



KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

Agneta Šmergelytė

**TREČIOS EILĖS MATRICŲ
ALGEBRINIŲ - STRUKTŪRINIŲ
PARAMETRŲ IŠSKYRIMAS IR TYRIMAS**

Magistro darbas

Vadovas
prof. dr. Z. Navickas

KAUNAS, 2014



KAUNO TECHNOLOGIJOS UNIVERSITETAS
MATEMATIKOS IR GAMTOS MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
doc. dr. N. Listopadskis
2014 06 05

TREČIOS EILĖS MATRICŲ
ALGEBRINIŲ - STRUKTŪRINIŲ
PARAMETRŲ IŠSKYRIMAS IR TYRIMAS

Taikomosios matematikos magistro baigiamasis darbas

Vadovas
prof. dr. Z. Navickas
2014 06 02

Recenzentas
dr. K. Berškienė
2014 06 02

Atliko
FMMM-2 gr. stud.
A. Šmergelytė
2014 05 30

KAUNAS, 2014

KVALIFIKACINĖ KOMISIJA

Pirmininkas: Juozas Augutis, profesorius (VDU)

Sekretorius: Eimutis Valakevičius, profesorius (KTU)

Nariai: Jonas Valantinas, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Kristina Šutienė, docentė (KTU)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, dr., direktoriaus pavaduotojas (UAB „Danet Baltic“)

Šmergelytė A. Trečios eilės matricių algebrinių-struktūrinių parametrų išskyrimas ir tyrimas: Taikomosios matematikos magistro baigiamasis darbas / vadovas prof. Dr. Z. Navickas; Taikomosios matematikos katedra, Matematikos ir gamtos mokslų fakultetas, Kauno technologijos universitetas. – Kaunas, 2014. – 106 p.

SANTRAUKA

Viena svarbiausių problemų medicinos srityje yra širdies ir kraujagyslių ligų diagnostika. Pagrindinis kardiologinis tyrimas yra atliekamas elektrokardiogramos (EKG) pagalba, tačiau žinomi EKG analizės metodai ne visada atskleidžia pakankamą klinikinės informacijos kiekį. Todėl šio darbo tikslas yra pritaikyti algebrinės-struktūrinės matricių analizės algoritmus EKG parametrų dinaminėms sąsajoms tirti. Signalų dinaminių sąsajų tyrimas buvo atliekamas kointegruojant juos į trečios eilės Lagranžo skirtumų matricių sekas, su kuriomis atliekamos algebrinės matricių transformacijos. Tokiu būdu gaunami algebriniai-struktūriniai parametrai, pagal kuriuos sprendžiama apie signalų tarpusavio sąsajas.

Pateikti algoritmai buvo realizuoti programinėmis priemonėmis ir pastebėta, kad gauti algebriniai-struktūriniai parametrai atspindi signalų dinamines sąsajas. EKG duomenų analizei atlikti buvo paimta dešimties sportininkų elektrokardiografiniai duomenys, iš kurių nagrinėjami trys EKG parametrai: RR, DQRS ir DJT. Atlikus šių duomenų tyrimą pastebėta, kad gauti algebriniai-struktūriniai parametrai bei jų sklaida apie vidurkį atkartoja fiziologines sportininkų būsenas ir apibūdina būsenų kitimo specifiką jėgos, greitumo bei ištvėmės sportuose.

Šmergelytė A. Distinction and analysis of third order matrices algebraic-structural parameters: Master's work in applied mathematics / supervisor dr. prof. Z. Navickas; Department of Applied mathematics, Faculty of Mathematics and Natural Sciences, Kaunas University of Technology. – Kaunas, 2014. – 106 p.

SUMMARY

Cardiovascular disease diagnosis is one of the most important problems in medical studies. Main cardiological study is performed with the help of an electrocardiogram (ECG). However, known ECG analysis methods do not always reveal a sufficient amount of clinical information. The goal of this research is to adapt algorithms of algebraic-structural matrix analysis to the examination of ECG parameters dynamic connections. The research was performed with cointegration of signals to the matrices of Lagrange distributions. With sequences of these matrices, algebraic matrix transformations were made and algebraic-structural parameters were found. Based on these parameters, analysis of signals dynamic connections was performed.

Presented algorithms were implemented using software tools. Analysis revealed that algebraic-structural parameters are representing dynamic connections of signals. In order to perform the analysis of ECG parameters, data was taken from 10 cardiograms which are taken from athletes. Three ECG parameters were analysed: RR, DQRS and DJT. Analysis showed that obtained algebraic-structural parameters and their standard deviation represents physiological status of athletes and describes the specifics of phase transitions in power, speed and endurance sports.

TURINYS

LENTELIŲ SĄRAŠAS	7
PAVEIKSLŲ SĄRAŠAS	7
ĮVADAS	8
1. BENDROJI DALIS	10
1.1. KOMPLEKSNĖS SISTEMOS	10
1.2. ŠIRDIS KAIP KOMPLEKSNĖS ADAPTYVIOS SISTEMOS DALIS	10
1.3. ELEKTROKARDIOGRAMA.....	11
1.4. LAGRANŽO SKIRTUMŲ MATRICA.....	13
1.5. MATRICOS EILUČIŲ IR STULPELIŲ STRUKTŪROS	13
1.6. PAGRINDINĖS ALGEBRINIŲ MATRICŲ STRUKTŪROS	14
1.7. IDEMPOTENTŲ IR NULPOTENTŲ SISTEMOS.....	15
1.8. MATRICOS IŠDĖSTYMAS IDEMPOTENTAIS IR NULPOTENTAIS ...	16
1.9. MATRICOS SUVEDIMAS Į TRIKAMPĘ MATRICĄ	18
1.10. PARAMETRŲ SKAIČIAVIMO ATSKIRI ATVEJAI	19
1.11. TRIKAMPĖS MATRICOS ALGEBRINĖS IŠRAIŠKOS	31
2. TIRIAMOJI DALIS IR REZULTATAI.....	35
2.1. TREČIOS EILĖS MATRICOS TRANSFORMACIJŲ PAVYZDŽIAI	35
2.2. TRIJŲ FUNKCIJŲ ALGEBRINĖ KORELIACIJA	37
2.3. ELEKTROKARDIOGRAMOS PARAMETRŲ TYRIMAS.....	39
2.4. PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI	43
IŠVADOS	50
PADĖKOS	51
LITERATŪRA	52
1 priedas. IŠ EKG DUOMENŲ GAUTŲ STRUKTŪRINIŲ PARAMETRŲ STANDARTINIŲ NUOKRYPIŲ GRAFIKAI	53
2 priedas. PROGRAMOS KODAS	59

LENTELIŲ SĄRAŠAS

2.1 lentelė. Funkcijų parametrų reikšmės	37
2.2 lentelė. EKG parametrų fiziologinės ribos	39

PAVEIKSLŲ SĄRAŠAS

1.1 pav. Pagrindiniai EKG parametrai.....	12
2.1 pav. Sinchroninių eilučių grafikas	38
2.2 pav. Tikrinių reikšmių grafikas.....	38
2.3 pav. Parametrų α^* , β^* , γ^* grafikas.....	38
2.4 pav. Parametrų α , β , γ grafikas.....	39
2.5 pav. Parametrų α^* , β^* , γ^* standartinių nuokrypių vidurkiai.....	40
2.5 pav. Parametrų α , β , γ standartinių nuokrypių vidurkiai.....	40
2.6 pav. Pirmosios grupės α^* parametro standartiniai nuokrypiai.....	41
2.7 pav. Antrosios grupės α^* parametro standartiniai nuokrypiai	41
2.8 pav. Parametro β^* standartiniai nuokrypiai	41
2.9 pav. Parametro γ^* standartiniai nuokrypiai	42
2.10 pav. Parametro α standartiniai nuokrypiai	42
2.11 pav. Parametro β standartiniai nuokrypiai	42
2.12 pav. Parametro γ standartiniai nuokrypiai.....	42
2.13 pav. Pirmasis programos langas.....	43
2.14 pav. Antrasis programos langas	44
2.15 pav. Trečiasis programos langas	45
2.16 pav. Ketvirtasis programos langas	45
2.17 pav. Penktasis programos langas	46
2.18 pav. Šeštasis programos langas.....	46
2.19 pav. Septintasis programos langas	47
2.20 pav. Aštuntasis programos langas.....	48
2.21 pav. Devintasis programos langas	48
2.22 pav. Dešimtas programos langas.....	48
2.23 pav. Vienuoliktasis programos langas	49

IVADAS

Žmogaus fiziologija yra vienas aktualiausių bei sudėtingiausių kompleksinių sistemų analizės uždavinių. Viena svarbiausių problemų medicinos srityje yra širdies ir kraujagyslių ligų diagnostika. Pagrindinis kardiologinis tyrimas yra atliekamas elektrokardiogramos (EKG) pagalba, tačiau žinomi EKG analizės metodai ne visada atskleidžia pakankamą klinikinės informacijos kiekį. Todėl yra reikalinga naujų metodų paieška, norint atlikti elektrokardiografinių signalų analizę, kuri padėtų atskleisti kokybiškai naują informaciją signaluose.

Vienas iš novatoriškų matematinių metodų, elektrokardiografinių signalų dinaminių sąsajų analizei atlikti, gali būti algebrinė-struktūrinė matricų analizė. Pagrindinės tokios analizės sąvokos yra idempotentai (matricos, kurios keliamos laipsniu nekinta) ir nulpotentai (matricos, kurios keliant laipsniu virsta nulinėmis). Naudojantis tokiomis struktūrinės matricų analizės sąvokomis ir atliekant transformacijas, galima tirti signalų dinamines sąsajas. Tokia ideologija, naudojant antros eilės matricas, taikoma tokiose srityse kaip medicina [3, 12] ir dinaminių sistemų tyrimai [13, 14].

Darbo tikslas – pritaikyti algebrinės-struktūrinės matricų analizės algoritmus EKG parametrų dinaminių sąsajų analizei.

Darbo uždaviniai:

1. Atlikti kompleksinių sistemų teorijos analizę, naudojamą EKG signalų tyrimui;
2. Aprašyti tiriamus elektrokardiografinius signalus bei jų reikšmių kointegraciją į trečios eilės Lagranžo skirtumų matricų seką;
3. Pateikti algebrinės-struktūrinės matricų analizės metodus, skirtus parametrų dinaminių sąsajų tyrimui;
4. Rasti bei įrodyti trečios eilės matricų struktūrinių koeficientų skaičiavimo atskirų atvejų išraiškas;
5. Išskirti algebrinius-struktūrinius parametrus iš trečios eilės kvadratinių matricų;
6. Pritaikyti algebrinius-struktūrinius parametrus sinchroninių eilučių dinaminių sąsajų tyrimui;
7. Ištirti elektrokardiogramų signalų tarpusavio sąsajas, naudojant algebrinės-struktūrinės matricų teorijos metodus;
8. Sukurti programinę įrangą naudojamiems algebrinės-struktūrinės analizės metodams bei elektrokardiografinių signalų tyrimui.

Bendrojoje darbo dalyje aprašoma kompleksinių sistemų bei širdies, kaip kompleksinės adaptyvios sistemos dalies, analizė. Pateikiama teorija apie elektrokardiogramą bei iš jos gaunamus parametrus. Aprašoma EKG parametrų tyrimo, naudojant algebrinę-struktūrinę matricų analizę, ideologija. Pateikiamos algebrinės-struktūrinės matricų teorijos pagrindinės sąvokos bei

transformacijų, atliekamų naudojant idempotentų ir nulpotentų sistemas, aprašas. Aprašomas algebrinių-struktūrinių parametrų gavimas, atskirai įrodant atskirų skaičiavimo atvejų išraiškas.

Tiriamajoje dalyje pateikiamas trečios eilės matricos algebrinių-struktūrinių parametrų išskyrimo pavyzdys, atliekant bendrojoje dalyje aprašytas transformacijas. Norint įvertinti struktūrinių parametrų naudą signalų dinaminių sąsajų analizei, pateikiamas iš trijų funkcijų gautų sinchroninių eilučių tyrimas. Atliekant eilučių analizę, iš jų narių sudaroma Lagranžo skirtumų matricų seka, tada kiekvienam sekos nariui atliekamos algebrinės matricų transformacijos, išskiriant struktūrinius parametrus. Tokia pati metodika taikoma tirti trims pasirinktiems EKG parametrams: RR, DQRS, DJT. Duomenys gauti iš dešimties asmenų kardiogramų, paimtų skirtingo fizinio būvio metu. Norint įvertinti EKG parametrų dinaminių sąsajų tyrimo rezultatus, žvelgiama ne į struktūrinių parametrų reikšmes, bet į gautų reikšmių vidutinį kvadratinį nuokrypį, gautą kiekvienos minutės laikotarpiu. Tyrimas atliktas naudojantis sukurta bei aprašyta programine įranga, kuri atlieka visus aprašytus skaičiavimus matricoms, generuojamoms sinchroninėms eilutėms bei EKG duomenims, be to, gauti rezultatai pateikiami grafiškai.

1. BENDROJI DALIS

1.1. KOMPLEKSNĖS SISTEMOS

Kompleksinė sistema – tai tokia sistema, kurią sudaro didelis skaičius sudedamųjų dalių, kurios tarpusavyje sąveikauja netiesiškai ir tarp atskirų dalių egzistuoja hierarchija. Tokias sąveikas yra sudėtinga matematiškai aprašyti, prognozuoti ar valdyti [5]. Galima išskirti tokias kompleksinių sistemų pagrindines savybes [3]:

1. Sudėtinga apibrėžti sistemos ribas – sunku atskirti pačią sistemą ir jos aplinką, todėl tai iš anksto turi apibrėžti tyrėjas;
2. Kompleksinės sistemos turi fraktalinę struktūrą. Ji gali būti nusakoma keliais lygiais;
3. Kompleksinės sistemos yra dinaminės ir praeities būseną gali įtakoti dabartį bei ateitį;
4. Kompleksinė sistema pasižymi kooperacija sistemos viduje;
5. Sistema pasižymi konkurencija sistemos išorėje;
6. Kompleksinių sistemų dedamosios dalys tarpusavyje sąveikauja netiesiškai;
7. Sistemos geba staigiai pakeisti elgseną.

Galima išskirti du pagrindinius kompleksinių sistemų tipus: chaotinės sistemos ir kompleksinės adaptyvios sistemos (KAS). Kompleksinės adaptyvios sistemos susideda iš didelio kiekio tarpusavyje susijusių elementų, kurie geba prisitaikyti bei keistis. KAS elgesys charakterizuojamas kaip jos sudedamųjų dalių netiesinės sąveikos (erdvėje ir laike) rezultatas, skirtinguose sistemos organizacijos lygiuose [5], [10].

1.2. ŠIRDIS KAIP KOMPLEKSNĖS ADAPTYVIOS SISTEMOS DALIS

Širdis yra tuščiaaviduris raumeninis organas, pumpuojantis kraują ir visą organizmą aprūpinantis deguonimi bei maisto medžiagomis. Širdis susideda iš keturių ertmių – dviejų skilvelių ir dviejų prieširdžių. Kairįjį ir dešinįjį skilvelius skiria tvirta raumeninė pertvara, o prieširdžius skiria plonesnė pertvara. Kairįjį skilvelį nuo kairiojo prieširdžio skiria dviburis vožtuvas, o dešinįjį skilvelį nuo dešiniojo prieširdžio skiria triburis vožtuvas. Iš kairiojo skilvelio išeina aorta (stambiausia organizmo arterija), iš dešiniojo - plaučių arterija. Nuo skilvelių šias kraujagysles skiria aortos ir plaučių arterijos vožtuvai. Širdis yra pagrindinė širdies ir kraujagyslių sistemos dalis [15].

XVI a. viduryje Vezalijus pastebėjo, kad žmogaus organizme yra kelios išskirtinėmis savybėmis pasižyminčios sistemos. Tai tokios sistemos, kurios savo struktūra apima visą žmogaus organizmo veiklą. Jos vadinamos holistinėmis sistemomis [11]:

1. Griaucių – raumenų sistema;
2. Širdies ir kraujagyslių sistema (aprūpinančioji sistema);

3. Reguliacinė sistema.

Paminėta ir ketvirtoji sistema – kvėpavimo sistema, kuri absorbuoja deguonį ir išskiria anglies dvideginį. Šią sistemą kartu su širdies ir kraujagyslių sistema galima apjungti į vieną aprūpinančiąją sistemą.

Žmogaus organizmui nuolat taikantis prie kintančios aplinkos, reaguoja visos trys holistinės sistemos kartu. Tačiau vertinant žmogaus kaip kompleksinės sistemos savybes, širdies ir kraujagyslių sistemą galime vadinti esmine struktūros dedamąja, sąlygojančia visą organizmo funkcionalumą daugeliu lygių [7].

