta'_1 = 135b_0 R. \quad (3.68)$$

Taip pat gauname, kad:

$$\eta_0^2 = \frac{a_1^2 - a_1 b_1 + b_1^2}{3} - a_0 a_2 - b_0 b_2. \quad (3.69)$$

Šiame poskyryje minimos programos P222 tekstą galima rasti C.6 priede.

3.2.3. Antros eilės solitoninio sprendinio koeficientų sekų kanoninių išraiškų konstravimas

Šiame ir toliau sekančiuose poskyriuose parametras η laikomas fiksuotu ir žinomu, t.y. $\eta = \eta_0 > 0$ (galima naudoti ir $-\eta_0$ pagal (3.51)).

Charakteristinės sekų $(\hat{p}_j; j \in \mathbb{Z}_0)$ ir $(\hat{q}_j; j \in \mathbb{Z}_0)$ šaknys $\hat{\rho}_1, \hat{\rho}_2$ apskaičiuojamos išsprendus algebrinių lygčių sistemą (Programa P223):

$$\begin{vmatrix} \hat{p}_1 & \hat{p}_2 & \hat{p}_3 \\ \hat{p}_2 & \hat{p}_3 & \hat{p}_4 \\ 1 & \hat{\rho} & \hat{\rho}^2 \end{vmatrix} = 0; \quad \begin{vmatrix} \hat{q}_1 & \hat{q}_2 & \hat{q}_3 \\ \hat{q}_2 & \hat{q}_3 & \hat{q}_4 \\ 1 & \hat{\rho} & \hat{\rho}^2 \end{vmatrix} = 0. \quad (3.70)$$

Simboliniai pertvarkymai įrodo, kad abi (3.70) lygtys turi tuos pačius sprendinius $\hat{\rho}_k := \hat{\rho}_k(c, s, t), k = 1, 2$, kurių forma yra:

$$\hat{\rho}_k(\hat{c}, s, t) = \frac{1}{\hat{c}} \rho_k(s, t); \quad k = 1, 2, \quad (3.71)$$

čia $\rho_k := \rho_k(s, t)$ nepriklauso nuo \hat{c} .

Tiesiškai rekurentinės sekos $(\hat{p}_j; j \in \mathbb{N})$ koeficientai, rasti iš tiesinių lygčių sistemos (Prog-

rama P223) lygūs:

$$\lambda_1 = \frac{\widehat{\rho}_2 - \widehat{\rho}_1 \widehat{\rho}_2}{\widehat{\rho}_1 (\widehat{\rho}_1 - \widehat{\rho}_2)}; \quad \lambda_2 = \frac{\widehat{\rho}_2 - \widehat{\rho}_1 \widehat{\rho}_1}{\widehat{\rho}_2 (\widehat{\rho}_2 - \widehat{\rho}_1)}. \quad (3.72)$$

Analogiškai, sekos $(\widehat{q}_j; j \in \mathbb{N})$ koeficientai lygūs:

$$\mu_1 = \frac{\widehat{q}_2 - \widehat{q}_1 \widehat{\rho}_2}{\widehat{\rho}_1 (\widehat{\rho}_1 - \widehat{\rho}_2)}; \quad \mu_2 = \frac{\widehat{q}_2 - \widehat{q}_1 \widehat{\rho}_1}{\widehat{\rho}_2 (\widehat{\rho}_2 - \widehat{\rho}_1)}, \quad (3.73)$$

Pažymėsime, kad $\lambda_1, \lambda_2, \mu_1, \mu_2$ nepriklauso nuo \widehat{c} .

Patikrinamos sąlygos, kurios privalo būti tenkinamos, jeigu (3.42) sprendiniai yra solitoninių funkcijų vaizdai visiems $s, t \in \mathbb{R}$ (Programa P223):

$$\begin{aligned} \text{(i)} \quad & \lambda_1 + \lambda_2 = s - \sigma; \quad \mu_1 + \mu_2 = t - \gamma. \\ \text{(ii)} \quad & \mathbf{D}\widehat{\rho}_k = \widehat{\rho}_k^2; \quad \mathbf{D}\lambda_k = \lambda_k \widehat{\rho}_k; \quad \mathbf{D}\mu_k = \mu_k \widehat{\rho}_k, \end{aligned} \quad (3.74)$$

kur $k = 1, 2$.

Pastebėsime, kad iš (3.51), (3.71) ir (3.74) išplaukia, kad

$$\widehat{\rho}_j = \sigma \theta^j + \lambda_1 \widehat{\rho}_1^j + \lambda_2 \widehat{\rho}_2^j; \quad \widehat{q}_j = \gamma \theta^j + \mu_1 \widehat{\rho}_1^j + \mu_2 \widehat{\rho}_2^j; \quad j = 0, 1, \dots \quad (3.75)$$

Šiame poskyryje minimos programos P223 tekstą galima rasti C.7 priede.

3.2.4. Uždara antros eilės solitoninio sprendinio forma

Kai galioja (3.75) pagal (3.40), bendrasis sistemos (3.42) sprendinys $\widehat{y} = \widehat{y}(\widehat{x}, \widehat{c}, s, t)$, $\widehat{z} = \widehat{z}(\widehat{x}, \widehat{c}, s, t)$ įgyja tokią formą:

$$\widehat{y} = \sigma + \frac{\lambda_1}{1 - \widehat{\rho}_1(\widehat{x} - \widehat{c})} + \frac{\lambda_2}{1 - \widehat{\rho}_2(\widehat{x} - \widehat{c})}; \quad \widehat{z} = \gamma + \frac{\mu_1}{1 - \widehat{\rho}_1(\widehat{x} - \widehat{c})} + \frac{\mu_2}{1 - \widehat{\rho}_2(\widehat{x} - \widehat{c})}. \quad (3.76)$$

Pagal keitinį (2.31), atitinkamos suderintos sistemos

$$\begin{aligned} y'_x &= a_0 + a_1 y + a_2 y^2 + b_2 y z; \\ z'_x &= b_0 + b_1 z + b_2 z^2 + a_2 y z, \end{aligned} \quad (3.77)$$

sprendinys turi tokią formą:

$$\begin{aligned} y &= y(x, c, s, t) = \sigma + \frac{\lambda_1(s, t)}{1 - \rho_1(s, t) (e^{\eta_0(x-c)} - 1)} + \frac{\lambda_2(s, t)}{1 - \rho_2(s, t) (e^{\eta_0(x-c)} - 1)}; \\ z &= z(x, c, s, t) = \gamma + \frac{\mu_1(s, t)}{1 - \rho_1(s, t) (e^{\eta_0(x-c)} - 1)} + \frac{\mu_2(s, t)}{1 - \rho_2(s, t) (e^{\eta_0(x-c)} - 1)}, \end{aligned} \quad (3.78)$$

kurią galima pertvarkyti į:

$$y(x, c, s, t) = \sigma \frac{Y_* (e^{\eta_0(x-c)}, s, t)}{X_* (e^{\eta_0(x-c)}, s, t)}; \quad z(x, c, s, t) = \gamma \frac{Z_* (e^{\eta_0(x-c)}, s, t)}{X_* (e^{\eta_0(x-c)}, s, t)}, \quad (3.79)$$

čia X_*, Y_*, Z_* pagalbinės funkcijos:

$$\begin{aligned} X_*(x, s, t) &:= (x - x_1(s, t))(x - x_2(s, t)); \\ Y_*(x, s, t) &:= (x - y_1(s, t))(x - y_2(s, t)); \\ Z_*(x, s, t) &:= (x - z_1(s, t))(x - z_2(s, t)). \end{aligned} \quad (3.80)$$

Parametrai $x_k(s, t), y_k(s, t), z_k(s, t); k = 1, 2$ lygūs (Programa P224):

$$x_k = 1 + \frac{1}{\rho_k}; y_{1,2} = \frac{1}{2} \left(L_y \pm \sqrt{L_y^2 - 4K_y} \right); \quad z_{1,2} = \frac{1}{2} \left(L_z \pm \sqrt{L_z^2 - 4K_z} \right), \quad (3.81)$$

kur K_y, K_z, L_y, L_z - funkcijos, priklausančios nuo s, t , kurios tenkina lygybes:

$$\begin{aligned} y_1(s, t)y_2(s, t) &= K_y(s, t); \quad z_1(s, t)z_2(s, t) = K_z(s, t); \\ y_1(s, t) + y_2(s, t) &= L_y(s, t); \quad z_1(s, t) + z_2(s, t) = L_z(s, t). \end{aligned} \quad (3.82)$$

Jų išraiškos yra:

$$\begin{aligned} K_y &:= K_y(s, t) = \left(1 + \frac{1}{\rho_1}\right) \left(1 + \frac{1}{\rho_2}\right) + \frac{\lambda_1}{\sigma\rho_1} \left(1 + \frac{1}{\rho_2}\right) + \frac{\lambda_2}{\sigma\rho_2} \left(1 + \frac{1}{\rho_1}\right); \\ K_z &:= K_z(s, t) = \left(1 + \frac{1}{\rho_1}\right) \left(1 + \frac{1}{\rho_2}\right) + \frac{\mu_1}{\gamma\rho_1} \left(1 + \frac{1}{\rho_2}\right) + \frac{\mu_2}{\gamma\rho_2} \left(1 + \frac{1}{\rho_1}\right); \\ L_y &:= L_y(s, t) = 2 + \frac{1}{\rho_1} + \frac{1}{\rho_2} + \frac{\lambda_1}{\sigma\rho_1} + \frac{\lambda_2}{\sigma\rho_2}; \\ L_z &:= L_z(s, t) = 2 + \frac{1}{\rho_1} + \frac{1}{\rho_2} + \frac{\mu_1}{\gamma\rho_1} + \frac{\mu_2}{\gamma\rho_2}. \end{aligned} \quad (3.83)$$

Pastebėsime, kad pradinės sąlygos s, t tenkina lygybes:

$$\sigma \frac{Y_*(1, s, t)}{X_*(1, s, t)} = s; \quad \gamma \frac{Z_*(1, s, t)}{X_*(1, s, t)} = t. \quad (3.84)$$

Simboliniai pertvarkymai leidžia surasti sistemos (3.42) invariantus. Pažymėkime juos $\text{inv}_y, \text{inv}_z, \text{Inv}_y, \text{Inv}_z \in \mathbb{R}$. Šie invariantai išlieka pastovūs su bet kokiais s, t reikšmėmis (Programa P224):

$$\begin{aligned} \text{Inv}_y &:= \frac{y_1(s, t)y_2(s, t)}{x_1(s, t)x_2(s, t)}, \quad \text{Inv}_z := \frac{z_1(s, t)z_2(s, t)}{x_1(s, t)x_2(s, t)}; \\ \text{inv}_y &:= \frac{\lambda_1(s, t)}{1 + \rho_1(s, t)} + \frac{\lambda_2(s, t)}{1 + \rho_2(s, t)}, \quad \text{inv}_z := \frac{\mu_1(s, t)}{1 + \rho_1(s, t)} + \frac{\mu_2(s, t)}{1 + \rho_2(s, t)}. \end{aligned} \quad (3.85)$$

Invariantai taip pat tenkina šias lygybes:

$$\text{Inv}_y = 1 + \frac{1}{\sigma} \text{inv}_y; \quad \text{Inv}_z = 1 + \frac{1}{\gamma} \text{inv}_z. \quad (3.86)$$

Darkart pastebėsime, kad operatorinis metodas ne tik leidžia apskaičiuoti analizinę bendrojo

sprendinio išraišką, bet taip pat parodo šio sprendinio priklausomybę nuo pradinių sąlygų s, t [6, 10]. Taigi, kaip darėme ir atvirkštinio balansavimo metode, formaliai galima operatorinį metodą užrašyti atitiktimi:

$$(a_0, a_1, a_2, b_0, b_1, b_2) \rightarrow (\eta, \sigma, y_1(s, t), y_2(s, t), \gamma, z_1(s, t), z_2(s, t)). \quad (3.87)$$

Pasinaudoję šio poskyrio rezultatais galime užrašyti kokius parametrus gautume, pasirinkę ne teigiamą, o neigiamą η šaknį:

$$(a_0, a_1, a_2, b_0, b_1, b_2) \rightarrow \left(-\eta, \sigma \text{Inv}_y, \frac{1}{y_1(s, t)}, \frac{1}{y_2(s, t)}, \gamma \text{Inv}_z, \frac{1}{z_1(s, t)}, \frac{1}{z_2(s, t)} \right). \quad (3.88)$$

Taigi, jeigu (3.79) yra bendrasis (3.77) sprendinys, tai funkcijų pora

$$y^+(x, c, s, t) = \sigma \frac{Y_*(-e^{\eta_0(x-c)}, s, t)}{X_*(-e^{\eta_0(x-c)}, s, t)}; \quad z^+(x, c, s, t) = \gamma \frac{Z_*(-e^{\eta_0(x-c)}, s, t)}{X_*(-e^{\eta_0(x-c)}, s, t)}, \quad (3.89)$$

taip yra bendrasis (3.77) sprendinys, t.y. su bet kokiais $s_0, t_0 \in \mathbb{R}$ egzistuoja tokios pradinės sąlygos $s_1, t_1 \in \mathbb{R}$, kad $y_k(s_0, t_0) = -y_k(s_1, t_1); z_k(s_0, t_0) = -z_k(s_1, t_1), k = 1, 2$. Taip pat galime pastebėti, kad konkrečioms parametru s_0, t_0 reikšmėms, funkcijos $y^+(x, c, s_0, t_0), z^+(x, c, s_0, t_0)$ duoda „likusių“ antros eilės kreivės, kurios fragmentu yra $y(x, c, s_0, t_0), z(x, c, s_0, t_0)$, dalį.