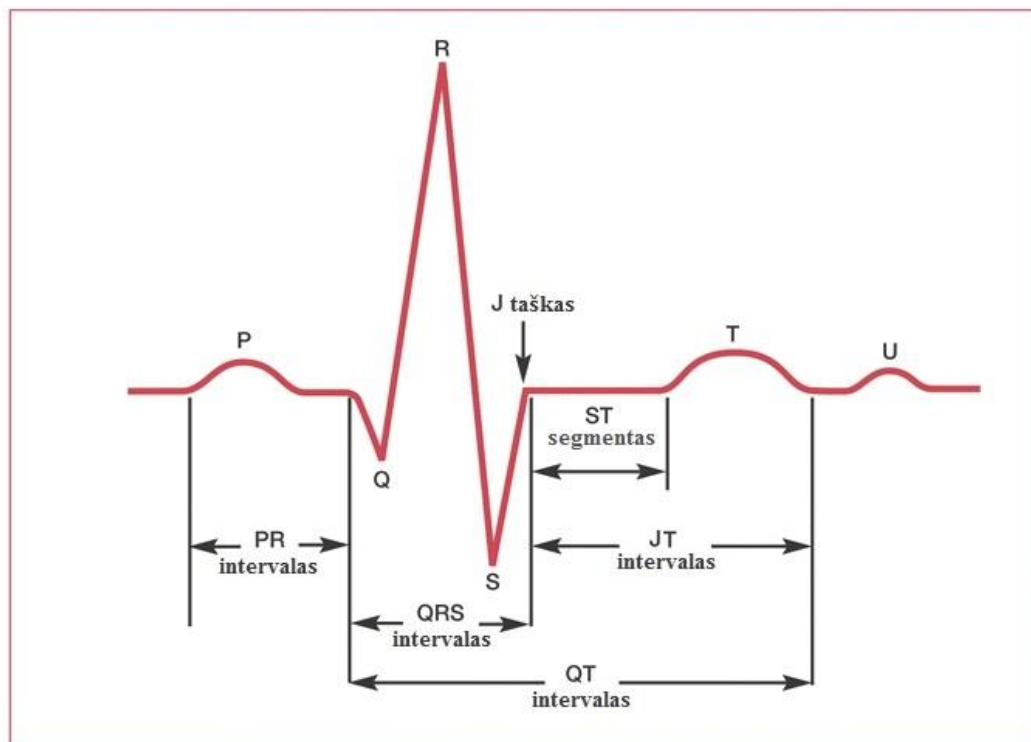
Kompleksinės adaptyvios sistemos požiūriu nagrinėjant širdies ir kraujagyslių sistemą, galime ją apibūdinti kaip sistemą, sąveikaujančią su visomis organizmo sistemomis ir savo funkcionalumą išreiškiančią elektriniais procesais – elektrokardiograma, kurią ir aprašysime sekančiame skyrelyje.

1.3. ELEKTROKARDIOGRAMA

Elektrokardiografija yra elektrofiziologijos metodas, kurio metu specialiu prietaisu – elektrokardiografu užrašoma kreivė, vaizduojanti širdies ląstelių sukurto suminio elektros lauko potencialo kitimą. Pagal elektrokardiografijos metu gautos kreivės ypatumus sprendžiama apie širdies veiklą [9].

Elektrokardiograma (EKG) yra bioelektrinis signalas, gaunamas registruojant širdyje vykstančius procesus ir / arba elektrinius reiškinius, sukurtus tam tikro dirbtinio išorinio poveikio. Ji gaunama elektrokardiografijos metu ir yra sudaryta iš matavimų, kurie, priklausomai nuo aparatūros, atliekami nuo kelių šimtų iki kelių tūkstančių kartų per sekundę. Elektriniai signalai plinta per skystą ir laidžią vidinę organizmo terpę iki registruojančių elektrodų, kurie dedami sutartiniuose kūno paviršiaus taškuose ir yra laidais sujungti su aparatu, registruojančiu elektros impulsus, kylančius kiekvieno širdies susitraukimo metu. EKG fiksuoja širdies bioelektrinius potencialus ir kartu su klinikiniais duomenimis leidžia diagnozuoti širdies veiklos pokyčius, stebėti patologinių procesų eigą bei prognozuoti jų baigtis. Elektrokardiograma yra vienas pagrindinių ir dažniausiai naudojamų medicininių tyrimų metodų [9].

Pagrindiniai EKG parametrai yra žymimi raidėmis:



1.1 pav. Pagrindiniai EKG parametrai. Adaptuota iš [1].

Pavaizduotų EKG parametrų fiziologinė prasmė [8], [3]:

1. PR intervalas – matuojamas nuo P bangos pradžios iki QRS komplekso pradžios. PR intervalas vaizduoja laiką nuo prieširdžių depoliarizacijos pradžios iki skilvelių depoliarizacijos pradžios.
2. QRS kompleksas - vaizduoja greitą kairiojo ir dešiniojo skilvelio depoliarizaciją. QRS komplekso trukmė DQRS (matuojamas ms) apibūdina širdies vidinę reguliacinę sistemą.
3. QT intervalas – matuojamas nuo QRS komplekso pradžios iki T bangos pabaigos. QT intervalas pavaizduoja skilvelių depoliarizaciją ir repoliarizaciją t.y. elektrinę sistolę. Kuo dažnesnis širdies ritmas, tuo trumpesnė skilvelių elektrinė sistolė.
4. ST segmentas – jungiasi su QRS kompleksu ir T banga. ST segmentas apibūdina laikotarpį, per kurį skilveliai depoliarizuojasi.
5. JT intervalas - matuojamas nuo taško J iki T bangos pabaigos. Šio intervalo pokyčiams įtakos turi reguliacinė nervų sistema. JT intervalo trukmė DJT (matuojama ms) apibūdina skilvelių repoliarizacijos trukmę.
6. J taškas – šiame taške baigiasi QRS kompleksas ir prasideda ST segmentas. J taškas naudojamas išmatuoti ST pakilimo arba nusileidimo laipsnį.
7. RR intervalas, matuojamas ms, - intervalas tarp dviejų gretimų R bangų – laiko tarpas tarp dviejų širdies susitraukimų. RR intervalu gali būti apibūdinta viso organizmo būseną. Normalus ramybės būsenos širdies ritmas yra tarp 60 ir 100 dūžių per minutę.

Šiame darbe bus nagrinėjami trys parametrai – RR, DQRS ir DJT. Šių trijų parametru sąveikos apibūdina vidinius širdies reguliacinius procesus.

1.4. LAGRANŽO SKIRTUMŲ MATRICA

Darbe tiriamų elektrokardiogramos parametru dinaminėms sąsajoms vertinti pritaikysime matricų analizės metodus. Šiam tikslui naudosime Lagranžo skirtumų matricų seką. Tokias matricas sudarysime iš trijų sinchronizuotų laiko eilučių $x = (x_j, j = 0, 1, 2, \dots)$, $y = (y_j, j = 0, 1, 2, \dots)$ ir $z = (z_j, j = 0, 1, 2, \dots)$ ir gausime laiko eilučių tarpusavio ryšį apibūdinančių trečios eilės matricų seką:

$$W_k = \begin{pmatrix} x_k & x_{k+1} - y_{k+1} & z_{k+1} - x_{k+1} \\ x_{k-1} - y_{k-1} & y_k & y_{k+1} - z_{k+1} \\ z_{k-1} - x_{k-1} & y_{k-1} - z_{k-1} & z_k \end{pmatrix}, k = 1, 2, \dots \quad (1.1)$$

Iš gautos Lagranžo matricų sekos, algebrinės matricų analizės metodais išskirsime struktūrinius–algebrinius parametrus, pagal kuriuos vertinsime parametru dinamines sąsajas. Šiam tikslui pirmiausiai apibrėšime pagrindines algebrinių matricų struktūrų sąvokas, reikalingas tokių parametru radimui.

1.5. MATRICOS EILUČIŲ IR STULPELIŲ STRUKTŪROS

Darbe nagrinėjamos trečios eilės matricos dažnai interpretuojamos kaip matricų-stulpelių ir matricų-eilučių sandaugos. Dėl šios priežasties šiame skyrelyje aprašysime pagrindines tokių struktūrų sąvokas bei savybes.

Apibrėžimas 1.1. Trečios eilės matrica-stulpelis turi tokį pavidalą:

$$P = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, x_k \in \mathbb{C}, k = 1, 2, 3. \quad (1.2)$$

Apibrėžimas 1.2. Trečios eilės matrica-eilutė turi tokį pavidalą:

$$R = (y_1 \ y_2 \ y_3), y_k \in \mathbb{C}, k = 1, 2, 3. \quad (1.3)$$

Daugindami eilutę iš stulpelio gausime skaliarą, o stulpelį iš eilutės – trečios eilės kvadratinę matricą. Operacijoms su matricomis-eilutėmis ir matricomis-stulpeliais galioja šios elementariosios savybės:

1. $\alpha(PR) = (\alpha P)R = P(\alpha R)$, čia $\alpha \in \mathbb{C}$ - konstanta.
2. Jeigu turime dvi matricas-stulpelius P_1, P_2 ir dvi matricas-eilutes R_1, R_2 , tai galioja tokia lygybė:

$$P_1 R_1 + P_2 R_2 = (P_1 \ P_2) \begin{pmatrix} R_1 \\ R_2 \end{pmatrix} \quad (1.4)$$

3. Distributyvumas eilutės atžvilgiu:

$$P_1 R + P_2 R = (P_1 + P_2) R \quad (1.5)$$

4. Distributyvumas stulpelio atžvilgiu:

$$PR_1 + PR_2 = P(R_1 + R_2) \quad (1.6)$$

5. Jeigu

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} x_1 y_1 & x_1 y_2 & x_1 y_3 \\ x_2 y_1 & x_2 y_2 & x_2 y_3 \\ x_3 y_1 & x_3 y_2 & x_3 y_3 \end{pmatrix}, a_{11} \neq 0$$

Tada

$$x_1 := a, x_2 = \frac{a_{21}}{a_{11}} a, x_3 = \frac{a_{31}}{a_{11}} a, \quad (1.7)$$

$$y_1 = \frac{a_{11}}{a}, y_2 = \frac{a_{12}}{a}, y_3 = \frac{a_{13}}{a}.$$

Matricų eilučių ir stulpelių rangams bei pėdsakams galioja tokios pagrindinės savybės:

1. Jeigu $(|x_1| + |x_2| + |x_3|) \cdot (|y_1| + |y_2| + |y_3|) \neq 0$, tada:

$$\text{rang}P = \text{rang}R = \text{rang}PR = 1 \quad (1.8)$$

$$\text{Tr}PR = RP = x_1 y_1 + x_2 y_2 + x_3 y_3 \quad (1.9)$$

2. Jeigu $(|x_1| + |x_2| + |x_3|) \cdot (|y_1| + |y_2| + |y_3|) = 0$, tada:

$$\text{rang}P \cdot \text{rang}R = 0 \quad (1.10)$$

$$\text{Tr}PR = RP = 0 \quad (1.11)$$

1.6. PAGRINDINĖS ALGEBRINIŲ MATRICŲ STRUKTŪROS

Šiame skyrelyje apibrėšime pagrindines matricų struktūras, tokias kaip elementarieji idempotentai ir nulpotentai. Aprašysime pagrindines tokių struktūrų savybes bei ortogonalumo sąlygas.

Apibrėžimas 1.3. Elementariuoju idempotentu vadiname trečios eilės matricą D , jeigu egzistuoja tokia matrica-stulpelis P ir matrica-eilutė R , kad $D = PR$ ir $\text{rang}D = \text{Tr}D = 1$.

Apibrėžimas 1.4. Elementariuoju nulpotentu vadiname trečios eilės matricą N , jeigu egzistuoja tokia matrica-stulpelis P ir matrica-eilutė R , kad $N = PR$ ir $\text{rang}N = 1$, $\text{Tr}N = 0$.

Teorema 1.1. Tegul $D \in \mathbb{C}^{3 \times 3}$ ir $N \in \mathbb{C}^{3 \times 3}$ atitinkamai yra elementarusis idempotentas ir elementarusis nulpotentas. Tada $D^2 = D$ ir $N^2 = O$, kur O – trečios eilės nulinė matrica.

Irodymas. Iš (1.8) lygybės bei 1.3 apibrėžimo turime:

$$D^2 = (P_1 R_1)^2 = P_1 R_1 P_1 R_1 = P_1 (\text{Tr}P_1 R_1) R_1 = P_1 \text{Tr}D R_1 = P_1 \cdot 1 \cdot R_1 = D$$

Iš (1.8) lygybės bei 1.4 apibrėžimo turime:

$$N^2 = (P_2 R_2)^2 = P_2 R_2 P_2 R_2 = P_2 (\text{Tr}P_2 R_2) R_2 = P_2 \text{Tr}N R_2 = P_2 \cdot 0 \cdot R_2 = 0. \blacksquare$$

Pastaba. Tegul duoti bet kokie du kompleksinių skaičių trejetai (x_1, x_2, x_3) ir (y_1, y_2, y_3) . Iš jų galima sudaryti matricą $Z = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} (y_1 \ y_2 \ y_3)$ bei apskaičiuoti $\sigma := x_1 y_1 + x_2 y_2 + x_3 y_3$. Jeigu $\sigma \neq 0$, tai $\frac{1}{\sigma} Z$ yra elementarusis idempotentas, o jeigu $\sigma = 0$, tai γZ – elementarusis nulpotentas, kai $\gamma \in \mathbb{C}$.

Apibrėžimas 1.5. Du elementarieji idempotentai D_1 ir D_2 vadinami ortogonaliais, jeigu $D_1 D_2 = D_2 D_1 = 0$.

Išvada. Jeigu D_1 ir D_2 yra ortogonalūs idempotentai, tai $D_1 + D_2$ – taip pat idempotentas, t.y. $(D_1 + D_2)^2 = D_1 + D_2$. Be to, $\text{rang}(D_1 + D_2) = \text{Tr}(D_1 + D_2) = 2$. Pastebime, kad šis idempotentas nėra elementarusis.

Teorema 1.2. Tegul duoti elementarieji idempotentai D_k , $k = 1, 2, 3$, kurie išreiškiami lygybėmis:

$$D_k = \begin{pmatrix} x_{1k} \\ x_{2k} \\ x_{3k} \end{pmatrix} (y_{k1} \ y_{k2} \ y_{k3}) \quad (1.12)$$

Šie idempotentai ortogonalūs tarpusavyje tada ir tik tada, kai

$$y_{l1} x_{1k} + y_{l2} x_{2k} + y_{l3} x_{3k} = \delta_{lk} \quad (1.13)$$

čia $k, l = 1, 2, 3$ ir $\delta_{lk} = \begin{cases} 0, & l \neq k \\ 1, & l = k \end{cases}$ – Kronekerio delta.

Irodymas.

Būtinumas. Elementarieji idempotentai D_k , $k = 1, 2, 3$ – ortogonalūs tarpusavyje. Kai $k = l$, tai $y_{l1} x_{1k} + y_{l2} x_{2k} + y_{l3} x_{3k} = \text{Tr} D_k = 1$. Kai $k \neq l$ iš 1.5 apibrėžimo gauname, kad $y_{l1} x_{1k} + y_{l2} x_{2k} + y_{l3} x_{3k} = 0$.

Pakankamumas. $y_{l1} x_{1k} + y_{l2} x_{2k} + y_{l3} x_{3k} = \delta_{lk}$, čia $k, l = 1, 2, 3$. Tenkinama idempotentų ortogonalumo sąlyga, vadinasi, D_k , $k = 1, 2, 3$ – ortogonalūs. ■

Apibrėžimas 1.6. Elementarusis nulpotentas N_{12} vadinamas suderintu su ortogonaliais elementariaisiais idempotentais D_1 ir D_2 jeigu $(D_1 + D_2 + N_{12})^2 = D_1 + D_2 + \alpha N_{12}$, $\alpha \in \mathbb{C}$.

1.7. IDEMPOTENTŲ IR NULPOTENTŲ SISTEMOS

Šiame skyrelyje apibrėšime pagrindines sąvokas apie idempotentų ir nulpotentų sistemas. Parodysime, kad bet kokia neišsigimusi matrica $X \in \mathbb{C}^{3 \times 3}$, kartu su savo atvirkštine matrica Y gali sugeneruoti idempotentų ir nulpotentų sistemą [2].

Apibrėžimas 1.7. Idempotentu vadiname tokią trečios eilės matricą D , kurią padauginus iš jos pačios, gaunama pradinė matrica:

$$D^2 = D \quad (1.14)$$

Idempotentą visada galima diagonalizuoti, o jo tikrinės reikšmės yra lygios nuliui arba vienam [6]. Tokios matricos pėdsakas yra lygus jos rangui, kuris visada yra sveikas skaičius.

Apibrėžimas 1.8. Nulpotentu vadiname tokią trečios eilės matricą N , kuri tenkina lygybę:

$$N^k = O \quad (1.15)$$

Čia O – nulinė trečios eilės matricą, k – natūralusis skaičius. Mažiausia k reikšmė, tenkinanti (1.15) sąlygą, vadinama matricos N laipsniu.

Apibrėžimas 1.9. Generuojančių matricų pora vadinama matricų pora (X, Y) , kur $X \in \mathbb{C}^{3 \times 3}$ – neišsigimusi matrica, o $Y = X^{-1}$.

Įvedame pažymėjimus X stulpeliams ir Y eilutėms:

$$X = \begin{pmatrix} x_{11} & x_{12} & x_{13} \\ x_{21} & x_{22} & x_{23} \\ x_{31} & x_{32} & x_{33} \end{pmatrix} = \left(\begin{bmatrix} x_{11} \\ x_{21} \\ x_{31} \end{bmatrix} \quad \begin{bmatrix} x_{12} \\ x_{22} \\ x_{32} \end{bmatrix} \quad \begin{bmatrix} x_{13} \\ x_{23} \\ x_{33} \end{bmatrix} \right) = (X_1 \quad X_2 \quad X_3) \quad (1.16)$$

$$Y = \begin{pmatrix} y_{11} & y_{12} & y_{13} \\ y_{21} & y_{22} & y_{23} \\ y_{31} & y_{32} & y_{33} \end{pmatrix} = \begin{pmatrix} [y_{11} \quad y_{12} \quad y_{13}] \\ [y_{21} \quad y_{22} \quad y_{23}] \\ [y_{31} \quad y_{32} \quad y_{33}] \end{pmatrix} = \begin{pmatrix} Y_1 \\ Y_2 \\ Y_3 \end{pmatrix} \quad (1.17)$$

Atsižvelgus į 1.6 skyrelyje pateiktą teoriją ir į 1.9 apibrėžimą, gauname ortogonalius elementariusius idempotentus:

$$D_k := X_k Y_k, \quad k = 1, 2, 3 \quad (1.18)$$

Analogiškai gauname elementariusius nulpotentus, suderintus su idempotentais D_k ir D_l :

$$N_{kl} := X_k Y_l, \quad k, l = 1, 2, 3, \quad k \neq l \quad (1.19)$$

Apibrėžimas 1.10. Idempotentų ir nulpotentų aibę, kuri sugeneruota iš generuojančiųjų matricų poros (X, Y) pagal (1.18) ir (1.19) lygybes, vadiname idempotentų ir nulpotentų sistema (INS) ir žymime

$$\Psi_X = \{X_k Y_l | l = 1, 2, 3\} = \{D_1, D_2, D_3, N_{12}, N_{13}, N_{21}, N_{23}, N_{31}, N_{32}\}$$

Generuojančių matricų porą (X, Y) atitinkančios INS elementai turi tokias savybes:

1. $(X_k Y_l)(X_m Y_n) = \delta_{lm} X_k Y_n$, $k, l, m, n \in \{1, 2, 3\}$;
2. $D_1 + D_2 + D_3 = I$, kur I – vienetinė matrica.

Iš pateiktų sąvokų taip pat galime pastebėti, jog turint idempotentų ir nulpotentų sistemą yra žinoma, kad egzistuoja tokia matricų pora (X, Y) , kuri ją sugeneruoja.

1.8. MATRICOS IŠDĖSTYMAS IDEMPOTENTAIS IR NULPOTENTAIS

Šiame skyrelyje nagrinėsime trečios eilės matricos $A \in \mathbb{C}^{3 \times 3}$ išdėstymą idempotentų ir nulpotentų tiesiniu dariniu. Matricos išdėstymas priklauso nuo jos tikrinių reikšmių $\lambda_1, \lambda_2, \lambda_3$.

Turėdami 1.10 apibrėžime aprašytą INS, matricą A , priklausomai nuo jos tikrinių reikšmių, galime išreikšti tokiomis lygybėmis [2]:

1. Jeigu visos matricos tikrinės reikšmės skirtingos, t.y. $\lambda_1 \neq \lambda_2 \neq \lambda_3$:

$$A = \lambda_1 D_1 + \lambda_2 D_2 + \lambda_3 D_3 \quad (1.20)$$

2. Jeigu dvi tikrinės reikšmės lygios, o trečia skirtinga, t.y. $\lambda_0 := \lambda_1 = \lambda_2 \neq \lambda_3$:

$$A = \lambda_0(D_1 + D_2) + \alpha N_{12} + \lambda_3 D_3, \quad \alpha \in \mathbb{C} \quad (1.21)$$

$$A = \lambda_0(D_1 + D_2) + \beta N_{21} + \lambda_3 D_3, \quad \beta \in \mathbb{C} \quad (1.22)$$

3. Jeigu matrica A turi tris vienodas tikrines reikšmes, t.y. $\lambda_0 := \lambda_1 = \lambda_2 = \lambda_3$:

$$A = \lambda_0 I + \alpha N_{12} + \beta N_{13} + \gamma N_{23}, \quad \alpha, \beta, \gamma \in \mathbb{C} \quad (1.23)$$

$$A = \lambda_0 I + \hat{\alpha} N_{21} + \hat{\beta} N_{31} + \hat{\gamma} N_{32}, \quad \hat{\alpha}, \hat{\beta}, \hat{\gamma} \in \mathbb{C} \quad (1.24)$$

Idempotentus ir nulpotentus matricai A rasime pasinaudojus Hamiltono-Keilio teorema, kuri teigia, kad matrica A yra savojo charakteristinio daugianario šaknis [4]. Tada atitinkamai pagal matricos tikrines reikšmes gauname:

1. Jeigu $\lambda_1 \neq \lambda_2 \neq \lambda_3$:

$$D_1 = \frac{1}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} (A - \lambda_2 I)(A - \lambda_3 I) \quad (1.25)$$

$$D_2 = \frac{1}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} (A - \lambda_1 I)(A - \lambda_3 I) \quad (1.26)$$

$$D_3 = \frac{1}{(\lambda_3 - \lambda_1)(\lambda_3 - \lambda_2)} (A - \lambda_1 I)(A - \lambda_2 I) \quad (1.27)$$

2. Jeigu $\lambda_0 := \lambda_1 = \lambda_2 \neq \lambda_3$:

$$D_1 + D_2 = \frac{(2(\lambda_0 + \lambda_3)I - A)(A - \lambda_3 I)}{(\lambda_0 - \lambda_3)^2} \quad (1.28)$$

$$\alpha N_{12} = \frac{(A - \lambda_3 I)(A - \lambda_0 I)}{\lambda_0 - \lambda_3} \quad (1.29)$$

$$D_3 = \frac{(A - \lambda_0 I)^2}{(\lambda_3 - \lambda_0)^2} \quad (1.30)$$

3. Jeigu $\lambda_0 := \lambda_1 = \lambda_2 = \lambda_3$:

$$\alpha N_{12} + \beta N_{13} + \gamma N_{23} = A - \lambda_0 I \quad (1.31)$$

$$\tilde{\gamma} N_{13} = (A - \lambda_0 I)^2, \quad \tilde{\gamma} \in \mathbb{C} \quad (1.32)$$

Apibrėžimas 1.11. Tegul turime generuojančiųjų matricų porą (X, Y) ir neišsigimusią matricą T . Tada INS, sugeneruojamos iš (X, Y) , Ψ_X transformacija $t(\Psi_X)$ vadiname INS, kurią sugeneruoja generuojančiųjų matricų pora (TX, YT^{-1}) .

Iš pateikto transformacijos apibrėžimo matome, kad $t(\Psi_X) = \Psi_{TX} = T\Psi_X T^{-1} = \{TX_k Y_l T^{-1} | k, l = 1, 2, 3\}$. Taip pat pastebime, kad jeigu matricą $A \in \mathbb{C}^{3 \times 3}$ atitinkanti INS yra Ψ_X , tai matricą TAT^{-1} atitinkanti INS bus $t(\Psi_X)$.

1.9. MATRICOS SUVEDIMAS Į TRIKAMPĘ MATRICĄ

Šiame skyrelyje aprašysime kvadratinės trečios eilės matricos suvedimą į trikampę. Šią transformaciją atliksime pagalbinių trikampių matricų pagalba. Apskaičiuojant tokias matricas naudosime koeficientus, kurių reikšmės, darbe atliekamame tyrime, bus vienas svarbiausių kriterijų.

Teorema 1.3. Tegul turime trečios eilės kvadratinę matricą A :

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \quad (1.33)$$

Tada egzistuoja matricų T ir T^{-1} pora, kai

$$T = \begin{pmatrix} 1 & 0 & 0 \\ \alpha^* & 1 & 0 \\ \frac{1}{2}\alpha^*\beta^* + \gamma^* & \beta^* & 1 \end{pmatrix}, T^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -\alpha^* & 1 & 0 \\ \frac{1}{2}\alpha^*\beta^* - \gamma^* & -\beta^* & 1 \end{pmatrix} \quad (1.34)$$

Su kuriomis teisingos tokios lygybės:

$$B := TAT^{-1} = \begin{pmatrix} \lambda_1 & a & c \\ 0 & \lambda_2 & b \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad (1.35)$$

$$A = T^{-1}BT \quad (1.36)$$

Kai $\lambda_1, \lambda_2, \lambda_3$ – matricos A tikrinės reikšmės, o matricų T ir T^{-1} koeficientai $\alpha^*, \beta^*, \gamma^*$ – specialiu būdu parinkti kompleksiniai skaičiai, tenkinantys tokias sąlygas:

$$\begin{cases} a = a_{12} - \beta^*a_{13} \\ b = a_{23} + \alpha^*a_{13} \\ c = a_{13} \end{cases} \quad (1.37)$$

$$\left\{ \begin{array}{l} a_{12}\alpha^* - a_{13}\left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right) = a_{11} - \lambda_1 \\ -a_{12}\alpha^* + a_{13}\alpha^*\beta^* + a_{23}\beta^* = a_{22} - \lambda_2 \\ -a_{23}\beta^* - a_{13}\left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right) = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2a_{12} + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)(a_{23} + a_{13}\alpha^*) = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) - (\beta^*)^2a_{23} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)(a_{12} - a_{13}\beta^*) = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*\left(\left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{12} + \beta^*a_{22} + a_{32}\right) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)\left(\left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{13} + \beta^*a_{23} + a_{33}\right) = 0 \end{array} \right. \quad (1.38)$$

$$a_{11} + a_{22} + a_{33} = \lambda_1 + \lambda_2 + \lambda_3 \quad (1.39)$$

Kai $\lambda_1, \lambda_2, \lambda_3$ yra matricos A tikrinės reikšmės.

Teorema 1.4. Koeficientus $\alpha^*, \beta^*, \gamma^*$ galime apskaičiuoti tokiu būdu:

$$\alpha^* = \frac{a_{11}a_{23} - a_{21}a_{13} - a_{23}\lambda_1}{a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1} \quad (1.40)$$

$$\beta^* = \frac{a_{33}a_{12} - a_{32}a_{13} - a_{12}\lambda_3}{a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3} \quad (1.41)$$

$$\gamma^* = \frac{1}{2a_{13}}(\lambda_3 - \lambda_1 - a_{33} + a_{11} - a_{12}\alpha^* - a_{23}\beta^*) \quad (1.42)$$

Kai $a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 \neq 0$, $a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 \neq 0$, $a_{13} \neq 0$.

Irodymas. Iš ketvirtosios ir penktosios (1.38) sistemos lygčių turime:

$$\frac{1}{2}\alpha^*\beta^* - \gamma^* = -\frac{a_{21} + (a_{11} - a_{22})\alpha^* - a_{12}(\alpha^*)^2}{a_{23} + a_{13}\alpha^*}$$

$$\frac{1}{2}\alpha^*\beta^* + \gamma^* = -\frac{a_{32} + (a_{22} - a_{33})\beta^* - a_{23}(\beta^*)^2}{a_{12} - a_{13}\beta^*}$$

Įstatę gautas išraiškas į pirmąją ir trečiąją (1.38) sistemos lygtis, gauname:

$$a_{12}\alpha^* + a_{13}\frac{a_{21} + (a_{11} - a_{22})\alpha^* - a_{12}(\alpha^*)^2}{a_{23} + a_{13}\alpha^*} = a_{11} - \lambda_1$$

$$-a_{23}\beta^* + a_{13}\frac{a_{32} + (a_{22} - a_{33})\beta^* - a_{23}(\beta^*)^2}{a_{12} - a_{13}\beta^*} = a_{33} - \lambda_3$$

Dabar, supaprastinę gautas išraiškas, turime:

$$a_{12}a_{23}\alpha^* + a_{13}a_{21} + (a_{11} - a_{22})a_{13}\alpha^* = (a_{11} - \lambda_1)(a_{23} + a_{13}\alpha^*)$$

$$-a_{23}a_{12}\beta^* + a_{13}a_{32} + (a_{22} - a_{33})a_{13}\beta^* = (a_{33} - \lambda_3)(a_{12} - a_{13}\beta^*)$$

Iš čia

$$\alpha^* = \frac{a_{11}a_{23} - a_{21}a_{13} - a_{23}\lambda_1}{a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1}, \beta^* = \frac{a_{33}a_{12} - a_{32}a_{13} - a_{12}\lambda_3}{a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3}$$

Dabar trečią (1.38) sistemos lygtį padauginę iš antrosios ir sudėję su antra lytimi turime:

$$-a_{12}\alpha^* - a_{23}\beta^* - 2\gamma^*a_{13} = a_{22} - \lambda_2 + 2a_{33} - 2\lambda_3$$

Iš pastarosios lygybės išreiškę γ^* ir panaudoję (1.39) išraišką, turime:

$$\gamma^* = \frac{1}{2a_{13}}(\lambda_3 - \lambda_1 - a_{33} + a_{11} - a_{12}\alpha^* - a_{23}\beta^*). \blacksquare$$

1.10. PARAMETRŲ SKAIČIAVIMO ATSKIRI ATVEJAI

Parametrai, nagrinėti 1.9 skyrelyje bendru atveju apskaičiuojami pagal (1.41), (1.42) ir (1.43) formules, tačiau pateiktu atveju turi būti tenkinamos šios trys sąlygos:

$$a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 \neq 0 \quad (1.43)$$

$$a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 \neq 0 \quad (1.44)$$

$$a_{13} \neq 0 \quad (1.45)$$

Šiame skyrelyje pateiksime parametrų skaičiavimo formules atskirais atvejais, kai viena ar daugiau iš šių sąlygų nėra tenkinamos (parametrų išraiškos bei jų įrodymai sudaryti autorės).

Pirmas atvejis. Jeigu pateiktos sąlygos (1.43) ir (1.44) išpildomos, tačiau $a_{13} = 0$, parametrai apskaičiuojami tokiu būdu:

$$\alpha^* = \frac{a_{11} - \lambda_1}{a_{12}} \quad (1.46)$$

$$\beta^* = \frac{\lambda_3 - a_{33}}{a_{23}} \quad (1.47)$$

$$\gamma^* = \frac{a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12}}{a_{23}} + \frac{\alpha^* \beta^*}{2} \quad (1.48)$$

Čia $a_{12} \neq 0, a_{23} \neq 0$.