3.3. Solitoninių sprendinių dinamika

Šiame poskyryje panagrinėsime dinamines Rikati lygčių, susietų multiplikatyviniais sąryšiais, sistemos savybes, kai egzistuoja jų solitoniniai sprendiniai.

3.3.1. Pusiausvyros taškai ir stabilumas

Sistemos (3.77) pusiausvyros taškai $(y_*^{(k)}, z_*^{(k)})$ gaunami išsprendus algebrinių lygčių sistemą:

$$\begin{aligned} a_0 + a_1 y + a_2 y^2 + b_2 y z &= 0; \\ b_0 + b_1 y + b_2 y^2 + a_2 y z &= 0. \end{aligned} \quad (3.90)$$

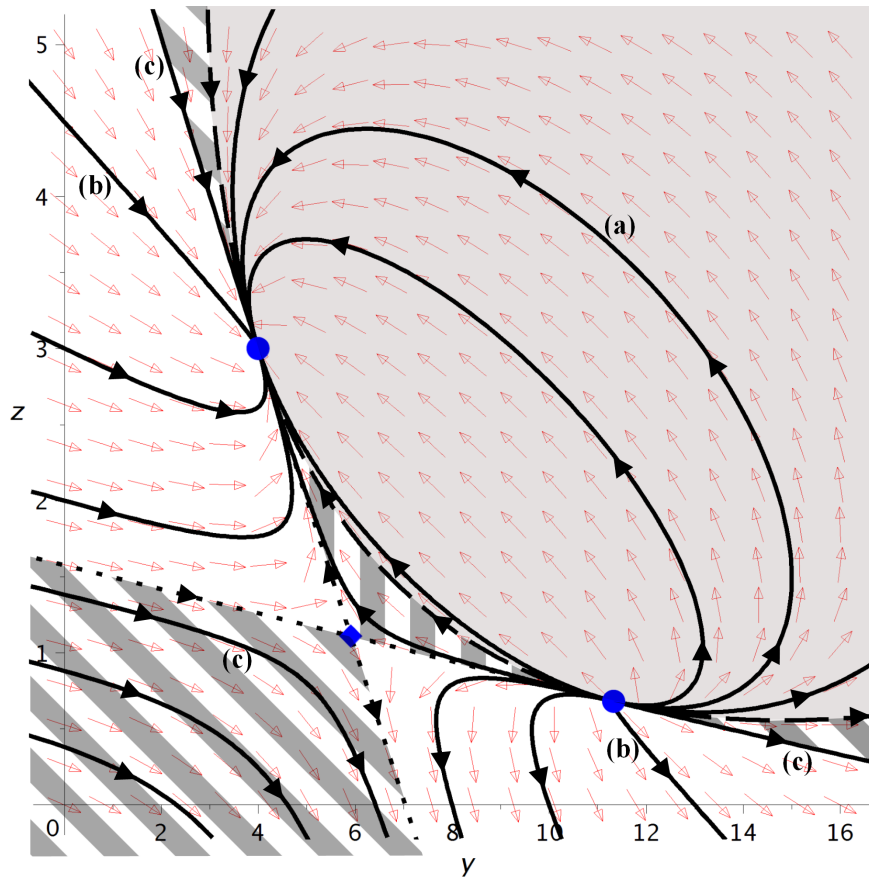
Sistemos (3.90) yra:

$$y_*^{(k)} = -\frac{(z_*^{(k)})^2 b_2 + z_*^{(k)} b_1 + b_0}{z_*^{(k)} a_2}; \quad z_*^{(k)} = z_*^{(k)}, \quad (3.91)$$

kur $z_*^{(k)}$ tenkina:

$$(b_1 b_2 - a_1 a_2) z^3 + (b_1^2 + b_0 b_2 + a_0 a_2 - a_1 b_1) z^2 + (2b_0 b_1 - a_1 b_0) z + b_0^2 = 0. \quad (3.92)$$

Iš (3.92) matome, kad sistema (3.77) turi daugiausiai tris pusiausvyros taškus. Tarkime, kad



3.1 Pav.: 3.3.2 poskyrio (3.94) sistemos fazinis portretas. Skrituliai žymi mazgus $(y_*^{(1)}, z_*^{(1)}) = (4, 3)$ ir $(y_*^{(2)}, z_*^{(2)}) = \left(\frac{34}{3}, \frac{17}{25}\right)$; rombas žymi balno tašką $(y_*^{(3)}, z_*^{(3)}) = \left(\frac{136}{23}, \frac{51}{46}\right)$; juodos linijos žymi sprendinio trajektorijas. Žymės (a), (b), (c) atitinka sprendinius iš 3.2 pav. Retais punktyrais pažymėta linija žymi elipsines ir hiperbolines trajektorijas skiriančią separatrisę. Tankiais punktyrais pažymėta linija žymi stabilias ir nestabilias balno taško daugdaras. Sprendiniai pilkai užtušiuotame regione neturi trūkio taškų, o jų trajektorijos yra elipsės fragmentai. Vertikaliai dryžuotame regione sprendiniai neturi trūkio taškų, o jų trajektorijos yra hiperbolės fragmentai. Neužtušiuotame regione esantys sprendiniai turi vieną trūkio tašką, o jų trajektorijos - hiperbolės fragmentai. Įstrižai užtušiuotame regione sprendiniai turi du trūkio taškus, jų trajektorijos - hiperbolės fragmentai.

taip ir yra, t.y. (3.92) tris skirtingas realias šaknis. Dviejų pusiausvyros taškų koordinatės galima rasti pasinaudojus sprendinio išraiška (3.79):

$$\begin{aligned} \lim_{x \rightarrow +\infty} (y(x, c, s, t), z(x, c, s, t)) &= (\sigma, \gamma) =: (y_*^{(1)}, z_*^{(1)}); \\ \lim_{x \rightarrow -\infty} (y(x, c, s, t), z(x, c, s, t)) &= (\sigma \text{Inv}_y, \gamma \text{Inv}_z) =: (y_*^{(2)}, z_*^{(2)}); \end{aligned} \quad (3.93)$$

visoms c, s, t reikšmės. Taigi $(y_*^{(1)}, z_*^{(1)}) = (\sigma, \gamma)$ yra stabilus mazgas; $(y_*^{(2)}, z_*^{(2)}) = (\sigma \text{Inv}_y, \gamma \text{Inv}_z)$ - nestabilus mazgas. Trečiasis pusiausvyros taškas $(y_*^{(3)}, z_*^{(3)})$ yra balno taškas.

Atskiru atveju sistemoje gali likti tik du ar net vienas pusiausvyros taškai. Šio skyriaus poskyriuose panagrinėsime šiuos atvejus.

Balno taško bei mazgų koordinatės, taip pat ir separatrisės atskiriančios elipsines ir hiperbolines fazines trajektorijas gali būti rastos analiziškai pasinaudojant simboliniais pertvarkymais. Šio

poskyrio skaičiavimai realizuoti programoje P231, jos tekstą galima rasti C.8 priede.

3.3.2. Trys pusiausvyros taškai

Ištirsime sistemą

$$\begin{aligned} y'_x &= \frac{136}{11} - \frac{828}{319}y + \frac{29}{187}y^2 - \frac{550}{1479}yz; \\ z'_x &= -\frac{51}{29} + \frac{345}{319}z - \frac{550}{1479}z^2 + \frac{29}{187}yz. \end{aligned} \quad (3.94)$$

Sistema (3.94) turi tris pusiausvyros taškus: $(y_*^{(1)}, z_*^{(1)}) = (4, 3)$ (stabilus mazgas); $(y_*^{(2)}, z_*^{(2)}) = \left(\frac{34}{3}, \frac{17}{25}\right)$ (nestabilus mazgas) ir $(y_*^{(3)}, z_*^{(3)}) = \left(\frac{136}{23}, \frac{51}{46}\right)$ (balno taškas).

Iš 3.1 ir 3.2 pav. matome, kad sistema turi tris rūšis sprendinių su atitinkamai nulių, vienu ar dviem trūkio taškais (poliais). Prie darbo pridėta vaizdinė medžiaga (sec_2_3_2_v1.mp4, sec_2_3_2_v2.mp4, sec_2_3_2_v3.mp4) parodo skirtumus šių sprendinių dinamikoje.

Pasinaudoję programa P232 apskaičiuojame charakteristines tiesiškai rekurentinės sekos šaknis $\rho_{1,2}$ tiriamai sistemai:

$$\begin{aligned} \rho_{1,2} &= -\frac{240}{319} + \frac{29s}{374} - \frac{275t}{1479} \\ &\pm \frac{\sqrt{6365529s^2 - 30528300st + 36602500t^2 - 171649782s - 312252600t + 1558425564}}{32538}. \end{aligned} \quad (3.95)$$

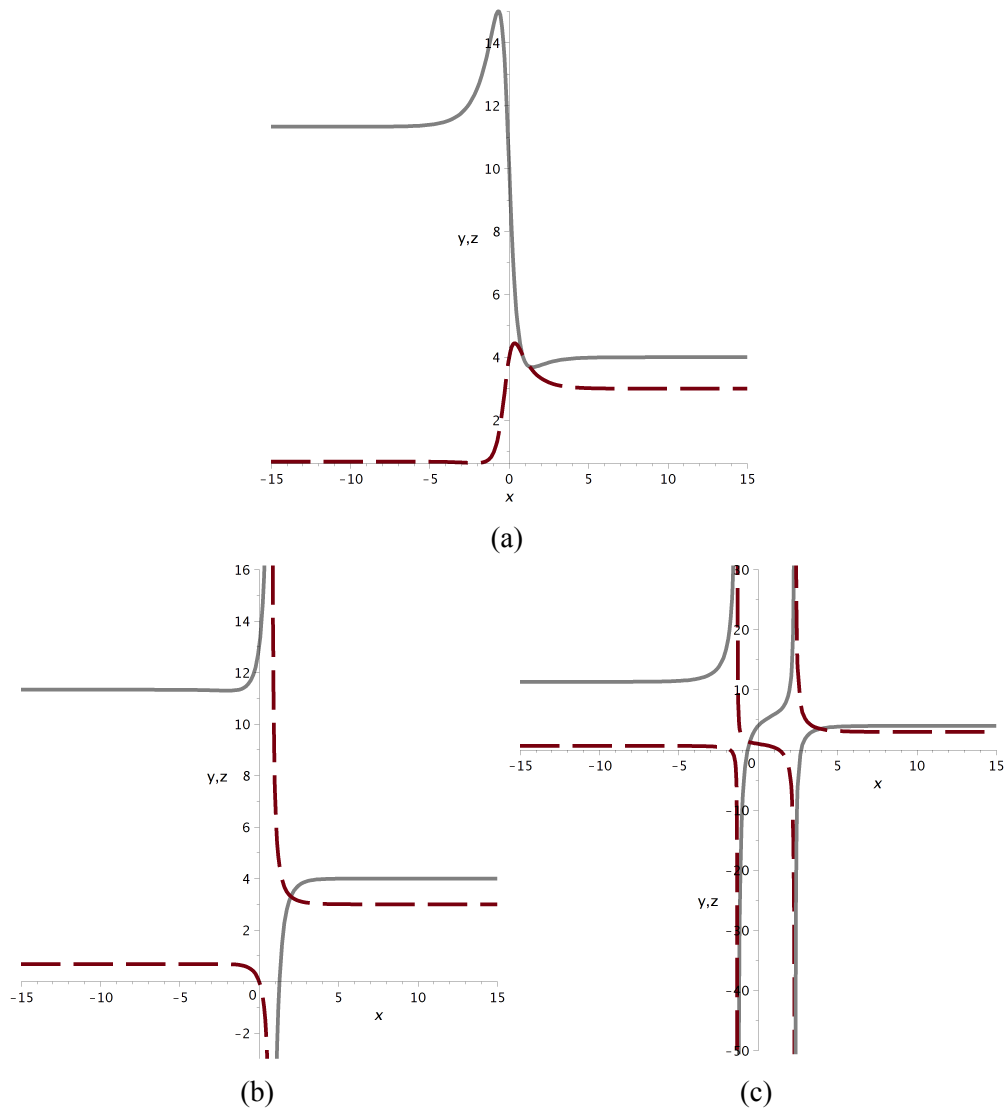
Pastebėsime, kad $x_k = 1 + \frac{1}{\rho_k}$, $k = 1, 2$, taigi x_k yra kompleksinis, jeigu ρ_k kompleksinis. Galima pastebėti, kad fazinės trajektorijos yra elipsinės, kai x_1 ir x_2 yra kompleksiniai; hiperbolės, kai x_1 ir x_2 realūs. Todėl separatrixės lygtis yra $\rho_{1,2}$ pošaknis:

$$6365529s^2 - 30528300st + 36602500t^2 - 171649782s - 312252600t + 1558425564 = 0. \quad (3.96)$$

Taip pat galima pastebėti, kad fazinės trajektorijos išsigimsta į tieses, kai $x_k = \frac{\rho_k + 1}{\rho_k}$ yra nulinis arba neapibrėžtas. Prilyginę nuliui skaitiklį ir vardiklį gauname lygtis:

$$\begin{aligned} \rho_k + 1 = 0 &\Rightarrow 88t + 87s - 612 = 0; \\ \rho_k = 0 &\Rightarrow 1100t + 87s - 1734 = 0. \end{aligned} \quad (3.97)$$

Pasinaudojant kompiuteriniu modeliavimu, galima pastebėti, kad balno taško daugdaros yra ant tiesių (3.97), be to, balno taškas yra šių tiesių susikirtime (3.1 pav.).