Irodymas. Į (1.38) lygčių sistemą įstatę reikšmę $a_{13} = 0$, gauname naują lygčių sistemą, kuri atrodo taip:

$$\left\{ \begin{array}{l} a_{12} \alpha^* = a_{11} - \lambda_1 \\ -a_{12} \alpha^* + a_{23} \beta^* = a_{22} - \lambda_2 \\ -a_{23} \beta^* = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12} + \left(\frac{1}{2} \alpha^* \beta^* - \gamma^*\right) a_{23} = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) - (\beta^*)^2 a_{23} + \left(\frac{1}{2} \alpha^* \beta^* + \gamma^*\right) a_{12} = 0 \\ a_{31} + \left(\frac{1}{2} \alpha^* \beta^* + \gamma^*\right) a_{11} + \beta^* a_{21} - \alpha^* \left(\left(\frac{1}{2} \alpha^* \beta^* + \gamma^*\right) a_{12} + \beta^* a_{22} + a_{32}\right) \\ + \left(\frac{1}{2} \alpha^* \beta^* - \gamma^*\right) (\beta^* a_{23} + a_{33}) = 0 \end{array} \right.$$

Dabar iš pirmos ir trečios šios sistemos lygčių turime:

$$\alpha^* = \frac{a_{11} - \lambda_1}{a_{12}}; \beta^* = \frac{\lambda_3 - a_{33}}{a_{23}}$$

Iš sistemos ketvirtosios lygties galime išreikšti parametro γ^* reikšmę:

$$\gamma^* = \frac{a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12}}{a_{23}} + \frac{\alpha^* \beta^*}{2}$$

Kadangi šiuo atveju yra išpildytos (1.43) ir (1.44) sąlygos, o parametras $a_{13} = 0$, tai įstačius į sąlygas šią reikšmę, gauname:

$$\begin{cases} a_{12} a_{23} \neq 0 \\ -a_{12} a_{23} \neq 0 \end{cases}$$

Taigi matome, kad parametrai a_{12}, a_{23} lygūs nuliui būti negali. ■

Antras atvejis. Jeigu išpildomos (1.43) ir (1.45) sąlygos, bet netenkinama (1.44) sąlyga. Šiuo atveju α^* ir γ^* atitinkamai apskaičiuojami pagal (1.40) ir (1.42) formules, o parametras β^* apskaičiuojamas tokiu būdu:

$$\beta^* = \frac{a_{22} - \lambda_2 + a_{12} \alpha^*}{a_{13} \alpha^* + a_{23}} \quad (1.49)$$

Čia turi būti teisinga tokia nelygybė:

$$a_{13} \alpha^* + a_{23} \neq 0 \quad (1.50)$$

Irodymas. Kadangi sąlygos (1.43) ir (1.45) išpildytos, α^* bei γ^* parametų formulės įrodomos analogiškai, kaip 1.4 teoremoje. Parametrą β^* gauname iš (1.38) sistemos antrosios lygties, kuri atrodo taip:

$$-a_{12} \alpha^* + a_{13} \alpha^* \beta^* + a_{23} \beta^* = a_{22} - \lambda_2$$

Taigi $\beta^* = \frac{a_{22} - \lambda_2 + a_{12}\alpha^*}{a_{13}\alpha^* + a_{23}}$, kai išpildyta (1.50) sąlyga. ■

Jeigu $a_{13}\alpha^* + a_{23} = 0$, t.y. netenkinama (1.50) sąlyga, tada bet koks $\beta^* \in \mathbb{C}$ parametras tenkins (1.38) lygčių sistemą, kai parametrai α^* ir γ^* apskaičiuojami tokiu būdu:

$$\alpha^* = -\frac{a_{23}}{a_{13}} \quad (1.51)$$

$$\gamma^* = \frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^* a_{23}}{2a_{13}} \quad (1.52)$$

Irodymas. Iš (1.50) nelygybės matome, kad ji nebus teisinga tik tuo atveju kai

$$\alpha^* = -\frac{a_{23}}{a_{13}}$$

Įstatę gautą α^* reikšmę į (1.38) lygčių sistemos pirmas keturias lygtis, turime:

$$\begin{cases} -\frac{a_{12}a_{23}}{a_{13}} + \frac{1}{2}a_{23}\beta^* + a_{13}\gamma^* = a_{11} - \lambda_1 \\ \frac{a_{12}a_{23}}{a_{13}} = a_{22} - \lambda_2 \\ -\frac{1}{2}a_{23}\beta^* - a_{13}\gamma^* = a_{33} - \lambda_3 \\ a_{21} - \frac{a_{23}}{a_{13}}(a_{11} - a_{22}) - \left(\frac{a_{23}}{a_{13}}\right)^2 a_{12} = 0 \end{cases}$$

Kadangi šiuo atveju negalioja (1.44) sąlyga, tai $a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 = 0$. Iš čia galime išsireikšti λ_3 :

$$\lambda_3 = a_{22} - \frac{a_{12}a_{23}}{a_{13}}$$

Iš antrosios lygties išsireiškiame λ_2 :

$$\lambda_2 = a_{22} - \frac{a_{12}a_{23}}{a_{13}}$$

Matome, kad $\lambda_2 = \lambda_3$. Dabar, iš trečios lygties, gauname γ^* parametą:

$$\gamma^* = \frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^* a_{23}}{2a_{13}}$$

Įstatę gautą reikšmę į pirmąją lygtį, gauname:

$$-\frac{a_{12}a_{23}}{a_{13}} + \frac{1}{2}a_{23}\beta^* + \lambda_3 - a_{33} - \frac{1}{2}a_{23}\beta^* = a_{11} - \lambda_1$$

Supaprastinę šią lygtį turime:

$$\lambda_1 + \lambda_3 = a_{11} + a_{33} + \frac{a_{12}a_{23}}{a_{13}}$$

Prie iš abiejų pusių pridėję atitinkamai λ_2 ir jos išraišką, gauname:

$$\lambda_1 + \lambda_2 + \lambda_3 = a_{11} + a_{22} + a_{33}$$

Kaip žinome iš pradinės sąlygos (1.39), ši lygybė yra teisinga, taigi γ^* išraiška tenkina šią lygtį. Dabar, pertvarkę ketvirtąją lygtį ir padauginę ją iš $a_{13} \neq 0$, turime:

$$a_{21}a_{13} - a_{23}a_{11} + a_{23} \left(a_{22} - \frac{a_{12}a_{23}}{a_{13}} \right) = 0$$

Taigi matome, kad įstačius turimą λ_3 reikšmę, galime gauti naują jos išraišką:

$$\lambda_3 = a_{11} - \frac{a_{13}a_{21}}{a_{23}}$$

Pagal gautas α^* ir γ^* parametrų išraiškas gauname tokias lygybes:

$$\frac{1}{2}\alpha^*\beta^* + \gamma^* = \frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^*a_{23}}{a_{13}}$$

$$\frac{1}{2}\alpha^*\beta^* - \gamma^* = -\frac{\lambda_3 - a_{33}}{a_{13}}$$

Dabar su šių lygybių pagalba perrašysime penktąją ir šeštąją (1.38) lygčių sistemos lygtis. Penktoji lygtis atrodoys taip:

$$a_{32} + \beta^*(a_{22} - a_{33}) - (\beta^*)^2a_{23} + \left(\frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^*a_{23}}{a_{13}} \right) (a_{12} - a_{13}\beta^*) = 0$$

Padauginę šią lygtį iš $a_{13} \neq 0$ ir ją supaprastinę, gauname:

$$a_{32}a_{13} + a_{12}(\lambda_3 - a_{33}) + \beta^*(a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3) = 0$$

Kadangi šiuo atveju netenkinama (1.44) sąlyga, todėl daugiklis prie β^* yra lygus nuliui ir pastebime, kad nuo β^* parametro lygties teisingumas nepriklauso.

Šeštoji (1.38) lygčių sistemos lygtis dabar atrodoys taip:

$$a_{31} + \left(\frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^*a_{23}}{a_{13}} \right) a_{11} + \beta^*a_{21} + \frac{a_{23}}{a_{13}} \left(\left(\frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^*a_{23}}{a_{13}} \right) a_{12} + \beta^*a_{22} + a_{32} \right) + \left(-\frac{\lambda_3 - a_{33}}{a_{13}} \right) \left(\left(\frac{\lambda_3 - a_{33}}{a_{13}} - \frac{\beta^*a_{23}}{a_{13}} \right) a_{13} + \beta^*a_{23} + a_{33} \right) = 0$$

Supaprastinę šią lygtį, gauname:

$$a_{31} + \frac{a_{11}(\lambda_3 - a_{33})}{a_{13}} - \frac{\beta^*a_{23} \left(a_{11} - \frac{a_{13}a_{21}}{a_{23}} \right)}{a_{13}} + \frac{a_{23}a_{12}(\lambda_3 - a_{33}) + a_{23}a_{32}}{a_{13}} + \frac{\beta^*a_{23} \left(a_{22} - \frac{a_{12}a_{23}}{a_{13}} \right)}{a_{13}} + \frac{a_{33}\lambda_3 - \lambda_3^2}{a_{13}} = 0$$

Įvertinus turimas λ_3 reikšmių išraiškas, gauname:

$$a_{31} + \frac{a_{11}(\lambda_3 - a_{33})}{a_{13}} + \frac{a_{33}\lambda_3 - \lambda_3^2}{a_{13}} = 0$$

Matome, kad nuo parametro β^* šios lygties teisingumas taip pat nepriklauso. Taigi β^* gali būti bet koks skaičius, kai parametras γ^* skaičiuojamas pagal (1.52) išraišką. ■

Trečias atvejis. Jeigu išpildomos (1.44) ir (1.45) sąlygos, bet (1.43) – ne. Šiuo atveju β^* ir γ^* atitinkamai apskaičiuojami pagal (1.41) ir (1.42) formules, o parametras α^* apskaičiuojamas tokiu būdu:

$$\alpha^* = \frac{a_{22} - \lambda_2 - a_{23}\beta^*}{a_{13}\beta^* - a_{12}} \quad (1.53)$$

Čia turi būti teisinga tokia nelygybė:

$$a_{13}\beta^* - a_{12} \neq 0 \quad (1.54)$$

Irodymas. Kadangi sąlygos (1.44) ir (1.45) išpildytos, β^* bei γ^* parametru formules įrodomos analogiškai, kaip 1.4 teoremoje. Parametrą α^* gauname iš (1.38) sistemos antrosios lygties:

$$-a_{12}\alpha^* + a_{13}\alpha^*\beta^* + a_{23}\beta^* = a_{22} - \lambda_2$$

Taigi $\alpha^* = \frac{a_{22} - \lambda_2 - a_{23}\beta^*}{a_{13}\beta^* - a_{12}}$, kai išpildyta (1.54) sąlyga. ■

Jeigu $a_{13}\beta^* - a_{12} = 0$, t.y. netenkinama (1.54) sąlyga, tada bet koks $\alpha^* \in \mathbb{C}$ parametras tenkins (1.38) lygčių sistemą, kai parametrai β^* ir γ^* apskaičiuojami tokiu būdu:

$$\beta^* = \frac{a_{12}}{a_{13}} \quad (1.55)$$

$$\gamma^* = \frac{a_{11} - \lambda_1}{a_{13}} - \frac{\alpha^* a_{12}}{2a_{13}} \quad (1.56)$$

Irodymas. Iš (1.54) nelygybės matome, kad ji nebus teisinga tik tuo atveju kai

$$\beta^* = \frac{a_{12}}{a_{13}}$$

Įstatę gautą β^* reikšmę į (1.38) lygčių sistemos pirmas tris lygtis, turime:

$$\begin{cases} \frac{1}{2}a_{12}\alpha^* + a_{13}\gamma^* = a_{11} - \lambda_1 \\ \frac{a_{12}a_{23}}{a_{13}} = a_{22} - \lambda_2 \\ -\frac{a_{12}a_{23}}{a_{13}} - \frac{1}{2}a_{12}\alpha^* - a_{13}\gamma^* = a_{33} - \lambda_3 \end{cases}$$

Iš pirmos lygties, gauname γ^* parametru:

$$\gamma^* = \frac{a_{11} - \lambda_1}{a_{13}} - \frac{\alpha^* a_{12}}{2a_{13}}$$

Kadangi neišpildyta (1.43) sąlyga, tai $a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 = 0$. Iš čia galime išsireikšti λ_1 :

$$\lambda_1 = a_{22} - \frac{a_{12}a_{23}}{a_{13}}$$

Iš antrosios lygties išsireikiame λ_2 :

$$\lambda_2 = a_{22} - \frac{a_{12}a_{23}}{a_{13}}$$

Matome, kad $\lambda_2 = \lambda_1$. Dabar, iš pirmosios lygties gautą γ^* parametro reikšmę įstatome į trečiąją lygtį:

$$-\frac{a_{12}a_{23}}{a_{13}} - \frac{1}{2}a_{12}\alpha^* - a_{13}\left(\frac{a_{11} - \lambda_1}{a_{13}} - \frac{\alpha^*a_{12}}{2a_{13}}\right) = a_{33} - \lambda_3$$

Supaprastinę šią lygtį turime:

$$\lambda_1 + \lambda_3 = a_{11} + a_{33} + \frac{a_{12}a_{23}}{a_{13}}$$

Iš abiejų pusių atitinkamai pridėję λ_2 ir jos išraišką, gauname:

$$\lambda_1 + \lambda_2 + \lambda_3 = a_{11} + a_{22} + a_{33}$$

Kaip žinome iš pradinės sąlygos (1.39), ši lygybė yra teisinga, taigi γ^* išraiška tenkina šią lygtį. Įstačius β^* parametro išraišką į penktąją sistemos lygtį, ji atrodoys taip:

$$a_{32} + \frac{a_{12}}{a_{13}}(a_{22} - a_{33}) - \left(\frac{a_{12}}{a_{13}}\right)^2 a_{23} = 0$$

Pertvarkę penktąją lygtį ir padauginę ją iš $a_{13} \neq 0$, turime:

$$a_{32}a_{13} - a_{33}a_{12} + a_{12}\left(a_{22} - \frac{a_{12}a_{23}}{a_{13}}\right) = 0$$

Taigi matome, kad įstačius turimą λ_2 reikšmę, galime gauti naują jos išraišką:

$$\lambda_2 = a_{33} - \frac{a_{13}a_{32}}{a_{12}}$$

Pagal gautas β^* ir γ^* parametrų išraiškas gauname tokias lygybes:

$$\frac{1}{2}\alpha^*\beta^* + \gamma^* = \frac{a_{11} - \lambda_1}{a_{13}}$$

$$\frac{1}{2}\alpha^*\beta^* - \gamma^* = \frac{a_{12}\alpha^*}{a_{13}} - \frac{a_{11} - \lambda_1}{a_{13}}$$

Dabar su šių lygybių pagalba perrašysime ketvirtąją ir šeštąją (1.38) lygčių sistemos lygtis. Ketvirtoji lygtis atrodoys taip:

$$a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12} + \left(\frac{a_{12}\alpha^*}{a_{13}} - \frac{a_{11} - \lambda_1}{a_{13}}\right)(a_{23} + a_{13}\alpha^*) = 0$$

Padauginę šią lygtį iš $a_{13} \neq 0$ ir ją supaprastinę, gauname:

$$a_{21}a_{13} - a_{23}(a_{11} - \lambda_1) + \alpha^*(a_{23}a_{12} - a_{22}a_{13} - a_{13}\lambda_1) = 0$$

Kadangi šiuo atveju netenkinama (1.43) sąlyga, todėl daugiklis prie α^* yra lygus nuliui ir pastebime, kad nuo α^* parametro lygties teisingumas nepriklauso. Šeštoji (1.38) lygčių sistemos lygtis dabar atrodoys taip:

$$a_{31} + \frac{a_{11} - \lambda_1}{a_{13}}a_{11} + \frac{a_{12}}{a_{13}}a_{21} - \alpha^*\left(\frac{a_{11} - \lambda_1}{a_{13}}a_{12} + \frac{a_{12}}{a_{13}}a_{22} + a_{32}\right) + \left(\frac{a_{12}\alpha^*}{a_{13}} - \frac{a_{11} - \lambda_1}{a_{13}}\right)\left(\frac{a_{11} - \lambda_1}{a_{13}}a_{13} + \frac{a_{12}}{a_{13}}a_{23} + a_{33}\right) = 0$$

Supaprastinę šią lygtį, gauname:

$$a_{31} + \frac{a_{12}a_{21}}{a_{13}} + \frac{a_{12}^2 a_{23}}{a_{13}^2} - \alpha^* \frac{a_{12}}{a_{23}} \left(a_{22} - \frac{a_{12}a_{23}}{a_{13}} \right) + \alpha^* \frac{a_{12}}{a_{23}} \left(a_{33} - \frac{a_{13}a_{32}}{a_{12}} \right) + \frac{(a_{11} - \lambda_1)(\lambda_1 - a_{33})}{a_{13}} - \frac{a_{12}a_{23}(a_{11} - \lambda_1)}{a_{13}^2} = 0$$

Kadangi $\lambda_1 = \lambda_2 = a_{22} - \frac{a_{12}a_{23}}{a_{13}} = a_{33} - \frac{a_{13}a_{32}}{a_{12}}$, tai matome, kad daugiklis prie parametro α^* yra lygūs nuliui, todėl parametras α^* įtakos lygties teisingumui neturi ir gali būt bet koks skaičius, kai parametras γ^* skaičiuojamas pagal (1.56) išraišką. ■

Ketvirtas atvejis. Galimi trys atvejai, dėl kurių bus netenkinamos visos trys sąlygos (1.43), (1.44) ir (1.45):

1. Kai $a_{13} = 0, a_{12} = 0, a_{23} \neq 0$;
2. Kai $a_{13} = 0, a_{12} \neq 0, a_{23} = 0$;
3. Kai $a_{13} = 0, a_{12} = 0, a_{23} = 0$.