3.2 Pav.: Sistemos (3.94) sprendinių evoliucija kintamojo x atžvilgiu. Pilka linija žymi $y(x)$; juoda taškuota linija žymi $z(x)$. Pastebėsime, kad (b) atveju, fazinės trajektorijos „susitinka“ tik begalybėje, o (c) atveju tas pats efektas stebimas du kartus.

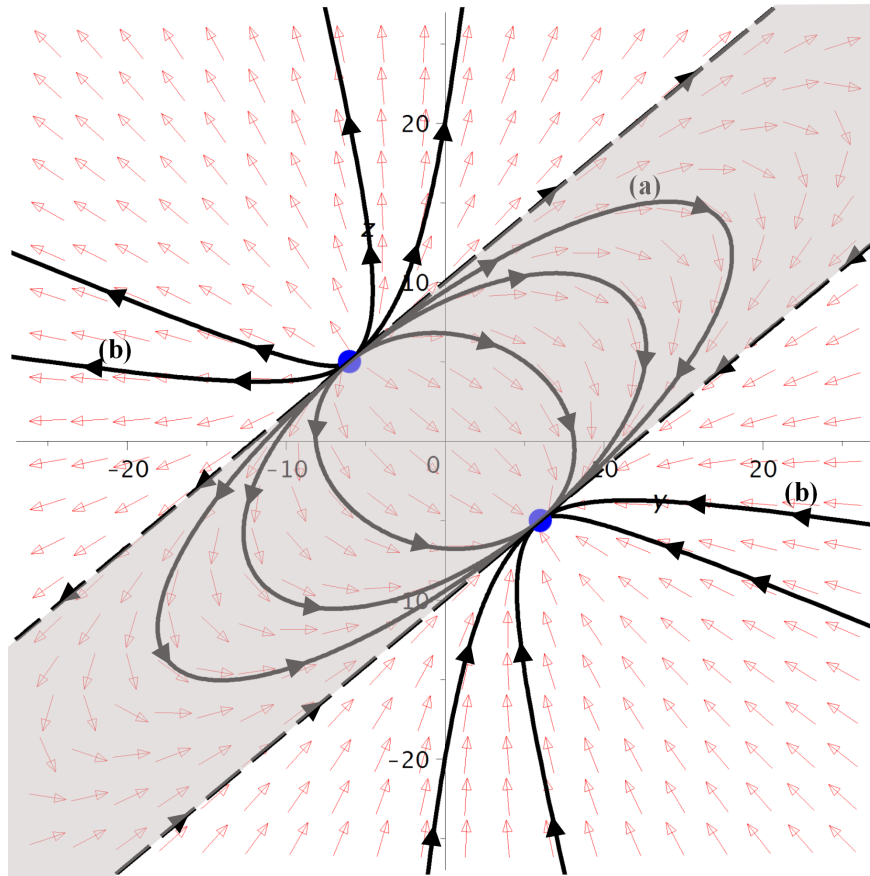
3.3.3. Du pusiausvyros taškai

Nagrinsime sistemą

$$\begin{aligned} y'_x &= 6 - \frac{1}{12}y^2 + \frac{1}{10}yz; \\ z'_x &= -5 + \frac{1}{10}z^2 - \frac{1}{12}yz. \end{aligned} \quad (3.98)$$

Šioje sistemoje yra du pusiausvyros taškai $(y_*^{(1)}, z_*^{(1)}) = (6, -5)$ (stabilus mazgas) ir $(y_*^{(2)}, z_*^{(2)}) = (-6, 5)$ (nestabilus mazgas).

Fazinis sistemos (3.98) portretas pavaizduotas 3.3 pav. Visus sistemos (3.98) sprendinius galima suskirstyti į dvi grupes: sprendinius, kurie lieka aprėžti, kai $x \rightarrow \pm\infty$ (3.4 (a) pav.) ir sprendinius, turinčius vieną trūkio tašką (3.4 (b) pav.). Šiam poskyriui taip pat sukurta vaizdinė medžiaga (sec_2_3_3_v1.mp4, sec_2_3_3_v2.mp4).



3.3 Pav.: 3.3.3 poskyrio (3.98) sistemos fazinis portretas. Skrituliai žymi mazgus $(y_*^{(1)}, z_*^{(1)}) = (6, -5)$ ir $(y_*^{(2)}, z_*^{(2)}) = (-6, 5)$; juodos linijos žymi sprendinio trajektorijas. Žymės (a), (b) atitinka sprendinius iš 3.4 pav. Retais punktyrais pažymėta linija žymi elipsines ir hiperbolines trajektorijas skiriančią separatrį. Tankiais punktyrais pažymėta linija žymi stabilias ir nestabilias balno taško daugdaras. Sprendiniai pilkai užtušiuotame regione neturi trūkio taškų, o jų trajektorijos yra elipsės fragmentai. Neužtušiuotame regione esantys sprendiniai turi vieną trūkio tašką, o jų trajektorijos - hiperbolės fragmentai.

Atliksime tuos pačius skaičiavimus, kaip ir praeitame poskyryje. Apskaičiuojame charakteristines tiesiškai rekurentinių sekų šaknis:

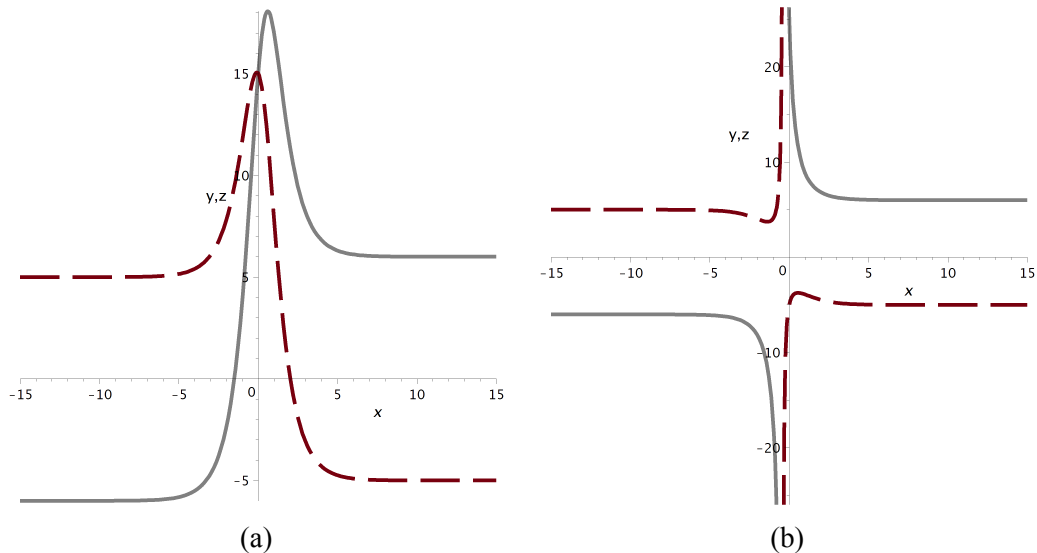
$$\rho_{1,2} = -\frac{1}{2} - \frac{s}{24} + \frac{t}{20} \pm \frac{\sqrt{(60 + 5s - 6t)(5s - 60 - 6t)}}{120}. \quad (3.99)$$

Šiuo atveju separatrį yra dvi tiesės:

$$5s - 6t + 60 = 0; \quad 5s - 6t - 60 = 0. \quad (3.100)$$

Nors sistema neturi balno taško, galime suskaičiuoti atvejus, kai x_k yra nulinis arba neegzistuoja. Gauname:

$$\begin{aligned} \rho_k + 1 = 0 &\Rightarrow 5s - 6t + 60 = 0; \\ \rho_k = 0 &\Rightarrow 5s - 6t - 60 = 0. \end{aligned} \quad (3.101)$$



3.4 Pav.: Sistemos (3.98) sprendinių evoliucija kintamojo x atžvilgiu. Pilka linija žymi $y(x)$; juoda taškuota linija žymi $z(x)$. Pastebėsime, kad (b) atveju, fazinės trajektorijos „susitinka“ tik begalybėje.

Šios lygybės sutampa su (3.100), taigi tiriamos sistemos atveju balno taškas yra be galo nutolęs taškas.

4. PROGRAMINĖ REALIZACIJA

Darbo rezultatų išvedimui naudojama kompiuterinės algebros sistema Maple. Tai - daugiafunkcinė matematinė programinė įranga, orientuota į simbolinius pertvarkymus, tačiau turinti ir skaitinių metodų paketų.

Dauguma iš darbe naudojamų programų yra „vienkartinės“ - skirtos konkrečiam analiziniam rezultatui išvesti ir pademonstruoti. Išimtis - programos P232 ir P233, kadangi jas redaguojant galima sukonstruoti kitus pavyzdžius.

Toliau šiame skyriuje pateikiame trumpą svarbiausių darbe naudotų Maple funkcijų bei sukurtų darbalapių²⁴ aprašymus.

4.1. Darbe naudojamų funkcijų aprašymai

1. `simplify(expr, par)` - suprastina duotą matematinę išraišką `expr`. Galimos įvairios suprastinimo parinktys: jeigu nurodomas parametras `symbolic` prastinami ir reiškiniai, kurie gali būti lygūs nuliui; parametras `siderels` leidžia suprastinti išraišką atsižvelgiant į kitus duotus sąryšius tarp prastinamos išraiškos kintamųjų; parametras `size` nurodo, kad išraiška turi būti suprastinta taip, kad užimtų kuo mažiau vietos.
2. `solve(eqns, vars)` - išsprendžia lygtį (arba lygčių sistemą) `eqns` kintamojo (arba kintamųjų) `vars` atžvilgiu.

²⁴angl. *worksheet*

3. `eval(expr, x = x0)` - apskaičiuoja matematinio reiškinių `expr` reikšmę, kai kintamasis x prilyginamas reikšmei (skaitinei ar kitam kintamajam) x_0 .
4. `subs(expr, [x = y])` - išraiškoje `expr` pakeičia kintamąjį x kintamuoju y .
5. `algsubs(expr, [x = y])` - išraiškoje `expr` pakeičia kintamąjį x kintamuoju y taip, kad kintamojo x algebrinių funkcijų išraiškoje neliktų: įvykdo kintamojo pakeitimą su pertvarkymu.
6. `lhs(expr)`, `rhs(expr)` - grąžina atitinkamai kairę ir dešinę matematinės išraiškos `expr` su vienu `=` ženklu pusę.
7. `numer(expr)`, `denom(expr)` - grąžina atitinkamai skaitiklį ir vardiklį matematinės išraiškos `expr`.
8. `factor(expr)` - išskaido išraišką `expr` dauginamaisiais.
9. `Determinant(A)`, `Rank(A)`, `Eigenvalues(A)` - apskaičiuoja matricos A determinantą, rangą ir tikrines reikšmes. Skaičiavimai vykdomi analiziškai, nenaudojant skaitinių metodų.
10. `coeff(expr, x)` - grąžina kintamojo x (bet ne x^k , $k \neq 1$) koeficientą išraiškoje `expr`.
11. `collect(expr, x)` - surenka koeficientus prie x laipsnių išraiškoje `expr`.
12. `indets(expr)` - grąžina nežinomųjų sąrašą išraiškoje `expr`.
13. `plot(f(x), x = x1..x2)` - nubraižo funkcijos $f(x)$ grafiką, kai $x \in [x_1, x_2]$.
14. `implicitplot(f(x,y) = 0, x = x1..x2, y = y1..y2)` - nubraižo neišreikštinėje formoje duotą plokštumos kreivę $f(x,y)$, kai $x \in [x_1, x_2]$, $y \in [y_1, y_2]$.
15. `DEplot(diffSist, opts)` - nubraižo antros eilės diferencialinių lygčių sistemos `diffSist` fazinį portretą.

4.2. Darbui parašytų darbalapių (programų) aprašymai

1. P11_1, P11_2 - 1.1, 1.2, 1.3, 1.4 pav. braižymas.
2. P137 - operatoriniu metodu išsprendžia nesusietą Rikati lygtį. Įvykdomi poskyrio 2.1.7 skaičiavimai.
3. P211 - pritaiko atvirkštinio balansavimo metodą Rikati lygčių sistemai susietai multiplikatyviuoju sąryšiu: gaunamos parametru $a_k, b_k, k = 1, \dots, 3$ išraiškos per solitoninių sprendinių parametrus. Įvykdomi poskyrio 3.1.1.
4. P213 - įrodymas, kad tiriami antros eilės solitonai fazinėje plokštumoje y, z sudaro antros eilės kreivių fragmentus. Įvykdomi poskyrio 3.1.3 skaičiavimai.
5. P222 - pakankamų antros eilės solitoninių sprendinių egzistavimo sąlygų radimas naudojantis operatoriniu metodu. Įvykdomi poskyrio 3.2.2 skaičiavimai.

6. P223 - suformuojama tiesiškai rekurentinė seka, reikalinga rasti antros eilės solitono išraiškai. Randamos šios tiesiškai rekurentinės sekos charakteristinės reikšmės ir jų koeficientai. Patikrinamos sąlygos, keliamos charakteristinėms reikšmėms ir jų koeficientams. Įvykdomi poskyrio 3.2.3 atskiram atvejui (bendru atveju skaičiavimai reikalauja itin daug kompiuterio resursų).
7. P231 - randamos bendros pusiausvyros taškų išraiškos. Įvykdomi poskyrio 3.3.1 skaičiavimai.
8. P232 - išnagrinėjamas poskyryje 3.3.2 pateiktas pavyzdys: surandami tiesiškai rekurentinės sekos parametrai; randami pusiausvyros taškai; analiziškai apskaičiuojama separatrixė tarp elipsinę ir hiperbolinę fazinę trajektoriją turinčių sprendinių; nubraižomas fazinis portretas ir sprendinių evoliucija laiko ašyje.
9. P233 - atliekami analogiški veiksmai kaip ir P232 poskyrio 3.3.3 sistemai.

5. REZULTATŲ APTARIMAS

5.1. Tyrimo apžvalga

Darbe įrodyta, kad Rikati lygčių susietų multiplikatyviuoju sąryšiu sistema prie tam tikrų sąlygų turi antros eilės solitoninius sprendinius. Šiam rezultatui gauti panaudoti du matematiniai įrankiai: atvirkštinio balansavimo metodas, sugeneruojantis būtinas solitoninių sprendinių egzistavimo sąlygas bei apibendrintojo diferencialinio operatoriaus (operatorinio skaičiavimo) metodas, ne tik sugeneruojantis pakankamas solitoninių sprendinių egzistavimo sąlygas, bet ir leidžiantis užrašyti bendrąją sistemos sprendinį.

Reikėtų pabrėžti, kad darbe naudoti metodai kokybiškai skiriasi nuo plačiai paplitusių standartinių funkcijų metodų, nes šiuo atveju sprendinio forma ne atspėjama, o randama analiziškai, taigi nelieka abejonių dėl sprendinio parametrų išraiškų korektiškumo. Visi pateikti skaičiavimai realizuoti kompiuterinės algebros sistema Maple.

5.2. Analizinės sprendinio formos pranašumai

Kuomet diferencialines lygtis bei jų sistemas galima dideliu tikslumu išspręsti skaitiniais metodais, neretai kyla klausimas dėl analizinio sprendinio paieškos tikslingumo. Vis dėlto privalome pažymėti, kad tikslaus diferencialinių lygčių sprendinio radimas atveria naujas tyrimo galimybes ir neturi paprastų trūkumų, kuriais pasižymi net ir pati tiksliausia sprendinio aproksimacija skaitiniais metodais.

Vienas iš lengviausiai matomų analizinio sprendinių privalumų yra galimybė apskaičiuoti visą sprendinio evoliuciją laike, nepaisant to, kad sprendinys turi vieną ar kelis ypatingus taškus. Pavyzdžiui, 3.2 (c) pav., parinkus pradinių sąlygų s, t reikšmes taip, kad $c \in (-1, 2)$ net ir tiksliausias skaitinis integratorius negalės apskaičiuoti $y(x), z(x)$ reikšmių kai $x < -1$ ir $x > 2$, kadangi

neįmanoma peržengti ypatingų sprendinio taškų. Žinoma, egzistuoja tokios pradinės sąlygos s', t' , kurios atitinka to paties sprendinio tęsinį už ypatingo taško, tačiau neturint analizinio sprendinio jų nustatymas negalimas.

Kitas šiame darbe taikomų metodų privalumas yra galimybė tiksliai išnagrinėti sprendinių elgesį fazinėje plokštumoje: įrodoma, kad gautieji sprendiniai fazinėje plokštumoje gali būti tik antros eilės kreivės. Analiziškai randama separatrixė, atskirianti sprendinius turinčius ir neturinčius ypatingų taškų bei balno taško stabilios ir nestabilios daugdaros. Tai leidžia suskirstyti visus sistemos sprendinius į klases pagal jų kitimo pobūdį.

Nors darbo tikslas ir nėra interpretuoti gautus sprendinius biologiškai, kadangi panašios sistemos šioje srityje taikomos, o $y(x), z(x)$ gali būti interpretuojamos kaip skirtingų rūšių populiacijos, galima pasamprotauti, kad elipsines fazines trajektorijas turinčių sprendinių klasė yra vystymosi atvejis, kuomet nė viena rūšis neišnaikina kitos konkurencinėje kovoje, o jų vystymasis visuomet kinta tolydžiai ir nusistovi. Tuo tarpu kuomet sprendiniai turi singuliarinius taškus (hiperbolinių fazinių trajektorijų atvejis) įvyksta sistemos katastrofa ir viena ar abi rūšys išnyksta.

IŠVADOS

1. Būtinoms antros eilės solitoninio sprendinio **(0.5)** egzistavimo Rikati lygčių, susietų multiplikatyviuoju sąryšiu sistemoje **(0.4)** sąlygos:

$$a_3 = b_2, \quad b_3 = a_2; \quad (5.1)$$

$$\frac{X(y_1)}{X(y_2)} = -\frac{y_1}{y_2}, \quad \frac{X(z_1)}{X(z_2)} = -\frac{z_1}{z_2}. \quad (5.2)$$

2. Pakankama antros eilės solitoninio sprendinio **(0.5)** egzistavimo Rikati lygčių, susietų multiplikatyviuoju sąryšiu sistemoje **(0.4)** sąlyga:

$$9a_0a_1a_2 + 9b_0b_1b_2 - 18a_0a_2b_1 - 18b_0b_2a_1 + 3a_1b_1^2 + 3b_1a_1^2 - 2a_1^3 - 2b_1^3 = 0. \quad (5.3)$$

3. Apibendrintojo diferencialinio operatoriaus metodu randamas bendrasis **(0.4)** sprendinys bei jo priklausomybė nuo pradinių sąlygų.
4. Antros eilės solitoniniai sprendiniai fazinėje plokštumoje sudaro antros eilės kreives, taigi visus sprendinius galima sugrupuoti į elipsinius, parabolinius ir hiperbolinius. Elipsiniai sprendiniai neturi, o hiperboliniai turi vieną arba du ypatingus taškus.

REKOMENDACIJOS

Tolesnės tyrimo kryptys galėtų būti susiję su tolesniu Rikati sistemos **(0.4)** analiziniu tyrimu: likę neišnagrinėti atvejai, kuomet sistemoje neegzistuoja tokios formos sprendiniai, kokių buvo ieškoma šiame darbe. Tai nereiškia, kad **(0.4)** negali turėti kitų sprendinių, taip pat išreiškiamų

eksponenčių sumų santykiais. Tokių sprendinių formos galėtų būti pirmos eilės solitonai ((0.2), tikėtina kad jie egzistuoja tais atvejais, kai antros eilės solitonų išraiškose vardikliai virsta nuliu) arba kitokios formos antros eilės solitonai, pavyzdžiui:

$$y(x) = \sigma \frac{(e^{\eta(x-c)} - y_1)(e^{\eta(x-c)} - y_0)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - z_0)}; z(x) = \gamma \frac{(e^{\eta(x-c)} - z_1)(e^{\eta(x-c)} - z_0)}{(e^{\eta(x-c)} - x_1)(e^{\eta(x-c)} - y_0)}, \quad (5.4)$$

kur parametrai $x_1, y_0, y_1, z_0, z_1, \sigma, \gamma, \eta \in \mathbb{R}$. Verta pažymėti, kad pirmos eilės solitonas ir (5.4) egzistuos ne prie visų pradinių sąlygų, o tik tam tikroje s, t plokštumos aibėje.

Kita tyrimo kryptis - operatorinio bei atvirkštinio balansavimo metodų pritaikymai solitoninių sprendinių paieškai paprastosiame diferencialinėse lygtyse. Nors šioje srityje jau nemažai kas nuveikta [10], [45], dar lieka vietos naujoms išvalgoms, pavyzdžiui bendresniam algebrinės diferencialinės lygties

$$y_x^{(m)} + b_{m-1}y_x^{(m-1)} + \dots + b_1y_x' = a_ny^n + a_{n-1}y^{n-1} + \dots + a_0, \quad (5.5)$$

čia $b_j, a_k \in \mathbb{R}; j = 1, \dots, m-1, k = 0, \dots, n$, tyrimui. Šios lygties atskiri atvejai sulaukia didelio mokslininkų dėmesio.

Šiuose tyrimuose vis didesnis vaidmuo teks kompiuterinės algebras sistemoms, galinčioms susitvarkyti su itin didelės apimties simboliniais skaičiavimais. Tikėtina, kad didėjant šių sistemų spartai bus galima taikyti analizinių sprendinių paieškos metodus ir sudėtingesnėms (galbūt ir neautonominėms) diferencialinėms lygtims.

PADĖKA

Dėkoju darbo vadovui prof. Minvydui Ragulskiui, prof. Zenonui Navickui ir doc. Romui Marcinkevičiui už pasiūlytą darbo temą ir pagalbą bei vertingus patarimus jį rengiant.

LITERATŪRA

- [1] Allee, W. C. *Animal Aggregations. A study in General Sociology.* Chicago: University of Chicago Press, 1931.
- [2] McCallum H., Barlow N., Hone J. How should pathogen transmission be modelled? *Trends in Ecology and Evolution*, 2001, vol. 16, p. 295-300.
- [3] Courchamp F.; Berec L.; Gascoigne J. *Allee effects in ecology and conservation.* Oxford: Oxford University Press, 2008.
- [4] Kraenkel R., Manikandan K., Senthilvela M. On certain new exact solutions of a diffusive predator-prey system. *Communications in Nonlinear Science and Numerical Simulation*, 2013, vol. 18, p. 1269-1274.
- [5] Owen M. R., Lewis M. A. How predation can slow, stop or reverse a prey invasion. *Bulletin of Mathematical Biology*, 2001, vol. 63, p. 655-684.
- [6] Navickas Z., Ragulskis M., Bikulciene L. Be careful with the exp-function method - additional remarks. *Communications in Nonlinear Science and Numerical Simulation*, 2010, vol. 15, p. 3874-3886.
- [7] Navickas Z., Bikulciene L. Expressions of solutions of ordinary differential equations by standard functions. *Mathematical Modeling and Analysis*, 2006, vol. 11, p. 399-412.
- [8] Navickas Z., Ragulskis M. Comments on "Two exact solutions to the general relativistic Binet's equation". *Astrophysics and Space Science*, 2013, vol. 344, p. 281-285.
- [9] Navickas Z., Ragulskis M., Bikulciene L. Special solutions of Huxley differential equation. *Mathematical Modeling and Analysis*, 2011, vol. 16, p. 248-259.
- [10] Navickas Z., Bikulciene L., Ragulskis M. Generalization of Exp-function and other standard function methods. *Applied Mathematics and Computation*, 2010, vol. 216, p. 2380-2393.
- [11] *Encyclopedia of Nonlinear Science.* Ed. Scott, A. New York: Routledge, 2004.
- [12] Weisstein E. W. Korteweg-de vries equation. In: Mathworld - A Wolfram Web Resource [online]. Wolfram, 2015 [viewed 2015-03-03]. Available from: <http://mathworld.wolfram.com/Korteweg-deVriesEquation.html>.
- [13] Zabusky N. J.; Porter M. A. Soliton. In: Scholarpedia [online]. Scholarpedia, 2015 [viewed 2015-03-03]. Available from: <http://www.scholarpedia.org/article/Soliton>.
- [14] *Handbook of Mathematics for Engineers and Scientists.* Ed. Polyanin, A. D.; Manzhirov, A. V. New York: Chapman and Hall/CRC Press, 2007.