Irodymas. Visos trys sąlygos bus netenkinamos tada ir tik tada kai:

$$\begin{cases} a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 = 0 \\ a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 = 0 \\ a_{13} = 0 \end{cases}$$

Iš pateiktos lygčių sistemos trečiosios lygybės matome, kad a_{13} turi būti lygus nuliui. Įstatę šią reikšmę į pirmąsias dvi lygtis, gauname naują sistemą:

$$\begin{cases} a_{12}a_{23} = 0 \\ -a_{12}a_{23} = 0 \end{cases}$$

Iš gautos sistemos matome, kad egzistuoja trys atvejai, kada bus tenkinamos lygybės: kai vienas iš narių (a_{12} arba a_{23}) bus lygus nuliui, o kitas – ne, arba kai jie abu bus lygūs nuliui. ■

Išnagrinėsime kiekvieną iš pateiktų trijų atvejų:

1. Kai $a_{13} = 0, a_{12} = 0, a_{23} \neq 0$ parametrai gaunami tokiu būdu:

$$\alpha^* = \frac{a_{21}a_{33} - a_{23}a_{31} - a_{11}a_{21}}{a_{11}^2 - a_{11}a_{22} - a_{23}a_{32} + a_{22}a_{33} - a_{11}a_{33}} \quad (1.57)$$

$$\beta^* = \frac{a_{22} - \lambda_2}{a_{23}} \quad (1.58)$$

$$\gamma^* = \frac{a_{21} + (a_{11} - a_{22})\alpha^*}{a_{23}} + \frac{\alpha^*\beta^*}{2} \quad (1.59)$$

Čia $a_{11}^2 - a_{11}a_{22} - a_{23}a_{32} + a_{22}a_{33} - a_{11}a_{33} \neq 0$.

Irodymas. Įstatę $a_{13} = 0, a_{12} = 0$ reikšmes į (1.38) sistemos lygtis, gauname tokią sistemą:

$$\begin{cases} 0 = a_{11} - \lambda_1 \\ a_{23}\beta^* = a_{22} - \lambda_2 \\ -a_{23}\beta^* = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{23} = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) - (\beta^*)^2a_{23} = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)(\beta^*a_{23} + a_{33}) = 0 \end{cases}$$

Iš antrosios ir ketvirtosios lygčių, atitinkamai gauname parametrų β^* ir γ^* išraiškas:

$$\beta^* = \frac{a_{22} - \lambda_2}{a_{23}}; \gamma^* = \frac{a_{21} + (a_{11} - a_{22})\alpha^*}{a_{23}} + \frac{\alpha^*\beta^*}{2}$$

Įstatę gautą γ^* išraišką į sistemos šeštąją lygtį, gauname:

$$\begin{aligned} a_{31} + \left(\alpha^*\beta^* + \frac{a_{21} + (a_{11} - a_{22})\alpha^*}{a_{23}}\right)a_{11} + \beta^*a_{21} - \alpha^*\beta^*a_{22} - \alpha^*a_{32} \\ + \left(\frac{(a_{22} - a_{11})\alpha^* - a_{21}}{a_{23}}\right)(\beta^*a_{32} + a_{33}) = 0 \end{aligned}$$

Gautą išraišką supaprastinę turime:

$$a_{31} + \frac{a_{11}a_{21} + (a_{11}^2 - a_{11}a_{22})\alpha^*}{a_{23}} - \alpha^*a_{32} + \left(\frac{(a_{22}a_{33} - a_{11}a_{33})\alpha^* - a_{21}a_{33}}{a_{23}}\right) = 0$$

Dabar iš gautos išraiškos galime išsireikšti parametro α^* reikšmę:

$$\alpha^* = \frac{a_{21}a_{33} - a_{23}a_{31} - a_{11}a_{21}}{a_{11}^2 - a_{11}a_{22} - a_{23}a_{32} + a_{22}a_{33} - a_{11}a_{33}}. \blacksquare$$

Jeigu $a_{11}^2 - a_{11}a_{22} - a_{23}a_{32} + a_{22}a_{33} - a_{11}a_{33} = 0$, tada iš (1.38) sistemos matome, kad šis faktas pirmoms penkioms lygtims ir parametrams β^* ir γ^* įtakos neturi, tačiau suprastinus šeštąją lygtį, gauname:

$$(a_{11}^2 - a_{11}a_{22} - a_{23}a_{32} + a_{22}a_{33} - a_{11}a_{33})\alpha^* = a_{21}a_{33} - a_{23}a_{31} - a_{11}a_{21}$$

Taigi matome, kad norint jog lygybė būtų teisinga, išraiška $a_{21}a_{33} - a_{23}a_{31} - a_{11}a_{21}$ turi būti lygi nuliui. Tokiu atveju lygybė bus teisinga su bet koku $\alpha^* \in \mathbb{C}$.

2. Kai $a_{13} = 0, a_{12} \neq 0, a_{23} = 0$, parametrai gaunami tokiu būdu:

$$\alpha^* = \frac{a_{11} - \lambda_1}{a_{12}} \quad (1.60)$$

$$\beta^* = \frac{a_{11}a_{32} - a_{12}a_{31} - a_{32}a_{33}}{a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21} + a_{22}a_{33} - a_{33}^2} \quad (1.61)$$

$$\gamma^* = \frac{(a_{33} - a_{22})\beta^* - a_{32}}{a_{12}} - \frac{\alpha^*\beta^*}{2} \quad (1.62)$$

Čia $a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21} + a_{22}a_{33} - a_{33}^2 \neq 0$.

Irodymas. Įstatę $a_{13} = 0, a_{23} = 0$ reikšmes į (1.38) sistemos lygtis, gauname tokią naują lygčių sistemą:

$$\left\{ \begin{array}{l} a_{12}\alpha^* = a_{11} - \lambda_1 \\ -a_{12}\alpha^* = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12} = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right) a_{12} = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right) a_{11} + \beta^* a_{21} - \alpha^* \left(\left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right) a_{12} + \beta^* a_{22} + a_{32} \right) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right) a_{33} = 0 \end{array} \right.$$

Iš pirmosios ir penktosios šios sistemos lygčių, atitinkamai gauname parametrų α^* ir γ^* išraiškas:

$$\alpha^* = \frac{a_{11} - \lambda_1}{a_{12}}; \gamma^* = \frac{(a_{33} - a_{22})\beta^* - a_{32}}{a_{12}} - \frac{\alpha^*\beta^*}{2}$$

Dabar, įstatę gautą γ^* išraišką į sistemos šeštąją lygtį, gauname:

$$a_{31} + \left(\frac{(a_{33} - a_{22})\beta^* - a_{32}}{a_{12}} \right) a_{11} + \beta^* a_{21} - \alpha^* \left(\frac{(a_{33} - a_{22})\beta^* - a_{32}}{a_{12}} a_{12} + a_{22}\beta^* + a_{32} \right) \\ + \left(\alpha^*\beta^* + \frac{a_{32} + (a_{22} - a_{33})\beta^*}{a_{12}} \right) a_{33} = 0$$

Gautą išraišką supaprastinę turime:

$$\frac{(a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21})\beta^* + a_{12}a_{31} - a_{11}a_{32} + a_{32}a_{33} + (a_{22}a_{33} - a_{33}^2)\beta^*}{a_{12}} = 0$$

Iš gautos išraiškos turime parametro β^* reikšmę:

$$\beta^* = \frac{a_{11}a_{32} - a_{12}a_{31} - a_{32}a_{33}}{a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21} + a_{22}a_{33} - a_{33}^2}. \blacksquare$$

Jeigu $a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21} + a_{22}a_{33} - a_{33}^2 = 0$, tada iš (1.38) sistemos matome, kad tai pirmoms penkioms lygtims ir parametrams α^* ir γ^* įtakos neturi, tačiau supaprastinus šeštąją lygtį, gauname:

$$(a_{11}a_{33} - a_{11}a_{22} + a_{12}a_{21} + a_{22}a_{33} - a_{33}^2)\beta^* = a_{11}a_{32} - a_{12}a_{31} - a_{32}a_{33}$$

Taigi matome, kad norint jog lygybė būtų teisinga, išraiška $a_{11}a_{32} - a_{12}a_{31} - a_{32}a_{33}$ taip pat turi būti lygi nuliui. Tokiu atveju lygybė bus teisinga su bet koku $\beta^* \in \mathbb{C}$.

3. Kai $a_{13} = 0, a_{12} = 0, a_{23} = 0$ parametrai gaunami tokiu būdu:

$$\alpha^* = \frac{a_{21}}{a_{22} - a_{11}} \quad (1.63)$$

$$\beta^* = \frac{a_{32}}{a_{33} - a_{22}} \quad (1.64)$$

$$\gamma^* = \frac{a_{31}}{a_{33}-a_{11}} - \frac{a_{21}a_{32}(a_{11}+a_{33})}{2(a_{33}-a_{22})(a_{22}-a_{11})(a_{33}-a_{11})} \quad (1.65)$$

Čia $a_{22} - a_{11} \neq 0$, $a_{33} - a_{22} \neq 0$, $a_{33} - a_{11} \neq 0$, t.y. $a_{11} \neq a_{22} \neq a_{33}$.

Irodymas. Į (1.38) sistemos lygtis įstačius $a_{13} = 0$, $a_{12} = 0$, $a_{23} = 0$, gauname tokią lygčių sistemą:

$$\left\{ \begin{array}{l} 0 = a_{11} - \lambda_1 \\ 0 = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{33} = 0 \end{array} \right.$$

Iš ketvirtosios ir penktosios šios sistemos lygčių atitinkamai gauname α^* ir β^* parametru reikšmes:

$$\alpha^* = \frac{a_{21}}{a_{22} - a_{11}}; \beta^* = \frac{a_{32}}{a_{33} - a_{22}}$$

Pertvarkę šeštąją sistemos lygtį turime:

$$(a_{33} - a_{11})\gamma^* = \frac{1}{2}\alpha^*\beta^*(a_{11} - 2a_{22} + a_{33}) + a_{31} - \alpha^*a_{32} + \beta^*a_{21}$$

Dabar į gautą išraišką įstačius gautas α^* ir β^* reikšmes gauname:

$$(a_{33} - a_{11})\gamma^* = a_{31} - \frac{a_{21}a_{32}(a_{11} + a_{33})}{2(a_{33} - a_{22})(a_{22} - a_{11})}$$

Iš šios išraiškos turime γ^* parametro reikšmę:

$$\gamma^* = \frac{a_{31}}{a_{33}-a_{11}} - \frac{a_{21}a_{32}(a_{11}+a_{33})}{2(a_{33}-a_{22})(a_{22}-a_{11})(a_{33}-a_{11})}. \blacksquare$$

Kai $a_{11} = a_{22} \neq a_{33}$, tada bet koks $\alpha^* \in \mathbb{C}$ tenkins (1.38) lygčių sistemą, kai parametras γ^* apskaičiuojamas tokiu būdu (čia β^* skaičiuojamas pagal (1.64) išraišką):

$$\gamma^* = \frac{a_{31} - a_{32}\alpha^* + a_{21}\beta^*}{a_{33} - a_{11}} + \frac{\alpha^*\beta^*}{2} \quad (1.66)$$

Irodymas. Prie šių sąlygų, sistema atrodoys taip:

$$\left\{ \begin{array}{l} 0 = a_{11} - \lambda_1 \\ 0 = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{33} = 0 \end{array} \right.$$

Kaip matome iš šios sistemos, penktoji lygtis nepasikeitė, todėl parametą β^* vis dar galime apskaičiuoti pagal (1.64) išraišką. Taip pat galime pastebėti, kad pirmoms penkioms lygtims parametą α^* ir γ^* reikšmės įtakos neturi, taigi jų reikšmės svarbios tik šeštojoje lygtyje, kurią supaprastinus ir vietoj a_{22} įrašius a_{11} (nes $a_{11} = a_{22}$), gauname:

$$\gamma^*(a_{11} - a_{33}) + a_{31} + \beta^*a_{21} - \alpha^*a_{32} + \frac{1}{2}\alpha^*\beta^*a_{33} + \frac{1}{2}\alpha^*\beta^*a_{11} - \alpha^*\beta^*a_{11} = 0$$

Iš šios lygties išsireiškiame γ^* parametą ir turime:

$$\gamma^* = \frac{a_{31} - a_{32}\alpha^* + a_{21}\beta^*}{a_{33} - a_{11}} + \frac{\alpha^*\beta^*}{2}$$

Taigi matome, kad lygybė tenkinama su bet koku $\alpha^* \in \mathbb{C}$, jei γ^* apskaičiuojamas pagal pateiktą išraišką. ■

Kai $a_{11} \neq a_{22} = a_{33}$, tada bet koks $\beta^* \in \mathbb{C}$ tenkins (1.38) lygčių sistemą, kai parametras γ^* apskaičiuojamas tokiu būdu (čia α^* skaičiuojamas pagal (1.63) išraišką):

$$\gamma^* = \frac{a_{31} - a_{32}\alpha^* + a_{21}\beta^*}{a_{33} - a_{11}} - \frac{\alpha^*\beta^*}{2} \quad (1.67)$$

Irodymas. Prie šių sąlygų, sistema atrodo taip:

$$\left\{ \begin{array}{l} 0 = a_{11} - \lambda_1 \\ 0 = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) = 0 \\ a_{32} = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{33} = 0 \end{array} \right.$$

Iš pateiktos lygčių sistemos matome, kad ketvirtoji lygtis liko ta pati, todėl parametras α^* skaičiuojamas pagal (1.63) išraišką. Be to, iš sistemos matome, kad esant šioms sąlygoms, pirmoms penkioms lygtims parametrai β^* ir γ^* įtakos neturi, taigi jų reikšmės svarbios tik šeštojoje lygtyje. Pateiktą išraišką supaprastinus ir vietoj a_{22} įrašius a_{33} (nes $a_{33} = a_{22}$), gauname:

$$\gamma^*(a_{11} - a_{33}) + a_{31} + \beta^*a_{21} - \alpha^*a_{32} + \frac{1}{2}\alpha^*\beta^*a_{11} + \frac{1}{2}\alpha^*\beta^*a_{33} - \alpha^*\beta^*a_{33} = 0$$

Iš šios lygties išsireiškiame γ^* parametą:

$$\gamma^* = \frac{a_{31} - a_{32}\alpha^* + a_{21}\beta^*}{a_{33} - a_{11}} - \frac{\alpha^*\beta^*}{2}$$

Taigi matome, kad lygybė tenkinama su bet koku $\beta^* \in \mathbb{C}$, kai γ^* apskaičiuojama pagal pateiktą išraišką. ■

Kai $a_{11} = a_{33} \neq a_{22}$, tada bet koks $\gamma^* \in \mathbb{C}$ tenkins (1.38) lygčių sistemą, o parametrai α^* ir β^* atitinkamai apskaičiuojami pagal (1.63) ir (1.64) išraiškas.

Irodymas. Prie šių sąlygų, (1.38) lygčių sistema atrodys taip:

$$\left\{ \begin{array}{l} 0 = a_{11} - \lambda_1 \\ 0 = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} + \alpha^*(a_{11} - a_{22}) = 0 \\ a_{32} + \beta^*(a_{22} - a_{33}) = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{33} = 0 \end{array} \right.$$

Iš penktosios ir šeštosios šios sistemos lygčių matome, kad α^* ir β^* atitinkamai gaunami pagal (1.63) ir (1.64) išraiškas. Be to, pirmose penkiose lygtyse, parametras γ^* neegzistuoja. Pertvarkę šeštąją lygtį, turime:

$$\gamma^*(a_{11} - a_{33}) + a_{31} + \beta^*a_{21} - \alpha^*a_{32} + \frac{1}{2}\alpha^*\beta^*a_{11} + \frac{1}{2}\alpha^*\beta^*a_{33} - \alpha^*\beta^*a_{33} = 0$$

Kaip matome iš gautos išraiškos, daugiklis prie γ^* yra lygus nuliui, todėl su bet koku $\gamma^* \in \mathbb{C}$ bus tenkinamos lygybės, nes jis įtakos lygtims neturi. ■

Kai $a_{11} = a_{22} = a_{33}$, tada bet koks $\gamma^* \in \mathbb{C}$ tenkins (1.38) lygčių sistemą, o parametrams α^* ir β^* turi galioti tokia lygybė:

$$a_{32}\alpha^* - a_{21}\beta^* = a_{31} \quad (1.68)$$

Irodymas. Įstačius šias sąlygas į (1.38) sistemos lygtis, turime tokią lygčių sistemą:

$$\left\{ \begin{array}{l} 0 = a_{11} - \lambda_1 \\ 0 = a_{22} - \lambda_2 \\ 0 = a_{33} - \lambda_3 \\ a_{21} = 0 \\ a_{32} = 0 \\ a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_{11} + \beta^*a_{21} - \alpha^*(\beta^*a_{22} + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_{33} = 0 \end{array} \right.$$

Kaip matome iš pirmųjų penkių lygčių, parametrų $\alpha^*, \beta^*, \gamma^*$ reikšmės lygtims įtakos neturi. Įvedame pakeitimą $a_0 = a_{11} = a_{22} = a_{33}$ ir šeštosios lygties gauname:

$$\begin{aligned} a_{31} + \left(\frac{1}{2}\alpha^*\beta^* + \gamma^*\right)a_0 + \beta^*a_{21} - \alpha^*(\beta^*a_0 + a_{32}) \\ + \left(\frac{1}{2}\alpha^*\beta^* - \gamma^*\right)a_0 = 0 \end{aligned}$$

Gautą išraišką supaprastinę, turime:

$$a_{31} + \beta^*a_{21} - \alpha^*a_{32} + \frac{1}{2}\alpha^*\beta^*a_0 + \frac{1}{2}\alpha^*\beta^*a_0 - \alpha^*\beta^*a_0 + \gamma^*(a_0 - a_0) = 0$$

Iš šios išraiškos matome, kad:

$$a_{32}\alpha^* - a_{21}\beta^* = a_{31}. \quad \blacksquare$$

Kiti atvejai. Situacija, kai išpildoma tik viena iš (1.43) ir (1.44) sąlygų, o (1.45) sąlyga netenkinama, yra negalima.