- [15] Weisstein E. W. Sine-Gordon equation. In: Mathworld - A Wolfram Web Resource [online]. Wolfram, 2015 [viewed 2015-03-03]. Available from: <http://mathworld.wolfram.com/Sine-GordonEquation.html>.
- [16] Griffiths G. W., Schiesser W. E. Linear and nonlinear waves. In: Scholarpedia [online]. Scholarpedia, 2015 [viewed 2015-03-03]. Available from: http://www.scholarpedia.org/article/Linear_and_nonlinear_waves.
- [17] Aslan I., Marinakis V. Some remarks on Exp-function method and its applications. *Communications in Theoretical Physics*, 2011 vol. 56, p. 397-403.
- [18] He J. H., Wu X. H. Exp function method for nonlinear wave equations. *Chaos Solitons Fractals*, 2006, vol. 30, p. 700-708.
- [19] Chun C. New solitary wave solutions to nonlinear evolution equations by the Exp-function method. *Computers and Mathematics With Application*, 2010, vol. 61, p. 2107-2110.
- [20] Wazwaz A. New solitary wave solutions to nonlinear evolution equations by the Exp-function method. *Applied Mathematics Letters*, 2014, vol. 38, p. 174-179.
- [21] Navickas Z., Ragulskis M. How far can one go with the Exp-function method? *Applied Mathematics and Computation*, 2009, vol. 211, p. 522-530.
- [22] Kudryashov N. A., Loguinova N. B. Be careful with Exp-function method. *Communications in Nonlinear Science and Numerical Simulation*, 2009, vol. 14, p. 1891-1900.
- [23] Kudryashov N. A. Seven common errors in finding exact solutions of nonlinear differential equations. *Communications in Nonlinear Science and Numerical Simulation*, 2009, vol. 14, p. 3507-3529.
- [24] Popovych R. O. More common errors in finding exact solutions of nonlinear differential equations: Part I. *Communications in Nonlinear Science and Numerical Simulation*, 2010, vol. 15, p. 3887-3899.
- [25] Navickas Z., Telksnys T., Ragulskis M. Comments on "The exp-function method and generalized solitary solutions". *Computers and Mathematics with Applications*, 2015, vol. 69, p. 798-803.
- [26] Fan E. G. Extended tanh-function method and its application to nonlinear equations. *Physics Letters A*, 2000, vol. 277, p. 212-218.
- [27] Wazwaz A. The extended tanh method for abundant solitary wave solutions of nonlinear wave equations. *Applied Mathematics and Computation*, 2007, vol. 187, p. 1131-1142.
- [28] Zayed E. M. E., Abdelaziz M. A. M. Exact solutions for the nonlinear Schrödinger equation with variable coefficients using the generalized extended tanh-function, the sine-cosine and

- the Exp-function methods. *Applied Mathematics and Computation*, 2011, vol. 218, p. 2259-2268.
- [29] Kudryashov N. A. A note on the (G'/G)-expansion method. *Applied Mathematics and Computation*, 2010, vol. 217, p. 1755-1758.
- [30] Wang M., Li X., Zhang J. The (G'/G) expansion method and travelling wave solutions of nonlinear evolution equations in mathematical physics. *Physics Letters A*, 2008, vol. 372, p. 417-423.
- [31] Zayed E. M. E. New travelling wave solutions for higher dimensional nonlinear evolution equations using a generalized (G'/G) expansion method. *Journal of Physics A*, 2009, vol. 42.
- [32] Hafez M. G., Alam M. N., Akbar M. A. Exact traveling wave solutions to the Klein–Gordon equation using the novel (G'/G)-expansion method. *Results in Physics*, 2014, vol. 4.
- [33] Ling-Xiao L., Ming-Liang W. The (G'/G)-expansion method and travelling wave solutions for a higher-order nonlinear Schrödinger equation. *Applied Mathematics and Computation*, 2009, vol. 208, p. 440-445.
- [34] Kudryashov N. A. Simplest equation method to look for exact solutions of nonlinear differential equations. *Chaos Solitons Fractals*, 2005, vol. 24, p. 1217–1231.
- [35] Yusufoglu E., Bekir A. The tanh and the sine–cosine methods for exact solutions of the MBBM and the Vakhnenko equations. *Chaos Solitons Fractals*, 2008, vol. 38, p. 1126–1133.
- [36] Kudryashov N. A., Zakharchenko A. S. Analytical properties and exact solutions of the Lotka–Volterra competition system. *Applied Mathematics and Computation*, 2015, vol. 254, p. 440-445.
- [37] Hirota R. Direct method of finding exact solutions of nonlinear evolution equations. *Lecture Notes in Mathematics*, 1976.
- [38] Rogers C.; Shadwick W. F. Bäcklund transformations and their applications. London: Academic Press, 1982.
- [39] Ablowitz M.; Segur H. Solitons and inverse scattering transform. Philadelphia: SIAM, 1981.
- [40] Kurakin V. L., Kuzmin A. S., Mikhalev A. V., Nechaev A. A. Linear recurring sequences over rings and modules. *Journal of Mathematical Sciences*, 1995, vol. 76, p. 2793-2915.
- [41] Polyanin A. D.; Zaitsev V. F. Handbook of Exact Solutions for Ordinary Differential Equations. Chapman and Hall/CRC, 2003.
- [42] Jürgen G. Modern computer algebra. Cambridge: Cambridge University Press, 2003.

- [43] Ginsburg D.; Goose B.; Taylor J.; Vernescu B. The history of the calculus and the development of computer algebra systems. In: Math WPI [online]. Math WPI, 2015 [viewed 2015-03-18]. Available from: <http://www.math.wpi.edu/IQP/BVCalcHist/calctoc.html>.
- [44] Weisstein E. W. Conic section discriminant. In: Mathworld - A Wolfram Web Resource [online]. Wolfram, 2015 [viewed 2015-03-23]. Available from: <http://mathworld.wolfram.com/ConicSectionDiscriminant.html>.
- [45] Navickas Z., Bikulciene L., Rahula M., Ragulskis M. Algebraic operator method for the construction of solitary solutions to nonlinear differential equations. *Communications in Nonlinear Science and Numerical Simulation*, 2013, vol. 18, p. 1374-1389.

PRIEDAI

A. Žymėjimų sąrašas

\mathbb{R}	realiųjų skaičių aibė;
\mathbb{C}	kompleksinių skaičių aibė;
\mathbb{N}	natūraliųjų skaičių aibė;
\mathbb{Z}_0	neneigiamųjų sveikųjų skaičių aibė;
\mathbb{F}	formaliųjų eilučių aibė (algebra);
i	menamasis vienetas: $i^2 = -1$;
$\Re z$	kompleksinio skaičiaus z realioji dalis;
$\Im z$	kompleksinio skaičiaus z menamoji dalis;
$\text{order } P$	tiesiškai rekurentinės sekos P eilė;

B. Apibendrintųjų diferencialinių operatorių lentelė

B.1 Lentelė: Diferencialinių lygčių sprendinių operatorinės išraiškos

Diferencialinė lygtis ir pradinės sąlygos	Apibendrintasis diferencialinis operatorius	Sprendinio išraiška
$y'_x = P(x, y); \quad y = y(x, c, s),$ $y(c, c, s) = s$	$\mathbf{D} = \mathbf{D}_c + P(c, s)\mathbf{D}_s$	$y(x, c, s) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s$
$y''_{xx} = P(x, y, y'_x); \quad y = y(x, c, s, t),$ $y(c, c, s, t) = s, \quad y'_x(x, c, s, t) \Big _{x=c} = t$	$\mathbf{D} = \mathbf{D}_c + t\mathbf{D}_s + P(c, s, t)\mathbf{D}_t$	$y(x, c, s, t) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s$
$y_x^{(n)} = P(x, y, y'_x, \dots, y_x^{(n-1)}); \quad y = y(x, c, s_0, \dots, s_{n-1}),$ $y(c, c, s_0, \dots, s_{n-1}) = s_0, \quad y_x^{(k)}(x, c, s_0, \dots, s_{n-1}) \Big _{x=c} = s_k,$ $k = 1, \dots, n-1$	$\mathbf{D} = \mathbf{D}_c + \sum_{k=1}^{n-1} s_k \mathbf{D}_{s_k} + P(c, s_1, \dots, s_{n-1}) \mathbf{D}_{s_{n-1}}$	$y(x, c, s_0, \dots, s_{n-1}) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s_0$
$y'_x = P(x, y, z); \quad y = y(x, c, s, t);$ $z'_x = Q(x, y, z); \quad z = z(x, c, s, t);$ $y(c, c, s, t) = s, \quad z(c, c, s, t) = t$	$\mathbf{D} = \mathbf{D}_c + P(c, s, t)\mathbf{D}_s + Q(c, s, t)\mathbf{D}_t$	$y(x, c, s, t) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j s,$ $z(x, c, s, t) = \sum_{j=0}^{+\infty} \frac{(x-c)^j}{j!} \mathbf{D}^j t$

C. Programų išeities tekstai

Visos programos parašytos Maple programavimo kalba.

C.1. Programa P11_1. Pav. 1.1 ir 1.2 braižymas

```

restart;
with(plots):
plotsetup(default);
# KdV solitonas
u := (t,x) -> c/2 * sech(c/2*(x-c*t-x0))^2;
c := 1;
x0 := 0;
P := plot3d(u(t,x), t = -5..5, x = -5..5);
unassign('u');
plotsetup(default);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 12], color = gray);
plotsetup(png,plotoutput=`fig1a_kdv_soliton.png`,plotoptions=`height=2000,width=2000`);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 50], color = gray);

# Sine-Gordon solitonas
u := (t,x) -> 4/lambda * arctan( exp(b*lambda *(k*x+mu*t+theta[0])/(sqrt(b*lambda*(mu^2-a*k^2)) )
));
a := 1;
b := 1;
lambda := 1;
mu := 2;
theta[0] := 1;
k := 1;
P := plot3d(u(t,x), t = -5..5, x = -5..5);
unassign('u');
plotsetup(default);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 12], color = gray, orientation =
[135,45]);
plotsetup(png,plotoutput=`fig1b_SG_soliton.png`,plotoptions=`height=2000,width=2000`);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 50], color = gray, orientation =
[135,45]);

# Kubines Sriodingerio lygties solitonas
u := (t,x) -> sqrt(alpha/q) * sech(sqrt(alpha)*(x-V*t));
alpha := 4;
q := 1;
V := 8;
P := plot3d(u(t,x), t = -5..5, x = -5..5, numpoints = 10000);
unassign('u');
plotsetup(default);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 12], color = gray);
plotsetup(png,plotoutput=`fig2_SCHR_soliton.png`,plotoptions=`height=2000,width=2000`);
display(P, labels = ['t', 'x', 'u'], font = ["TIMES", "ROMAN", 50], color = gray);

```

C.2. Programa P11_1. Pav. 1.3 ir 1.4 braižymas

```

restart;
with(plots):
# KdV solitonas
u := eta -> c/2 * sech(c/2*(eta-x0))^2;
c := 1;
x0 := 0;
P := plot(u(eta), eta = -12..12, thickness = 3, labels = ['eta', 'u'],

```

```

labelfont = ["TIMES", "ROMAN", 14], numpoints = 2000);
display(P);
# Sine-Gordon solitonas
u := eta -> 4/lambda * arctan( exp(b*lambda *(eta+theta[0])/(sqrt(b*lambda*(mu^2-a*k^2)) ) ));
a := 1;
b := 1;
lambda := 1;
mu := 2;
theta[0] := 1;
k := 1;
P := plot(u(eta), eta = -12..12, thickness = 3, labels = ['eta', 'u'],
labelfont = ["TIMES", "ROMAN", 14], numpoints = 2000);
display(P);
# Kubines Sriodingerio lygties solitonas
u := eta -> sqrt(alpha/q) * sech(sqrt(alpha)*(eta));
alpha := 4;
q := 1;
V := 8;
P := plot(u(eta), eta = -7..7, thickness = 3, labels = ['eta', 'u'],
labelfont = ["TIMES", "ROMAN", 14], numpoints = 2000);
display(P);

```

C.3. Programa P137. (2.1.7 poskyris)

```

restart;
with(LinearAlgebra):
# Priskiriamos konkretos parametru reiksmes.
# Norint gauti bendraji sprendini, sito bloko nevykdyti.
#beta := 1;
#y1 := -1;
#y2 := 1;
#expand(beta*(s-y1)*(s-y2));
# Apibreziamas apibendrintasis diferencialinis operatorius atitinkantis
# transformuota (su pakeistu nepriklausomu kintamuoju) Rikati lygti.
DD := qq -> diff(qq,c) + beta/(eta * c) * (s - y1) * (s-y2) * diff(qq,s);
# Apskaiciuojami lygties sprendinio laipsnines eilutes koeficientai.
p[0] := s;
p[1] := DD(p[0]);
p[2] := simplify(DD(p[1]));
p[3] := simplify(DD(p[2]));
p[4] := simplify(DD(p[3]));
# Sukonstruojama antros eiles Hankelio matrica ir apskaiciuojamas jos determinantas
H2 := Matrix([[p[0]/0!, p[1]/1!], [p[1]/1!, p[2]/2!]]);
Determinant(H2);
# Sukonstruojama trecios eiles Hankelio matrica ir apskaiciuojamas jos determinantas
H := Matrix([[p[0]/0!, p[1]/1!, p[2]/2!], [p[1]/1!, p[2]/2!, p[3]/3!], [p[2]/2!, p[3]/3!, p[4]/4!]]);
simplify(Determinant(H));
# Gautas determinantas isskaidomas daugikliais
factor(%);
# Determinantas ivertinamas prie eta reiksmes, kuri vercia ji nuliu
eval(\%, eta = beta*(y1-y2));
simplify(%);
unassign('lambda');
# Sukonstruojama ir sprendziama charakteristine lygtis
HC := Matrix([[p[0], p[1], p[2]/2!], [p[1], p[2]/2!, p[3]/3!], [1, rho, rho^2]]);
Determinant(HC);
RHOS := solve(\% = 0, rho);
eta := beta*(y1-y2);
rho[1] := simplify(RHOS[2], symbolic) assuming s::real;
rho[2] := simplify(RHOS[1], symbolic) assuming s::real;

```

```

# Randami charakteristiniu saknu koeficientai
unassign('l');
LS := solve([l[1] + l[2] = p[0], l[1]*rho[1] + l[2]*rho[2] = p[1]], [l[1], l[2]]);
l[1] := rhs(LS[1][1]);
l[2] := rhs(LS[1][2]);
# Tikrinamos teoremos apie solitoninio sprendinio israiska salygos
simplify(DD(rho[1]) - rho[1]^2);
simplify(DD(rho[2]) - rho[2]^2);
simplify(DD(l[1]) - l[1] * rho[1]);
simplify(DD(l[2]) - l[2] * rho[2]);
# Uzrasomas solitoninis sprendinys (vaizdas)
yHat := (xHat, cHat, s) -> (y1*(s-y2)- y2*(s-y1)*xHat/cHat)/((s-y2) - (s-y1)*xHat/cHat );
# Uraomas sprendinys
y := (x,c,s) -> yHat(exp(eta*x), exp(eta*c), s);
simplify(y(x,c,s),size);
# Sprendinys patikrinamas
simplify(diff(y(x,c,s),x) - beta*(y(x,c,s)-y1)*(y(x,c,s)-y2), size);

```

C.4. Programa P211. (3.1.1 poskyris)

```

restart;
with(PolynomialTools):
local gamma;
# Pastaba: kadangi visur 2.1.1 skyrelyje naudojamas santykis x/c, tai cia trumpumo delei rasysime
tiesiog x.
# Apibreziamas tiriama sistema (vaizdas)
RicSist := [eta*x*diff(y(x),x) = a[0] + a[1]*y(x) + a[2]*y(x)^2 + a[3]*y(x)*z(x),
eta*x*diff(z(x),x) = b[0] + b[1]*z(x) + b[2]*z(x)^2 + b[3]*y(x)*z(x)];
# Apibreziamos pagalbines funkcijos
sideRels := [(x - x1)*(x - x2) = X(x), (x - y1)*(x - y2) = Y(x), (x - z1)*(x - z2) = Z(x)];
# Apibreziami solitoninio sprendinio vaizdai:
y0 := x -> sigma * (x - y1)*(x - y2) / ((x - x1)*(x - x2));
z0 := x -> gamma * (x - z1)*(x - z2) / ((x - x1)*(x - x2));
# y0(x) ir z0(x) statomi i tiriama dif. lygciu sistema
RicSistN := simplify(subs([y(x) = y0(x), z(x) = z0(x)], RicSist), sideRels);
RicSistN := simplify([RicSistN[1] * X(x)^2, RicSistN[2] * X(x)^2], size);
RicSistN[1];
RicSistN[2];
# Formuojama tiesiniu lygciu sistema, kuri gaunama istacius vietoje x ivairias reiksmes i tiriama
sistema
RicSist := subs([X(x) = (x - x1)*(x - x2), Y(x) = (x - y1)*(x - y2), Z(x) = (x - z1)*(x - z2)],
RicSistN);
E[ 1] := simplify(eval(RicSist[1], x = x1)/( sigma * (x1 - y1)*(x1 - y2) ),symbolic);
E[ 2] := simplify(eval(RicSist[1], x = x2)/( sigma * (x2 - y1)*(x2 - y2) ),symbolic);
E[ 3] := simplify(eval(RicSist[2], x = x1)/( gamma * (x1 - z1)*(x1 - z2) ),symbolic);
E[ 4] := simplify(eval(RicSist[2], x = x2)/( gamma * (x2 - z1)*(x2 - z2) ),symbolic);
E[ 5] := simplify(eval(RicSist[1], x = y1),symbolic);
E[ 6] := simplify(eval(RicSist[1], x = y2),symbolic);
E[ 7] := simplify(eval(RicSist[2], x = z1),symbolic);
E[ 8] := simplify(eval(RicSist[2], x = z2),symbolic);
E[ 9] := simplify(eval(RicSist[1], x = 0),symbolic);
E[10] := simplify(eval(RicSist[2], x = 0),symbolic);
#sigma := 4;
#y1 := 10;
#y2 := 2;
#gamma := 3;
#z1 := 5;
#z2 := 1;
#eta := 1;
# Isprendziamos 1-4 lygtys
SOL1 := simplify(op(solve([E[1], E[2], E[3], E[4]], [a[2], a[3], b[2], b[3]])),size);

```

```

# Issprendžiama 5 ir 7 lygtys
SOL2 := solve([E[5], E[7]], [a[0], b[0]]);
# Isprendžiamos 9-10 lygtys
simplify(op(solve([E[9], E[10]], [a[2], b[2]])), size);
# Randami apribojimai kintamiesiems x1, x2
C[1] := simplify(lhs(E[5])/lhs(E[6]) * ((x1-y2)*(x2-y2)/((x1-y1)*(x2-y1)))
= rhs(E[5])/rhs(E[6]) * ((x1-y2)*(x2-y2)/((x1-y1)*(x2-y1))), symbolic);
C[1] := numer(lhs(C[1])) * denom(rhs(C[1])) = numer(rhs(C[1])) * denom(lhs(C[1]));
C[2] := simplify(lhs(E[7])/lhs(E[8]) * ((x1-z2)*(x2-z2)/((x1-z1)*(x2-z1)))
= rhs(E[7])/rhs(E[8]) * ((x1-z2)*(x2-z2)/((x1-z1)*(x2-z1))), symbolic);
C[2] := numer(lhs(C[2])) * denom(rhs(C[2])) = numer(rhs(C[2])) * denom(lhs(C[2]));
C[1] := lhs(expand(C[1]))-rhs(expand(C[1]));
C[1] := simplify(C[1], size);
C[2] := lhs(expand(C[2]))-rhs(expand(C[2]));
C[2] := simplify(C[2], size);
C[1] := -y1^2*y2+(-y2^2+2*XSUM*y2-XPROD)*y1-XPROD*y2;
C[2] := -z1^2*z2+(-z2^2+2*XSUM*z2-XPROD)*z1-XPROD*z2;
SOLDELTA := solve([C[1] = 0, C[2] = 0], [XSUM, XPROD]);
simplify(SOLDELTA[1],size);
Delta[0] := simplify(denom(rhs(SOLDELTA[1][1])),size);
Delta[0] := ((2*z1+2*z2)*y2-2*z1*z2)*y1-2*y2*z1*z2;
Delta[1] := -simplify(numer(rhs(SOLDELTA[1][1])),size);
Delta[1] := (z1+z2)*(y1+y2)*(y1*y2-z1*z2);
Delta[2] := 2*simplify(numer(rhs(SOLDELTA[1][2])),size);
Delta[2] := 2*y1*y2*z1*z2*(y1+y2-z1-z2);
x1 := (1/(2*Delta[0])) * (Delta[1] + sqrt(Delta[1]^2 - 4 * Delta[0] * Delta[2]));
x2 := (1/(2*Delta[0])) * (Delta[1] - sqrt(Delta[1]^2 - 4 * Delta[0] * Delta[2]));
# Su jau zinomais x1 ir x2 isprendžiamos jau anksčiau isspreistos lygtys:
simplify(op(solve([E[1], E[2], E[3], E[4], E[5], E[7], E[9], E[10]],
[a[0], a[1], a[2], a[3], b[0], b[1], b[2], b[3]])), size);

```

C.5. Programa P213. (3.1.3 poskyris)

```

restart;
local gamma:
with(LinearAlgebra):
# Apibrezžiama antros eiles kreivės lygtis
conic := A*y(x)^2 + 2*B*y(x)*z(x) + C*z(x)^2 + DD*y(x) + E*z(x) = 1;
# I lygti istatome solitono israiska
conic := simplify(subs([y(x) = sigma*Y(x)/X(x), z(x) = gamma*Z(x)/X(x)], conic*X(x)^2), symbolic)
;
X := x -> (x-x1)*(x-x2);
Y := x -> (x-y1)*(x-y2);
Z := x -> (x-z1)*(x-z2);
# Formuojame tiesiniu lygciu sistema
Y(y1) := 0: Y(y2) := 0: Z(z1) := 0: Z(z2) := 0: X(x1) := 0: X(x2) := 0:
E[1] := eval(conic, x = y1);
E[2] := eval(conic, x = y2);
E[3] := eval(conic, x = z1);
E[4] := eval(conic, x = z2);
E[5] := eval(conic, x = x1);
E[6] := eval(conic, x = x2);
# Formuojamas sistemos determinantas
MA := Matrix(6,5):
MB := Vector[column](6):
for j from 1 to 6 do;
    MA[j,1] := coeff(lhs(E[j]), A);
    MA[j,2] := coeff(lhs(E[j]), B);
    MA[j,3] := coeff(lhs(E[j]), C);
    MA[j,4] := coeff(lhs(E[j]), DD);
    MA[j,5] := coeff(lhs(E[j]), E);

```



```

    MB[j] := rhs(E[j]);
end do:
Determinant(<MA|MB>);
Rank(MA);
Rank(<MA|MB>);
# Isplestines ir sistemos matricu rangai sutampa, vadinasi egzistuoja vienintelis sprendinys.
unassign('Y','X','Z');
SOL := simplify(LinearSolve(MA, MB),size):
A := SOL[1];
B := SOL[2];
C := SOL[3];
DD := SOL[4];
E := SOL[5];
# Skaiciuojame diskriminanta
dsc := simplify(A*C - B^2, size);

```

C.6. Programa P222. (3.2.2 poskyris)

```

restart;
with(LinearAlgebra):
with(PolynomialTools):
# Apibreziamas apibendrintasis diferencialinis operatorius
DD := q -> diff(q,c) + 1 / (eta * c) * (a[0] + a[1] * s + a[2] * s^2 + b[2] * s * t) *
diff(q,s) + 1 / (eta*c) * (b[0] + b[1] * t + b[2] * t^2 + a[2] * s * t) * diff(q,t);
# Apskaiciuojamos reikiamos koeficientu, priklausanciu nuo s ir t sekos
p[1] := DD(s):
pp[1] := p[1]:
for k from 2 to 5 do;
    p[k] := simplify(DD(p[k-1]),symbolic);
    pp[k] := p[k] / k!;
od:
# Suformuojama Hankelio matrica ir apskaiciuojamas jos determinantas
H := Matrix([[pp[1], pp[2], pp[3]], [pp[2], pp[3], pp[4]], [pp[3], pp[4], pp[5]]]):
d := collect(Determinant(H) * eta^9 * c^9, eta):
# Surandami koeficientai A6, A4, A2, A0:
A[6] := simplify(coeff(d, eta^6),size);
K := simplify(A[6]^(1/3), symbolic);
A[4] := simplify(coeff(d, eta^4),size):
A[2] := simplify(coeff(d, eta^2),size):
A[0] := simplify(d - A[6] * eta^6 - A[4] * eta^4 - A[2] * eta^2, size):
indets(A[0]);
# Parodoma, kad A4 dalijasi is trecio laipsnio saknies is A6:
A[4] := factor(A[4]);
# Randamos s0 ir t0 reiksmes su kuriomis K = 0
ST0 := op(solve([A[6] = 0, A[4] = 0], [s,t]));
t := rhs(ST0[2]);
# Istatome i tiriama determinanta
d := collect(d, eta):
simplify(A[6]);
simplify(A[4]);
# Suprastiname A[2] ir A[0] israiskas:
A[2] := simplify(A[2], size);
A[0] := simplify(A[0], size);
# Apskaiciuojame A[0]/A[2]:
R := A[0]/A[2];
# Sudauginame koeficientus prie s ir konstantas kryzmiskai
alpha[1] := coeff( numer(R), s);
beta[1] := tcoeff( numer(R), s);
alpha[2] := coeff( denom(R), s);
beta[2] := tcoeff( denom(R), s);
simplify(alpha[1]*beta[2] - alpha[2] * beta[1], symbolic);