Irodymas. Tarkime (1.43) nelygybė yra teisinga, o (1.44) ir (1.45) sąlygos netenkinamos. Tada gauname tokią lygčių sistemą:

$$\begin{cases} a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 \neq 0 \\ a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 = 0 \\ a_{13} = 0 \end{cases}$$

Iš pateiktos lygčių sistemos trečiosios lygybės matome, kad a_{13} turi būti lygus nuliui. Įstatę šią reikšmę į pirmąsias dvi lygtis, gauname tokią sistemą:

$$\begin{cases} a_{12}a_{23} \neq 0 \\ -a_{12}a_{23} = 0 \end{cases}$$

Iš gautos sistemos matome, jog gavome prieštarą, taigi toks atvejis yra negalimas. Analogiškai galima parodyti, jog atvejis, kai (1.44) nelygybė yra teisinga, o (1.43) ir (1.45) sąlygos netenkinamos, taip pat yra negalimas. ■

1.11. TRIKAMPĖS MATRICOS ALGEBRINĖS IŠRAIŠKOS

Šiame skyrelyje aprašysime trikampės trečios eilės matricos transformacijas. 1.9 skyrelyje atlikome kvadratinės trečios eilės matricos A transformaciją ir gavome trikampę matricą B :

$$B = \begin{pmatrix} \lambda_1 & a & c \\ 0 & \lambda_2 & b \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad (1.69)$$

Gautos matricos B transformacija priklauso nuo jos tikrinių reikšmių $\lambda_1, \lambda_2, \lambda_3$. Jeigu visos tikrinės reikšmės yra skirtingos, su pateikta transformacija, matricą B diagonaluosime, o jeigu bent dvi iš tikrinių reikšmių sutampa, jos diagonalizuoti negalime ir gausime kitokią jos išraišką.

Pirmoji situacija. Jeigu matrica B turi tris skirtingas tikrines reikšmes $\lambda_1, \lambda_2, \lambda_3$, tai ją galime išreikšti tokiu idempotentų tiesiniu dariniu:

$$B = \lambda_1 D_1 + \lambda_2 D_2 + \lambda_3 D_3 \quad (1.70)$$

Čia idempotentai D_1, D_2, D_3 įgija tokias reikšmes:

$$D_1 = \begin{pmatrix} 1 & \frac{a}{\lambda_1 - \lambda_2} & \frac{ab}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} + \frac{c}{\lambda_1 - \lambda_3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (1.71)$$

$$D_2 = \begin{pmatrix} 0 & \frac{a}{\lambda_2 - \lambda_1} & \frac{ab}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} \\ 0 & 1 & \frac{b}{\lambda_2 - \lambda_3} \\ 0 & 0 & 0 \end{pmatrix} \quad (1.72)$$

$$D_3 = \begin{pmatrix} 0 & 0 & \frac{ab}{(\lambda_3 - \lambda_2)(\lambda_3 - \lambda_1)} + \frac{c}{\lambda_3 - \lambda_1} \\ 0 & 0 & -\frac{b}{\lambda_3 - \lambda_2} \\ 0 & 0 & 1 \end{pmatrix} \quad (1.73)$$

Tada egzistuoja matricų K ir K^{-1} pora, kai:

$$K = \begin{pmatrix} 1 & \alpha & \frac{1}{2}\alpha\beta + \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix}, K^{-1} = \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha\beta - \gamma \\ 0 & 1 & -\beta \\ 0 & 0 & 1 \end{pmatrix} \quad (1.74)$$

Su kuriomis teisingos tokios lygybės:

$$KBK^{-1} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad (1.75)$$

$$B = K^{-1} \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} K \quad (1.76)$$

Matricų K ir K^{-1} parametrai α, β, γ apskaičiuojami pagal šias išraiškas:

$$\alpha = \frac{a}{\lambda_1 - \lambda_2} \quad (1.77)$$

$$\beta = \frac{b}{\lambda_2 - \lambda_3} \quad (1.78)$$

$$\gamma = \frac{\alpha\beta(2\lambda_2 - \lambda_1 - \lambda_3) + 2(-c + \beta a - \alpha b)}{2(\lambda_3 - \lambda_1)} \quad (1.79)$$

Taigi, kai matricos A visos tikrinės reikšmės yra skirtingos, ji diagonalizuojama tokiu būdu:

$$L = KTAT^{-1}K^{-1} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad (1.80)$$

Be to, kiekvieną matricą A galima nusakyti tokiais hierarchizuotais parametrais: $A = A(\lambda_1, \lambda_2, \lambda_3, \alpha, \beta, \gamma, \alpha^*, \beta^*, \gamma^*)$, t.y. galima gauti tokį jos dėstinį:

$$A = T^{-1}K^{-1}LKT \quad (1.81)$$

Pastebėsime, kad idempotentus D_1, D_2, D_3 galima gauti iš matricų K ir K^{-1} :

$$1. \quad K \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} K^{-1} = \begin{pmatrix} 1 & \frac{a}{\lambda_1 - \lambda_2} & \frac{ab}{(\lambda_1 - \lambda_2)(\lambda_1 - \lambda_3)} + \frac{c}{\lambda_1 - \lambda_3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

$$2. \quad K \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix} K^{-1} = \begin{pmatrix} 0 & \frac{a}{\lambda_2 - \lambda_1} & \frac{ab}{(\lambda_2 - \lambda_1)(\lambda_2 - \lambda_3)} \\ 0 & 1 & \frac{b}{\lambda_2 - \lambda_3} \\ 0 & 0 & 0 \end{pmatrix}$$

$$3. \quad K \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} K^{-1} = \begin{pmatrix} 0 & 0 & \frac{ab}{(\lambda_3 - \lambda_2)(\lambda_3 - \lambda_1)} + \frac{c}{\lambda_3 - \lambda_1} \\ 0 & 0 & -\frac{b}{\lambda_3 - \lambda_2} \\ 0 & 0 & 1 \end{pmatrix}$$

Antroji situacija.

1. Jei matrica B turi dvi vienodas tikrines reikšmes, o trečia skirtinga, t.y. $\lambda_1 = \lambda_2 := \lambda_0 \neq \lambda_3$, tai matricą B galime išdėstyti tokiu idempotentų bei nulpotentų tiesiniu dariniu:

$$B = \lambda_0 D_1 + a N_1 + \lambda_3 D_2 \quad (1.82)$$

Čia idempotentai D_1, D_2 įgija tokias reikšmes:

$$D_1 = \begin{pmatrix} 1 & 0 & -\frac{ab}{(\lambda_0 - \lambda_3)^2} + \frac{c}{\lambda_0 - \lambda_3} \\ 0 & 1 & \frac{b}{\lambda_0 - \lambda_3} \\ 0 & 0 & 0 \end{pmatrix} \quad (1.83)$$

$$D_2 = \begin{pmatrix} 0 & 0 & \frac{ab}{(\lambda_0 - \lambda_3)^2} - \frac{c}{\lambda_0 - \lambda_3} \\ 0 & 0 & -\frac{b}{\lambda_0 - \lambda_3} \\ 0 & 0 & 1 \end{pmatrix} \quad (1.84)$$

Nulpotentas N_1 išreiškiamas šia lygybe:

$$N_1 = \begin{pmatrix} 0 & 1 & \frac{b}{\lambda_0 - \lambda_3} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (1.85)$$

Tada egzistuoja matricų K ir K^{-1} pora su parametrais:

$$\beta = \frac{b}{\lambda_0 - \lambda_3} \quad (1.86)$$

$$\gamma = \frac{\alpha\beta(\lambda_0 - \lambda_3) + 2(\beta a - ab - c)}{2(\lambda_3 - \lambda_0)} \quad (1.87)$$

Bei bet kokių $\alpha \in \mathbb{C}$, kuris tenkins tapatybę:

$$\begin{pmatrix} 1 & \alpha & \frac{1}{2}\alpha\beta + \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_0 & a & c \\ 0 & \lambda_0 & b \\ 0 & 0 & \lambda_3 \end{pmatrix} \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha\beta - \gamma \\ 0 & 1 & -\beta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda_0 & a & 0 \\ 0 & \lambda_0 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix} \quad (1.88)$$

2. Jei matrica B turi dvi vienodas tikrines reikšmes, o pirmąją skirtingą, t.y. $\lambda_1 \neq \lambda_0 = \lambda_2 = \lambda_3$, tada transformacijai naudojama matricų pora K ir K^{-1} su parametrais:

$$\alpha = \frac{a}{\lambda_1 - \lambda_0} \quad (1.89)$$

$$\gamma = \frac{\alpha\beta(\lambda_0 - \lambda_1) - 2(\beta a - ab - c)}{2(\lambda_0 - \lambda_1)} \quad (1.90)$$

Bei bet kokių $\beta \in \mathbb{C}$, kuris tenkins tapatybę:

$$\begin{pmatrix} 1 & \alpha & \frac{1}{2}\alpha\beta + \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_1 & a & c \\ 0 & \lambda_0 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha\beta - \gamma \\ 0 & 1 & -\beta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_0 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \quad (1.91)$$

3. Jei matrica B turi dvi vienodas tikrines reikšmes, o antrąją skirtingą, t.y. $\lambda_2 \neq \lambda_0 = \lambda_1 = \lambda_3$, tada transformacijai atlikti imame matricų porą K ir K^{-1} su parametrais:

$$\alpha = \frac{a}{\lambda_0 - \lambda_2} \quad (1.92)$$

$$\beta = \frac{b}{\lambda_2 - \lambda_0} \quad (1.93)$$

Bei bet kokiu $\gamma \in \mathbb{C}$, kuris tenkins tapatybę:

$$\begin{pmatrix} 1 & \alpha & \frac{1}{2}\alpha\beta + \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_0 & a & c \\ 0 & \lambda_2 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha\beta - \gamma \\ 0 & 1 & -\beta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda_0 & 0 & c + \frac{ab}{\lambda_0 - \lambda_2} \\ 0 & \lambda_2 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \quad (1.94)$$

Trečioji situacija. Jeigu matrica B turi tris vienodas tikrines reikšmes, t.y. $\lambda_1 = \lambda_2 = \lambda_3 := \lambda_0$, tai matricą B galime išdėstyti tokiu idempotentų bei nulpotentų tiesiniu dariniu:

$$B = \lambda_0 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} + \begin{pmatrix} 0 & a & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (1.95)$$

Čia:

$$\begin{pmatrix} 0 & a & 0 \\ 0 & 0 & b \\ 0 & 0 & 0 \end{pmatrix}^2 = \begin{pmatrix} 0 & 0 & ab \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad \begin{pmatrix} 0 & 0 & c \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}^2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Tada matricų pora K ir K^{-1} , su bet kokiu $\gamma \in \mathbb{C}$ ir parametrais α ir β , tenkinančiais tapatybę $\alpha b - \beta a + c = 0$, nusako tokią atitiktį:

$$\begin{pmatrix} 1 & \alpha & \frac{1}{2}\alpha\beta + \gamma \\ 0 & 1 & \beta \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \lambda_0 & a & c \\ 0 & \lambda_0 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \begin{pmatrix} 1 & -\alpha & \frac{1}{2}\alpha\beta - \gamma \\ 0 & 1 & -\beta \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \lambda_0 & a & 0 \\ 0 & \lambda_0 & b \\ 0 & 0 & \lambda_0 \end{pmatrix} \quad (1.96)$$

Šiame skyrelyje pateiktas trikampių trečios eilės matricų transformacijas bei jų pagalba gaunamus algebrinius–struktūrinius parametrus $\alpha^*, \beta^*, \gamma^*$ naudosime iš funkcijų gautų sinchroninių eilučių bei elektrokardiogramos duomenų analizei.

2. TIRIAMOJI DALIS IR REZULTATAI

2.1. TREČIOS EILĖS MATRICOS TRANSFORMACIJŲ PAVYZDŽIAI

Šiame skyrelyje, pagal aprašytus matematinius metodus, atliksime kvadratinės trečios eilės matricos transformacijas bei išskirsime jos algebrinius–struktūrinius parametrus. Skaičiavimus atliekame sukurtos programinės įrangos pagalba (žr. 2.4 skyrelį). Programoje rezultatai spausdinami 10^{-16} tikslumu, tačiau šiuose pavyzdžiuose parodysime tik tris skaičius po kablelio, kad būtų paprasčiau apžvelgti rezultatus.

I pavyzdys. Tegul turime tokią trečios eilės matricą:

$$A = \begin{pmatrix} 1 & 8 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix}$$

Matricos A tikrinės reikšmės:

$$\lambda_1 = -0,797 - 0,251i$$

$$\lambda_2 = -0,797 + 0,251i$$

$$\lambda_3 = 8,594$$

Dabar pagal (1.40), (1.41) ir (1.42) išraiškas, gauname parametrų α^* , β^* , γ^* reikšmes:

$$\alpha^* = 0,198 + 0,037i$$

$$\beta^* = 1,228$$

$$\gamma^* = 0,333 - 0,023i$$

Gautų parametrų pagalba pagal (1.34) išraiškas, gauname matricų T ir T^{-1} porą:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0,198 + 0,037i & 1 & 0 \\ 0,455 + 5,551i & 1,228 & 1 \end{pmatrix}; T^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -0,198 - 0,037i & 1 & 0 \\ 0,212 + 0,046i & -1,228 & 1 \end{pmatrix}$$

Šių matricų pagalba, gauname trikampę matricą B :

$$B = TAT^{-1} = \begin{pmatrix} -0,797 - 0,251i & 6,772 & 1 \\ 0 & -0,797 + 0,251i & 5,198 + 0,037i \\ 0 & 0 & 8,594 \end{pmatrix}$$

Pagal (1.77), (1.78) ir (1.79) išraiškas gauname parametrų α , β , γ reikšmes:

$$\alpha = 13,489i$$

$$\beta = -0,553 - 0,019i$$

$$\gamma = -0,379 - 3,730i$$

Gautų parametrų pagalba pagal iš (1.74) išraiškų, gauname matricų K ir K^{-1} porą:

$$K = \begin{pmatrix} 1 & 13,489i & -0,253 - 7,460i \\ 0 & 1 & -0,553 - 0,019i \\ 0 & 0 & 1 \end{pmatrix}; K^{-1} = \begin{pmatrix} 1 & -13,489i & 0,505 \\ 0 & 1 & 0,553 + 0,019i \\ 0 & 0 & 1 \end{pmatrix}$$

Gautų pagalbinių matricų pagalba diagonalizuojame iš matricos A gautą trikampę matricą B ir gauname matricą L :

$$L = KBK^{-1} = \begin{pmatrix} -0,797 - 0,251i & 0 & 0 \\ 0 & -0,797 + 0,251i & 0 \\ 0 & 0 & 8,594 \end{pmatrix}$$

Kaip matome, gautos matricos įstrižainės parametrai yra lygūs matricos A tikrinėms reikšmėms. Atlikę skaičiavimus, galime gauti tokį matricos A dėstinį:

$$A = T^{-1}K^{-1}LKT = \begin{pmatrix} 1 & 8 & 1 \\ 2 & 4 & 5 \\ 1 & 2 & 2 \end{pmatrix}$$