```

```

# Analogiskus veiksmus padarome kitam determinantui, Wz:
restart;
with(LinearAlgebra):
# Apibreziamas apibendrintasis diferencialinis operatorius
DD := q -> diff(q,c) + 1 / (eta * c) * (a[0] + a[1] * s + a[2] * s^2 + b[2] * s * t) *
diff(q,s) + 1 / (eta*c) * (b[0] + b[1] * t + b[2] * t^2 + a[2] * s * t) * diff(q,t);
# Apskaiciuojamos reikiamos koeficientu, priklausanciu nuo s ir t sekos
p[1] := DD(t):
pp[1] := p[1]:
for k from 2 to 5 do;
  p[k] := simplify(DD(p[k-1]),symbolic);
  pp[k] := p[k] / k!;
od:
# Suformuojama Hankelio matrica ir apskaiciuojamas jos determinantas
H := Matrix([[pp[1], pp[2], pp[3]], [pp[2], pp[3], pp[4]], [pp[3], pp[4], pp[5]]]):
d := collect(Determinant(H) * eta^9 * c^9, eta):
# Surandami koeficientai A6, A4, A2, A0:
A[6] := simplify(coeff(d, eta^6),size);
K := simplify(A[6]^(1/3), symbolic);
A[4] := simplify(coeff(d, eta^4),size):
A[2] := simplify(coeff(d, eta^2),size):
A[0] := simplify(d - A[6] * eta^6 - A[4] * eta^4 - A[2] * eta^2, size):
indets(A[0]);
# Parodoma, kad A4 dalijasi is trecio laipsnio saknies is A6:
A[4] := factor(A[4]);
# Randamos s0 ir t0 reiksmes su kuriomis K = 0
ST0 := op(solve([A[6] = 0, A[4] = 0], [s,t]));
s := rhs(ST0[1]);
# Istatome i tiriama determinanta
d := collect(d, eta):
simplify(A[6]);
simplify(A[4]);
# Suprastiname A[2] ir A[0] israiskas:
A[2] := simplify(A[2], size);
A[0] := simplify(A[0], size);
# Apskaiciuojame A[0]/A[2]:
R := A[0]/A[2];
# Sudauginame koeficientus prie s ir konstantas kryziskai
alpha[1] := coeff(numer(R), t);
beta[1] := tcoeff(numer(R), t);
alpha[2] := coeff(denom(R), t);
beta[2] := tcoeff(denom(R), t);
simplify(alpha[1]*beta[2] - alpha[2] * beta[1], symbolic);

```

C.7. Programa P223. (3.2.3 poskyris)

```

restart;
with(LinearAlgebra):
with(PolynomialTools):
# Priskiriamos lygciu sistemos koeficientu reiksmes
a[0] := -62; a[1] := 2420/73; a[2] := -1095/248;
b[0] := 186/73; b[1] := 2453/146; b[2] := 8/2263;
eta := 1;
# Apibreziamas apibendrintasis diferencialinis operatorius
DD := q -> diff(q,c) + 1 / (eta * c) * (a[0] + a[1] * s + a[2] * s^2 + b[2] * s * t) * diff(q,s)
+ 1 / (eta*c) * (b[0] + b[1] * t + b[2] * t^2 + a[2] * s * t) * diff(q,t);
# Apskaiciuojamos reikiamos koeficientu, priklausanciu nuo s ir t sekos
p[1] := DD(s):
pp[1] := p[1]:

```

```

for k from 2 to 5 do;
  p[k] := simplify(DD(p[k-1]),symbolic);
  pp[k] := p[k] / k!;
od:
# Konstruojama charakteristine lygtis koeficientams pp
H := Matrix([ [pp[1], pp[2], pp[3]], [pp[2], pp[3], pp[4]], [1, rho, rho^2] ]):
RHOP := solve(Determinant(H) = 0, rho):
rhop[1] := simplify(RHOP[1], size);
rhop[2] := simplify(RHOP[2], size);
# Randami sekos pp[j] koeficientai lambda
eq := lambda[1] * rho[1] + lambda[2] * rho[2] = P[1],
      lambda[1] * rho[1]^2 + lambda[2] * rho[2]^2 = P[2];
SLAMBDA := op(solve([eq], [lambda[1], lambda[2]]));
SLAMBDA := subs([P[1] = pp[1], P[2] = pp[2], rho[1] = rhop[1], rho[2] = rhop[2]], SLAMBDA):
lambda[1] := simplify(rhs(SLAMBDA[1]),symbolic);
lambda[2] := simplify(rhs(SLAMBDA[2]),symbolic);
# Tikrinamos salygos
simplify(DD(rhop[1]) - rhop[1]^2);
simplify(DD(rhop[2]) - rhop[2]^2);
simplify(DD(lambda[1]) - lambda[1] * rhop[1]);
simplify(DD(lambda[2]) - lambda[2] * rhop[2]);
simplify(lambda[1] + lambda[2],size);
# Apskaiciuojamos reikiamos koeficientu, priklausanciu nuo s ir t sekos
q[1] := DD(t):
qq[1] := q[1]:
for k from 2 to 5 do;
  q[k] := simplify(DD(q[k-1]),symbolic);
  qq[k] := q[k] / k!;
od:
# Konstruojama charakteristine lygtis koeficientams qq
H := Matrix([ [qq[1], qq[2], qq[3]], [qq[2], qq[3], qq[4]], [1, rho, rho^2] ]):
RHOQ := solve(Determinant(H) = 0, rho):
rhoq[1] := simplify(RHOQ[1], size);
rhoq[2] := simplify(RHOQ[2], size);
# Patikrinama ar gautos tos pacios saknys kaip ir su seka pp
simplify(rhoq[1]-rhop[1], size);
simplify(rhoq[2]-rhop[2], size);
# Randami sekos qq[j] koeficientai mu
eq := mu[1] * rho[1] + mu[2] * rho[2] = Q[1],
      mu[1] * rho[1]^2 + mu[2] * rho[2]^2 = Q[2];
SMU := op(solve([eq], [mu[1], mu[2]]));
SMU := subs([Q[1] = qq[1], Q[2] = qq[2], rho[1] = rhoq[1], rho[2] = rhoq[2]], SMU):
mu[1] := simplify(rhs(SMU[1]),symbolic);
mu[2] := simplify(rhs(SMU[2]),symbolic);
# Tikrinamos salygos
simplify(DD(rhoq[1]) - rhoq[1]^2);
simplify(DD(rhoq[2]) - rhoq[2]^2);
simplify(DD(mu[1]) - mu[1] * rhoq[1]);
simplify(DD(mu[2]) - mu[2] * rhoq[2]);
simplify(mu[1] + mu[2],size);

```

C.8. Programa P231. (3.3.1 poskyris)

```

restart;
with(plots):
with(DEtools):
with(LinearAlgebra):
# Apibreziamė algebrine lygciū sistema pusiausvyros taskams rasti
eqns := a[0] + a[1]*y + a[2]*y^2 + b[2]*y*z = 0,
        b[0] + b[1]*z + b[2]*z^2 + a[2]*y*z = 0;
# Sistema isprendžiama

```

```
alias(alpha = RootOf((-a[1]*b[2]+b[1]*b[2])*_Z^3+(a[0]*a[2]-a[1]*b[1]+b[0]*b[2]+b[1]^2)*_Z^2
+(-a[1]*b[0]+2*b[0]*b[1])*_Z+b[0]^2)):
EQUIL := op(solve([eqns], [z,y]));
EQUIL;
```

C.9. Programa P232. (3.3.2 poskyris)

```
# Fazinio portreto braizymas
restart;
with(DEtools): with(plots):
# Apibreziamą tiriama sistema
sist := diff(yy(x), x) = 136/11-(828/319)*yy(x)+(29/187)*yy(x)^2-(550/1479)*yy(x)*zz(x), diff(zz(
x), x) = -51/29+(345/319)*zz(x)-(550/1479)*zz(x)^2+(29/187)*yy(x)*zz(x);
# Randami pusiausvyros taskai
SOL := solve([rhs(sist[1]), rhs(sist[2])], [yy(x), zz(x)]);
N := numelems(SOL):
R := []:
for j from 1 to N do:
  R := [op(R), [rhs(SOL[j][1]), rhs(SOL[j][2])]]:
od:
# Apibreziamos pradines salygos
c := 0:
init := [[yy(c) = 10, zz(c) = 4], [yy(c) = 596258/89593, zz(c) = 20910/12799], [yy(c) = 12, zz(c)
= 2], [yy(c) = 596258/89593, zz(c) = 20910/12799-0.5], [yy(c) = 4, zz(c) = 5], [yy(c) = 1,
zz(c) = 1/4], [yy(c) = 3, zz(c) = 1/2], [yy(c) = 13, zz(c) = 0], [yy(c) = 4, zz(c) = 1], [yy(
c) = 0, zz(c) = 2], [yy(c) = 0, zz(c) = 3], [yy(c) = 8, zz(c) = 0], [yy(c) = 10, zz(c) = 0]]:
init2 := [[yy(c) = 16/5, zz(c) = 167315713/4000000], [yy(c) = 14, zz(c) = 11207/20000], [yy(c) =
10, zz(c) = 814107/1000000], [yy(c) = 10, zz(c) = 814107/1000000]]:
init3 := [[yy(c) = 136/23-0.0000001, zz(c) = 51/46], [yy(c) = 136/23+0.0000001, zz(c) = 51/46]]:
# Priskiriami parametrai ir braizomi grafikai
xv := 16;
yv := 5;
P1 := DEplot([sist], [yy(x), zz(x)], x = (c-20)..(c+20), yy = 0..xv, zz = 0..yv, init, linecolor
= black, arrows = medium, color = red, numpoints = 2000, linecolor = black, thickness = 8,
obsrange = false):
P4 := DEplot([sist], [yy(x), zz(x)], x = (c-20)..(c+20), yy = 0..xv, zz = 0..yv, init2, linecolor
= black, arrows = none, color = red, numpoints = 2000, linestyle = dash, thickness = 8,
obsrange = false):
P5 := DEplot([sist], [yy(x), zz(x)], x = (c-50)..(c+50), yy = 0..xv, zz = 0..yv, init3, linecolor
= black, arrows = none, color = red, numpoints = 2000, linestyle = dot, thickness = 8,
obsrange = false):
P2 := pointplot([R[1], R[2]], symbol = solidcircle, symbolsize = 30, color = blue);
P3 := pointplot([R[3]], symbol = soliddiamond, symbolsize = 30, color = blue);
P7 := implicitplot(6365529*s^2-30528300*s*t+36602500*t^2-171649782*s-312252600*t+1558425564, s =
0..16, t=0..5, color = black, thickness = 5, linestyle = dash);
P8 := plot([- (87/1100)*s+867/550, - (87/88)*s+153/22], s = 0..16, color = black, linestyle = dot,
thickness = 5);
P6 := pointplot([6.5, 6.5*(- (87/88))+153/22], symbol = cross, symbolsize = 20, color = blue);
plotsetup('default');
PL := display(P1,P7,P8,P2, P3, P6,view = [0..16,0..5]):
display(PL);
plotsetup(png,plotoutput='phase_arrows_case4dot2.png',plotoptions='height=2000,width=2000');
display(P1, P4, P5, P2, P3, labels = ['y', 'z'], axesfont = ["HELVETICA", "ARIAL", 40], labelfont
= ["HELVETICA", "ARIAL", 45]);
# Apskaiciuojami sistemos sprendinio parametrai
restart;
with(DEtools):
with(plots):
sist := diff(yy(x),x) = a[0] + a[1] * yy(x) + a[2] * yy(x)^2 + b[2] * yy(x) * zz(x),
diff(zz(x),x) = b[0] + b[1] * zz(x) + b[2] * zz(x)^2 + a[2] * yy(x) * zz(x);
```

```

DD := q-> diff(q,c) + 1 / (eta * c) * (a[0] + a[1] * s + a[2] * s^2 + b[2] * s * t) * diff(q,s) +
      1 / (eta*c) * (b[0] + b[1] * t + b[2] * t^2 + a[2] * s * t) * diff(q,t);
a[0] := 136/11; a[1] := -828/319; a[2] := 29/187;
b[0] := -51/29; b[1] := 345/319; b[2] := -550/1479;
eta := 1;

sist;
rho[1] := (-48960*c + 5046 * c * s -12100*c*t + 2 * sqrt(c^2 * (-553146 * (-1734 + 87*s + 1100*t)
      + (24480-2523*s+6050*t)^2 ) ) ) / (65076*c^2);
rho[1] := simplify(rho[1]) assuming c > 0;
rho[1] := simplify(rho[1],size);
solve(rho[1] = 0, t);
rho[2] := (-48960*c + 5046 * c * s -12100*c*t - 2 * sqrt(c^2 * (-553146 * (-1734 + 87*s + 1100*t)
      + (24480-2523*s+6050*t)^2 ) ) ) / (65076*c^2);
rho[2] := simplify(rho[2]) assuming c > 0;
rho[2] := simplify(rho[2],size);
solve(rho[2] = 0, t);
# Patikrinimas rho
simplify(DD(rho[1])-rho[1]^2);
simplify(DD(rho[2])-rho[2]^2);
rhat[1] := simplify(c * rho[1]) assuming c > 0;
rhat[2] := simplify(c * rho[2]) assuming c > 0;
latex(rhat[1]);
x[1] := simplify( (rhat[1] + 1) / rhat[1] );
x[2] := simplify( (rhat[2] + 1) / rhat[2] );
# Randamos balno tasko daugdaros
solve(rhat[1] = -1, t);
solve(rhat[2] = -1, t);
simplify(eval(rhat[1], t = -(87/88)*s+153/22));
simplify(eval(rhat[2], t = -(87/88)*s+153/22));
solve(rhat[1] = 0, t);
solve(rhat[2] = 0, t);
simplify(eval(rhat[1], t = -(87/1100)*s+867/550));
simplify(eval(rhat[2], t = -(87/1100)*s+867/550));

p[1] := DD(s):
p[2] := DD(p[1]):
pp[1] := simplify(p[1]):
pp[2] := 1/2 * simplify(p[2]):
SOLLAMBDA := solve([lambda[1] * rho[1] + lambda[2] * rho[2] = pp[1], lambda[1] * rho[1]^2 +
      lambda[2] * rho[2]^2 = pp[2]], [lambda[1], lambda[2]]):
lambda[2] := rhs(SOLLAMBDA[1][2]):
lambda[2] := simplify(lambda[2]) assuming c > 0:
lambda[2] := simplify(lambda[2],size);
lambda[1] := rhs(SOLLAMBDA[1][1]):
lambda[1] := simplify(lambda[1]) assuming c > 0:
lambda[1] := simplify(lambda[1],size);
sigma := simplify(s - (lambda[1] + lambda[2]));
p[1] := DD(t):
p[2] := DD(p[1]):
pp[1] := simplify(p[1]):
pp[2] := 1/2 * simplify(p[2]):
SOLMU := solve([mu[1] * rho[1] + mu[2] * rho[2] = pp[1], mu[1] * rho[1]^2 + mu[2] * rho[2]^2 = pp
      [2]], [mu[1], mu[2]]):

mu[2] := rhs(SOLMU[1][2]):
mu[2] := simplify(mu[2]) assuming c > 0:
mu[2] := simplify(mu[2],size);
mu[1] := rhs(SOLMU[1][1]):
mu[1] := simplify(mu[1]) assuming c > 0:
mu[1] := simplify(mu[1],size);
local gamma;

```

```

gamma := simplify(t - (mu[1] + mu[2]));
# Patikrinimas lambda
simplify(DD(lambda[1]) - rho[1] * lambda[1]);
simplify(DD(lambda[2]) - rho[2] * lambda[2]);
# Patikrinimas mu
simplify(DD(mu[1]) - rho[1] * mu[1]);
simplify(DD(mu[2]) - rho[2] * mu[2]);

deltay := simplify(( rhohat[1] * (sigma * rhohat[2] + sigma + lambda[2]) + rhohat[2] * (sigma *
  rhohat[1] + sigma + lambda[1]) ) / (sigma * rhohat[1] * rhohat[2]));
deltahaty := simplify((sigma * (rhohat[1] + 1) * (rhohat[2] + 1) + lambda[2] * (rhohat[1] + 1) +
  lambda[1] * (rhohat[2] + 1)) / (sigma * rhohat[1] * rhohat[2]));
deltaz := simplify((rhohat[1] * (gamma * rhohat[2] + gamma + mu[2]) + rhohat[2] * (gamma * rhohat
  [1] + gamma + mu[1]) ) / (gamma * rhohat[1] * rhohat[2]));
deltahatz := simplify((gamma * (rhohat[1] + 1) * (rhohat[2] + 1) + mu[2] * (rhohat[1] + 1) + mu
  [1] * (rhohat[2] + 1) ) / (gamma * rhohat[1] * rhohat[2]));
y[1] := (deltay + sqrt(deltay^2 - 4 * deltahaty))/2;
y[2] := (deltay - sqrt(deltay^2 - 4 * deltahaty))/2;
z[1] := (deltaz + sqrt(deltaz^2 - 4 * deltahatz)) / 2;
z[2] := (deltaz - sqrt(deltaz^2 - 4 * deltahatz))/2;

```

C.10. Programa P233. (3.3.3 poskyris)

```

# Apskaiciuojami sistemos parametrai
restart;
with(DEtools):
with(plots):
sist := diff(yy(x),x) = a[0] + a[1] * yy(x) + a[2] * yy(x)^2 + b[2] * yy(x) * zz(x),
  diff(zz(x),x) = b[0] + b[1] * zz(x) + b[2] * zz(x)^2 + a[2] * yy(x) * zz(x);
DD := q-> diff(q,c) + 1 / (eta * c) * (a[0] + a[1] * s + a[2] * s^2 + b[2] * s * t) * diff(q,s) +
  1 / (eta*c) * (b[0] + b[1] * t + b[2] * t^2 + a[2] * s * t) * diff(q,t);
a[0] := 6; a[1] := 0; a[2] := -1/12;
b[0] := -5; b[1] := 0; b[2] := 1/10;
eta := 1;
rho[1] := ( c * (-60 - 5 * s + 6 * t) - sqrt( c^2 * (60+5*s-6*t) * (5*s - 6 * (10+t)) ) ) / (120*
  c^2);
rho[1] := simplify(rho[1]) assuming c > 0;
rho[1] := simplify(rho[1],size);
rho[2] := ( c * (-60 - 5 * s + 6 * t) + sqrt( c^2 * (60+5*s-6*t) * (5*s - 6 * (10+t)) ) ) / (120*
  c^2);
rho[2] := simplify(rho[2]) assuming c > 0;
rho[2] := simplify(rho[2],size);
solve(rho[2] = 0 , t);
# Patikrinimas rho
simplify(DD(rho[1])-rho[1]^2);
simplify(DD(rho[2])-rho[2]^2);
rhohat[1] := simplify(c * rho[1]) assuming c > 0;
rhohat[2] := simplify(c * rho[2]) assuming c > 0;

# Randama separatrise
solve(rhohat[1] = -1, t);
solve(rhohat[1] = 0, t);
x[1] := simplify( (rhohat[1] + 1) / rhohat[1] );
x[2] := simplify( (rhohat[2] + 1) / rhohat[2] );
lambda[2] := ( c*(-6+s)*(60+5*s-6*t) - (6+s) * sqrt(c^2 * (60+5*s-6*t) * (5*s - 6 * (10+t))) ) /
  ( 2*c*(60+5*s-6*t) );
lambda[2] := simplify(lambda[2]) assuming c > 0;
lambda[2] := simplify(lambda[2],size);
lambda[1] := ( c*(-6+s)*(60+5*s-6*t) + (6+s) * sqrt(c^2 * (60+5*s-6*t) * (5*s - 6 * (10+t))) ) /
  ( 2*c*(60+5*s-6*t) );
lambda[1] := simplify(lambda[1]) assuming c > 0;

```

```

lambda[1] := simplify(lambda[1],size);
sigma := simplify(s - (lambda[1] + lambda[2]));

mu[2] := ( c*(60+5*s - 6*t)*(5+t) - (-5+t) * sqrt(c^2 * (60+5*s - 6*t) * (5*s - 6*(10+t)) ) ) /
( 2*c*(60+5*s-6*t) );
mu[2] := simplify(mu[2]) assuming c > 0;
mu[2] := simplify(mu[2],size);
mu[1] := ( c*(60+5*s - 6*t)*(5+t) + (-5+t) * sqrt(c^2 * (60+5*s - 6*t) * (5*s - 6*(10+t)) ) ) /
( 2*c*(60+5*s-6*t) );
mu[1] := simplify(mu[1]) assuming c > 0;
mu[1] := simplify(mu[1],size);
local gamma;
gamma := simplify(t - (mu[1] + mu[2]));
# Patikrinimas lambda
simplify(DD(lambda[1]) - rho[1] * lambda[1]);
simplify(DD(lambda[2]) - rho[2] * lambda[2]);
# Patikrinimas mu
simplify(DD(mu[1]) - rho[1] * mu[1]);
simplify(factor(DD(mu[2]) - rho[2] * mu[2]));

deltay := simplify(( rhohat[1] * (sigma * rhohat[2] + sigma + lambda[2]) + rhohat[2] * (sigma *
rhohat[1] + sigma + lambda[1]) ) / (sigma * rhohat[1] * rhohat[2]));
deltahaty := simplify((sigma * (rhohat[1] + 1) * (rhohat[2] + 1) + lambda[2] * (rhohat[1] + 1) +
lambda[1] * (rhohat[2] + 1)) / (sigma * rhohat[1] * rhohat[2]));
deltaz := simplify((rhohat[1] * (gamma * rhohat[2] + gamma + mu[2]) + rhohat[2] * (gamma * rhohat
[1] + gamma + mu[1]) ) / (gamma * rhohat[1] * rhohat[2]));
deltahatz := simplify((gamma * (rhohat[1] + 1) * (rhohat[2] + 1) + mu[2] * (rhohat[1] + 1) + mu
[1] * (rhohat[2] + 1) ) / (gamma * rhohat[1] * rhohat[2]));
y[1] := (deltay + sqrt(deltay^2 - 4 * deltahaty))/2;
y[2] := (deltay - sqrt(deltay^2 - 4 * deltahaty))/2;
z[1] := (deltaz + sqrt(deltaz^2 - 4 * deltahatz)) / 2;
z[2] := (deltaz - sqrt(deltaz^2 - 4 * deltahatz))/2;
# Sprendinys
Y := xx -> sigma* ( (exp(eta*(xx-c)) - y[1]) * (exp(eta*(xx-c)) - y[2]) ) / ( (exp(eta*(xx-c)) -
x[1]) * (exp(eta*(xx-c)) - x[2]) );
Z := xx -> gamma * ( (exp(eta*(xx-c)) - z[1]) * (exp(eta*(xx-c)) - z[2]) ) / ( (exp(eta*(xx-c)) -
x[1]) * (exp(eta*(xx-c)) - x[2]) );

# Braizomas fazinis portretas
restart;
with(DEtools): with(plots):
sist := diff(yy(x), x) = 6-(1/12)*yy(x)^2+(1/10)*yy(x)*zz(x), diff(zz(x), x) = -5+(1/10)*zz(x)
^2-(1/12)*yy(x)*zz(x);
# Randmi pusiausvyros taskai
solve([rhs(sist[1]), rhs(sist[2])], [yy(x), zz(x)]);
SOL := solve([rhs(sist[1]), rhs(sist[2])], [yy(x), zz(x)]);
N := numelems(SOL):
R := [];
for j from 1 to N do:
  R := [op(R), [rhs(SOL[j][1]), rhs(SOL[j][2])]];
od:
c := 0;
init := [[yy(c) = 10, zz(c) = 10], [yy(c) = -10, zz(c) = -10], [yy(c) = 5, zz(c) = 5], [yy(c) =
-5, zz(c) = -5], [yy(c) = 15, zz(c) = 15], [yy(c) = -15, zz(c) = -15], [yy(c) = 25, zz(c) =
-5], [yy(c) = 6, zz(c) = -20], [yy(c) = -15, zz(c) = 7], [yy(c) = 0, zz(c) = -20]];
xv := 25;
yv := 25;
P1 := DEplot([sist], [yy(x), zz(x)], x = (c-15)..(c+15), yy = -xv..xv, zz = -yv..yv, init,
linecolor = black, arrows = medium, color = red, numpoints = 2000, linecolor = black,
thickness = 8, obsrange = false):
P2 := pointplot(R, symbol = solidcircle, symbolsize = 30, color = blue):

```

```
P3 := plot([(5/6)*x + 10, (5/6) * x - 10], x = -25..25, color = black, thickness = 8, linestyle =
    dash);
plotsetup('default');
PL := display(P1, P3, P2);
display(PL);
plotsetup(png,plotoutput=`phase_arrows_case2.png`,plotoptions=`height=2000,width=2000`);
display(P1, P2, P3, labels = ['y', 'z'], axesfont = ["HELVETICA", "ARIAL", 40], labelfont = ["
    HELVETICA", "ARIAL", 45]);
```