2 pavyzdys. Šiame pavyzdyje pateiktas atvejis, kai matrica išsigimusi ir struktūrinių parametru α^* , β^* , γ^* negalime apskaičiuoti formulėmis, kurios galioja bendram atvejui. Tegul turime tokią trečios eilės matricą:

$$A = \begin{pmatrix} 2 & 4 & 0 \\ 0 & 1 & 1 \\ 0 & 4 & 4 \end{pmatrix}$$

Matricos A tikrinės reikšmės:

$$\lambda_1 = 0; \lambda_2 = 2; \lambda_3 = 5$$

Norint apskaičiuoti parametrus α^* , β^* , γ^* , tikriname sąlygas (1.43), (1.44) ir (1.45):

$$a_{12}a_{23} - a_{22}a_{13} + a_{13}\lambda_1 = 4 \cdot 1 - 1 \cdot 0 + 0 \cdot 0 \neq 0$$

$$a_{22}a_{13} - a_{12}a_{23} - a_{13}\lambda_3 = 1 \cdot 0 - 4 \cdot 1 - 0 \cdot 5 \neq 0$$

$$a_{13} = 0$$

Kaip matome, viena iš sąlygų yra netenkinama, todėl struktūriniai parametrai turi būti skaičiuojami pagal 1.10 skyrelyje pateiktus parametru skaičiavimo atskirus atvejus. Kadangi netenkinama trečioji sąlyga, parametrai apskaičiuojami pagal (1.46), (1.47) ir (1.48) lygybes:

$$\alpha^* = \frac{a_{11} - \lambda_1}{a_{12}} = \frac{2 - 0}{4} = 0,5$$

$$\beta^* = \frac{\lambda_3 - a_{33}}{a_{23}} = \frac{5 - 4}{1} = 1$$

$$\gamma^* = \frac{a_{21} + \alpha^*(a_{11} - a_{22}) - (\alpha^*)^2 a_{12}}{a_{23}} + \frac{\alpha^* \beta^*}{2} = \frac{0,5(2 - 1) - (0,5)^2 4}{1} + \frac{0,5}{2} = -0,25$$

Iš gautų parametru, pagal (1.34) išraiškas, sudarome matricų T ir T^{-1} porą:

$$T = \begin{pmatrix} 1 & 0 & 0 \\ 0,5 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}; T^{-1} = \begin{pmatrix} 1 & 0 & 0 \\ -0,5 & 1 & 0 \\ 0,5 & -1 & 1 \end{pmatrix}$$

Šių matricų pagalba, gauname trikampę matricą B :

$$B = TAT^{-1} = \begin{pmatrix} 1 & 4 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 5 \end{pmatrix}$$

Pagal (1.77), (1.78) ir (1.79) išraiškas gauname parametų α, β, γ reikšmes:

$$\begin{aligned}\alpha &= -2 \\ \beta &= -0,334 \\ \gamma &= 0,067\end{aligned}$$

Iš gautų parametų, pagal iš (1.74) išraiškas, gauname matricų K ir K^{-1} porą:

$$K = \begin{pmatrix} 1 & -2 & 0,4 \\ 0 & 1 & -0,334 \\ 0 & 0 & 1 \end{pmatrix}; K^{-1} = \begin{pmatrix} 1 & 2 & 0,267 \\ 0 & 1 & 0,334 \\ 0 & 0 & 1 \end{pmatrix}$$

Gautų pagalbinių matricų pagalba diagonalizuojame iš matricos A gautą trikampę matricą B ir gauname matricą L :

$$L = KBK^{-1} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 5 \end{pmatrix}$$

Kaip matome, gautos matricos įstrižainės parametrai yra lygūs matricos A tikrinėms reikšmėms. Atlikę skaičiavimus, galime gauti tokį matricos A dėstinį:

$$A = T^{-1}K^{-1}LKT = \begin{pmatrix} 2 & 4 & 0 \\ 0 & 1 & 1 \\ 0 & 4 & 4 \end{pmatrix}$$

Atlikti skaičiavimai parodo, kad atlikus aprašytas vienos trečios eilės matricos transformacijas, matricą A galime nusakyti tokiais hierarchizuotais parametrais: $A = A(\lambda_1, \lambda_2, \lambda_3, \alpha, \beta, \gamma, \alpha^*, \beta^*, \gamma^*)$.

2.2. TRIJŲ FUNKCIJŲ ALGEBRINĖ KORELIACIJA

Šiame skyrelyje tirsime trijų funkcijų algebrinę koreliaciją, naudodami iš jų gaunamas tris sinchronizuotas laiko eilutes:

$$\begin{aligned}x &= \cos(\rho_1 t + \varphi_1) \\ y &= \cos(\rho_2 t + \varphi_2) \\ z &= \cos(\rho_3 t + \varphi_3)\end{aligned}$$

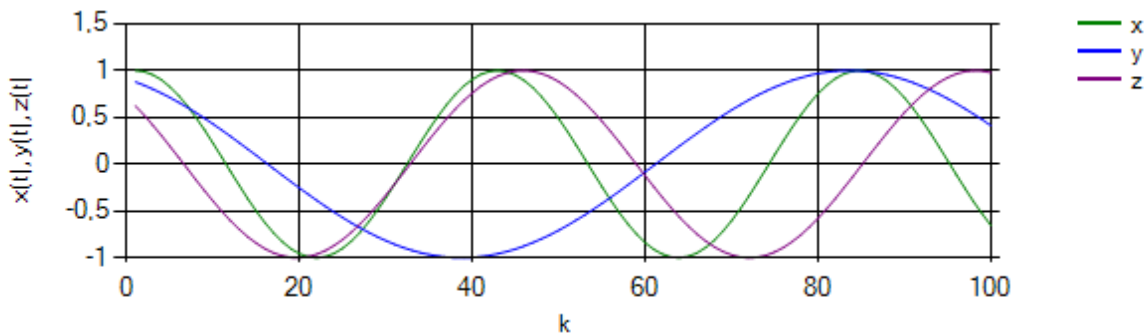
Čia $t = hn$ apibūdina laiko momentus, kur h - laiko intervalo žingsnis, $n = \overline{0, k}$ (k - eilutės narių kiekis). Parametrai $\rho_i, i = 1, 2, 3$ apibūdina dažnius, o $\varphi_i, i = 1, 2, 3$ - fazinius postūmius. Tegul turime tokias funkcijos parametų reikšmes:

2.1 lentelė

Funkcijų parametų reikšmės

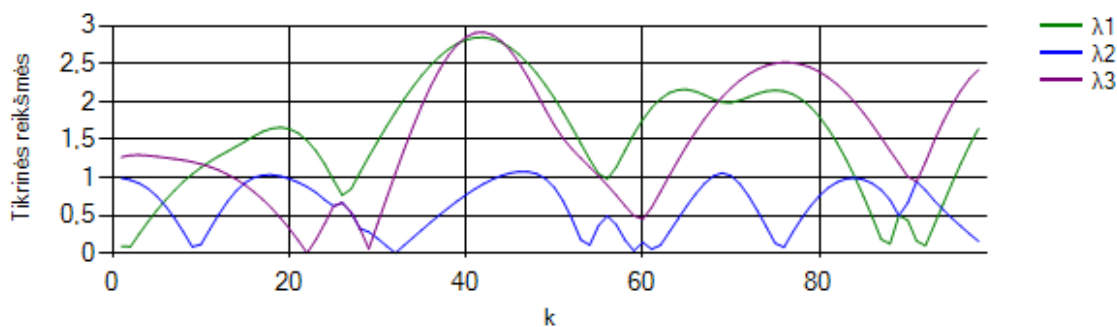
Parametras	ρ_1	ρ_2	ρ_3	φ_1	φ_2	φ_3	h	k
Reikšmė	1,5	0,7	1,2	0	0,5	0,9	0,1	100

Parinę šių parametru reikšmes, naudodamiesi sukurta programine įranga (žr. 2.4 skyrelį), sudarome Lagranžo skirtumų matricių seką (žr. 1.4 skyrelį) ir iš jų išskiriame matricas nusakančius parametrus $\lambda_1, \lambda_2, \lambda_3, \alpha, \beta, \gamma, \alpha^*, \beta^*, \gamma^*$ bei pavaizduojame juos grafiškai. Gautų sinchroninių eilučių grafikas:



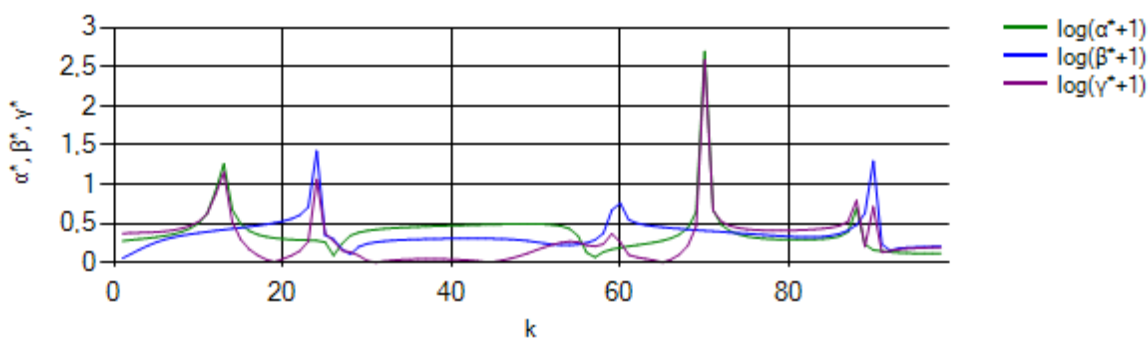
2.1 pav. Sinchroninių eilučių grafikas

Apskaičiuotas tikrines reikšmes $\lambda_1, \lambda_2, \lambda_3$ surikiuojame, tada apskaičiuojame jų modulį ir brėžiame grafiką:

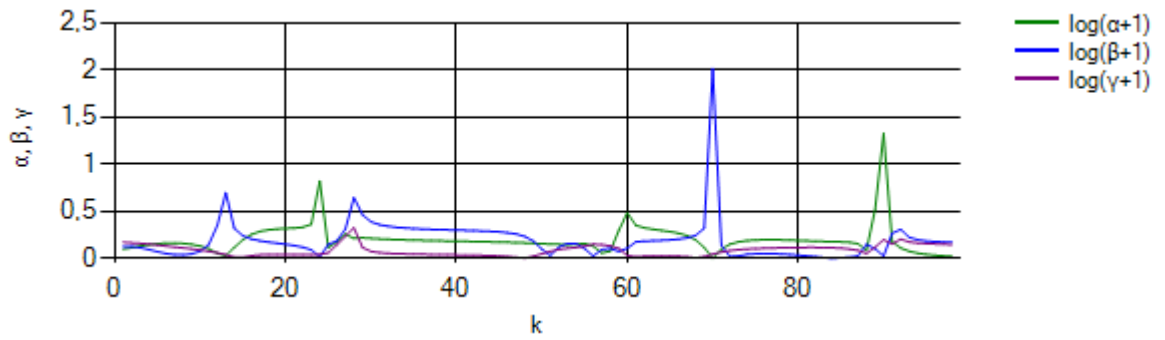


2.2 pav. Tikrinių reikšmių grafikas

Norint aiškiau matyti matricos struktūrinių parametru dinamiką, juos atvaizduojame logaritminėje skalėje, t.y. kai $x = \{\alpha, \beta, \gamma, \alpha^*, \beta^*, \gamma^*\}$, brėžiame $\log_{10}|x + 1|$ grafiką. Perskaičius struktūrinius parametrus, gauname tokius grafikus:



2.3 pav. Parametru $\alpha^*, \beta^*, \gamma^*$ grafikas



2.4 pav. Parametrų α , β , γ grafikas

Iš pateiktų grafikų matome, kad struktūrinių parametrų moduliai didžiausias reikšmes įgyja, kai funkcijos viena prie kitos artėja. Taip pat galima pastebėti, kad kuo didesnis atstumas tarp funkcijų, tuo pastovesnės struktūrinių parametrų modulių reikšmės.

2.3. ELEKTROKARDIOGRAMOS PARAMETRŲ TYRIMAS

Tyrimui atlikti buvo pasirinkta dešimties sportininkų elektrokardiogramos. EKG duomenys buvo fiksuojami 11 minučių: pirmąją minutę ramybės būsenoje, sekančias 5 minutes buvo minamas specialus dviratis, kas minutę didinant krūvį (50W, 100W, 150W, 200W, 250W), o paskutines 5 minutes vyko atsigavimas. Tyrimui atlikti nagrinėjame trijų EKG parametrų dinamines sąsajas: RR, DQRS, DJT. Kadangi šių parametrų reikšmių aibės labai skiriasi, todėl norint juos tarpusavyje lyginti, duomenys buvo normuojami:

$$x'_i = \frac{x_i - 2x_{min}}{x_{max}}$$

Čia i – kardiociklo numeris, x_i – pradinė parametro reikšmė taške i , x'_i - sunormuota parametro reikšmė taške i . x_{min} , x_{max} – parametrų minimali ir maksimali fiziologinės reikšmės.

2.2 lentelė

EKG parametrų fiziologinės ribos

Parametras	x_{min}	x_{max}
RR	150	1500
DQRS	30	140
DJT	50	400

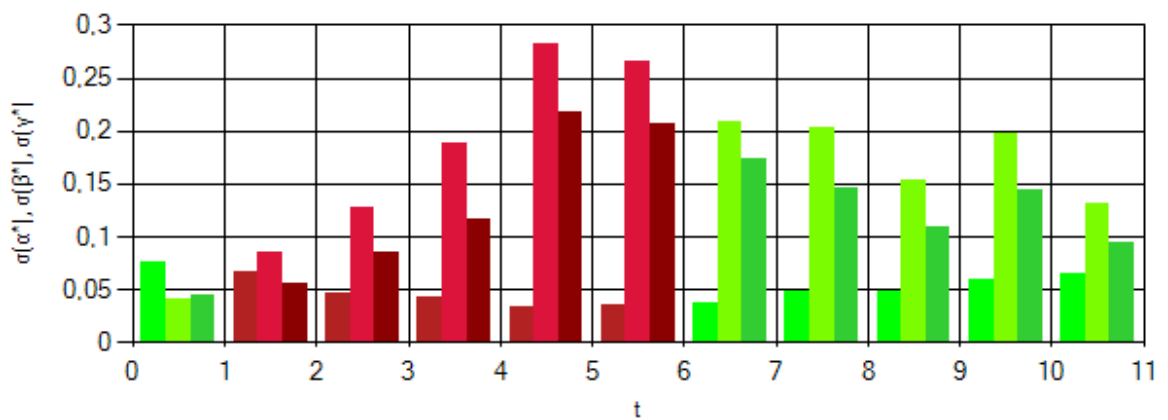
Sunormuotas parametrų reikšmes nagrinėsime naudodami sukurtą programinę įrangą (žr. 2.4 skyrelį), pirmiausiai sudarant Lagranžo skirtumų matricių seką (žr. 1.4 skyrelį) ir iš jų išskiriant matricių struktūrinius parametrus α^* , β^* , γ^* , α , β , γ .

Tiriant parametrų dinamines sąsajas, iš gautų struktūrinių parametrų, kiekvienos minutės laikotarpiu, apskaičiuojame jų standartinius nuokrypius:

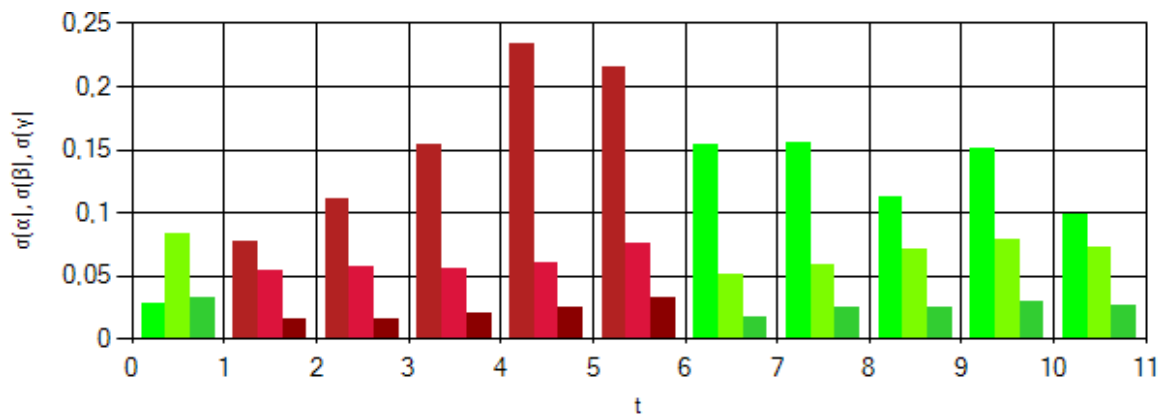
$$\sigma = \sqrt{\frac{\sum_{i=1}^n (x_i - \bar{x})^2}{n}}$$

Čia x_i – EKG parametro reikšmė i -tojo matavimo metu, \bar{x} – parametro reikšmių vidurkis, n – matavimų skaičius.

Iš dešimties tiriamų elektrokardiogramų apskaičiuojame struktūrinių parametrų standartinius nuokrypius, juos pateikiame stulpeline diagrama, kur žalsvi stulpeliai nurodo ramybės būseną, o rausvi – krūvį. Rezultatai pateikiami logaritminėje skalėje. Kiekvieno tirto asmens atskiri rezultatai pateikiami 1 priede, o šiame skyrelyje pateikiame suvidurkintus parametrų grafikus:



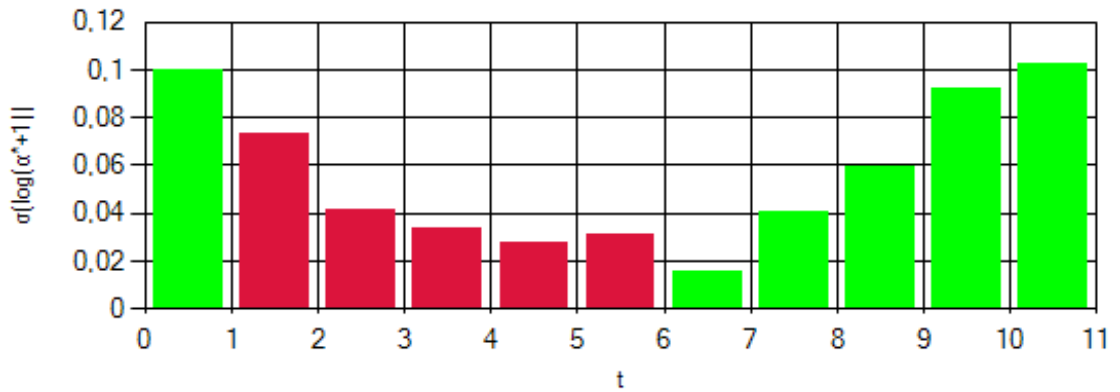
2.5 pav. Parametrų α^* , β^* , γ^* standartinių nuokrypių vidurkiai



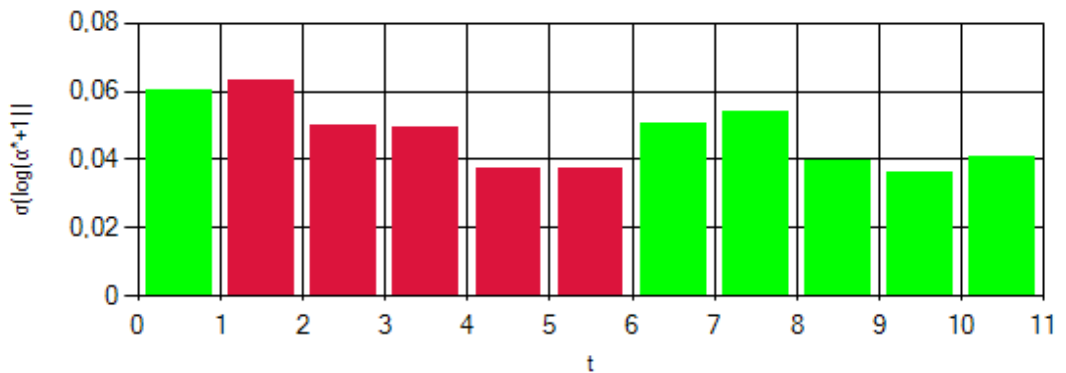
2.5 pav. Parametrų α , β , γ standartinių nuokrypių vidurkiai

Iš grafikų matome, kad rezultatai keičiasi, priklausomai nuo tiriamojo būsenos EKG duomenų gavimo metu. Taip pat pastebime, kad struktūrinių parametrų standartiniai nuokrypiai skiriasi tarpusavyje, todėl nagrinėsime kiekvieną iš jų atskirai.

Nagrinėjant parametą α^* , tiriamuosius išskirsime į dvi grupes: pirmai grupei priklausys 1, 4, 9 ir 10, o antrajai 2, 3, 5, 6, 7 ir 8 tiriamieji (žr. 1 priedą):



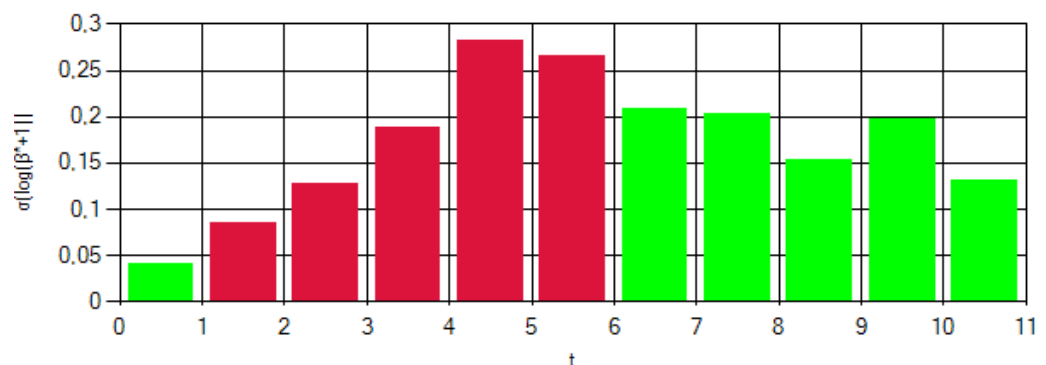
2.6 pav. Pirmosios grupės α^* parametro standartiniai nuokrypiai



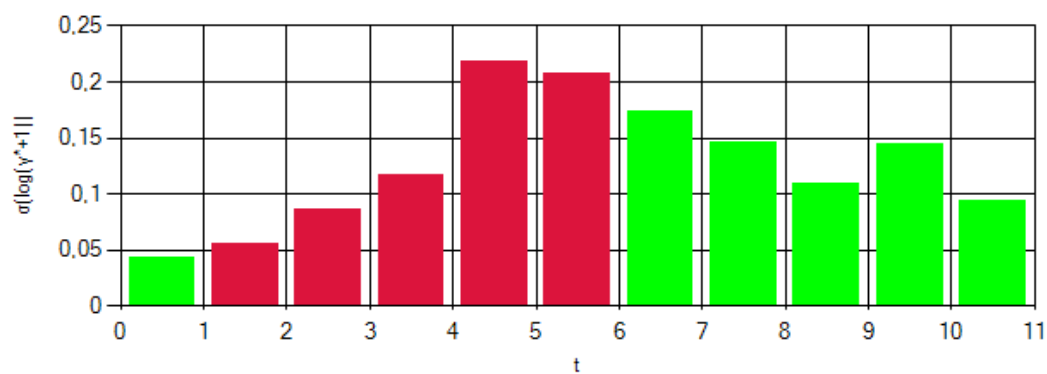
2.7 pav. Antrosios grupės α^* parametro standartiniai nuokrypiai

Pirmosios grupės parametų dinamika rodo, kad didėjant krūviui, sklaida apie vidurkį mažėja, o ramybės būsenoj ir atsigaunant – didėja. Antrosios grupės rezultatai atsigavimo laikotarpiu prastesni ir tai gali būti siejama su tuo, kad tiriamieji buvo pavargę EKG matavimo pradžioje.

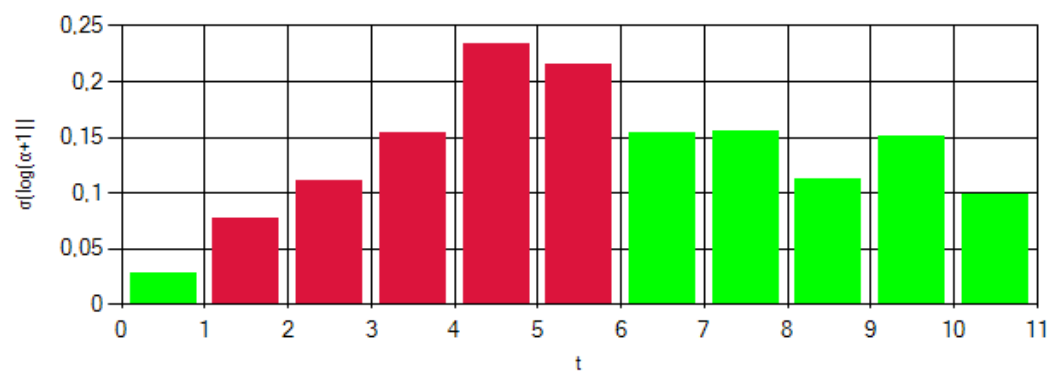
Likusiųjų struktūrinių parametų standartiniai nuokrypiai visiems tiriamiesiems:



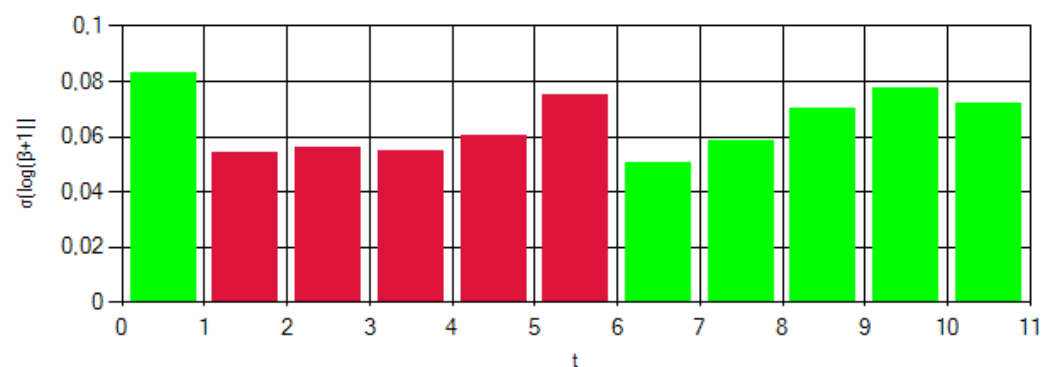
2.8 pav. Parametro β^* standartiniai nuokrypiai



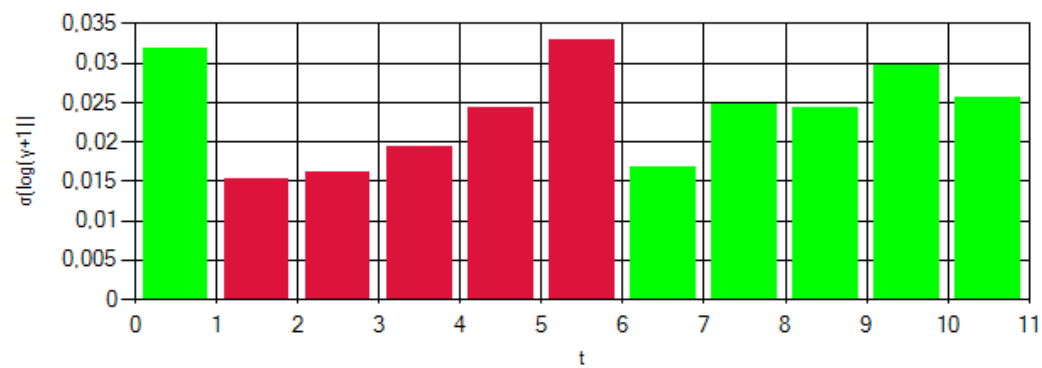
2.9 pav. Parametro γ^* standartiniai nuokrypiai



2.10 pav. Parametro α standartiniai nuokrypiai



2.11 pav. Parametro β standartiniai nuokrypiai



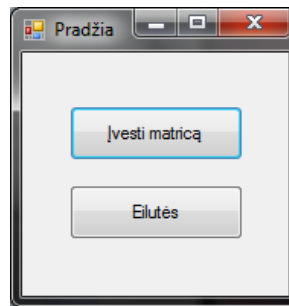
2.12 pav. Parametro γ standartiniai nuokrypiai

Tarpusavyje lyginant parametrus, gautus matricų suvedimo į trikampę metu $(\alpha^*, \beta^*, \gamma^*)$, su parametrais, gautais matricas verčiant diagonaliomis (α, β, γ) galime pastebėti, kad pastaroji parametų grupė atkartoja pirmąją, tik su priešinga dinamika, t.y. kai pirmosios grupės reikšmės mažėja, antrosios didėja.

Iš gautų grafikų matome, kad prie maksimalaus krūvio, gautos reikšmės staigiai krenta. Tai gali būti sietina su anaerobinio slenksčio pasiekimu, t.y. kai aerobinis kvėpavimas pereina į anaerobinį. Krūvio metu dauguma iš parametų $(\beta^*, \gamma^*, \alpha, \beta, \gamma)$ išreiškia panašų charakterį – jų sklaida apie vidurkį didėja, didėjant krūviui. Matome, kad pirmų dviejų ir paskutinių trijų minučių rezultatų dinamika stipriai skiriasi ir tai gali būti siejama su funkcinės būsenos kitimu per pirmąsias dvi minutes. Struktūriniai parametrai atkartoja fiziologines sportininkų būsenas ir kalba apie tos būsenos kitimo specifiką: jėgos, greičio bei ištvėmės sportuose.

2.4. PROGRAMINĖ REALIZACIJA IR INSTRUKCIJA VARTOTOJUI

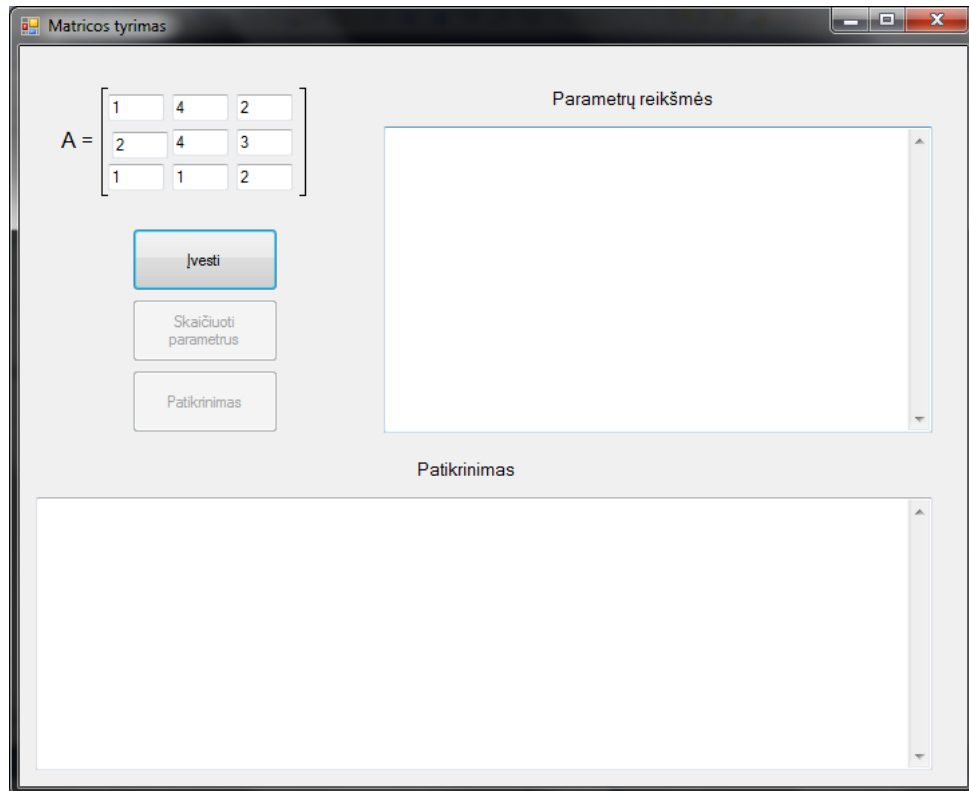
Sukurta programa, realizuojanti darbe aprašytus matematinius metodus ir jų pritaikymą elektrokardiogramos duomenims, grafiškai atvaizduojant rezultatus. Programinė realizacija sukurta Visual Basic programavimo kalba ir turi patogią vartotojo sąsają. Programa pritaikyta vienos matricos, sugeneruotų pagal tam tikrą funkciją bei įvestų EKG duomenų analizei. Programoje pirmiausiai atidaromas langas, kuriame galima pasirinkti ar norime analizuoti vieną matricą, ar iš eilučių sudarytų matricų seką:



2.13 pav. Pirmasis programos langas

Mygtukas „Įvesti matricą“ atidaro antrąjį programos langą (žr. 2.14 pav.), o mygtukas „Eilutės“ atidaro trečiąjį programos langą (žr. 2.15 pav.).

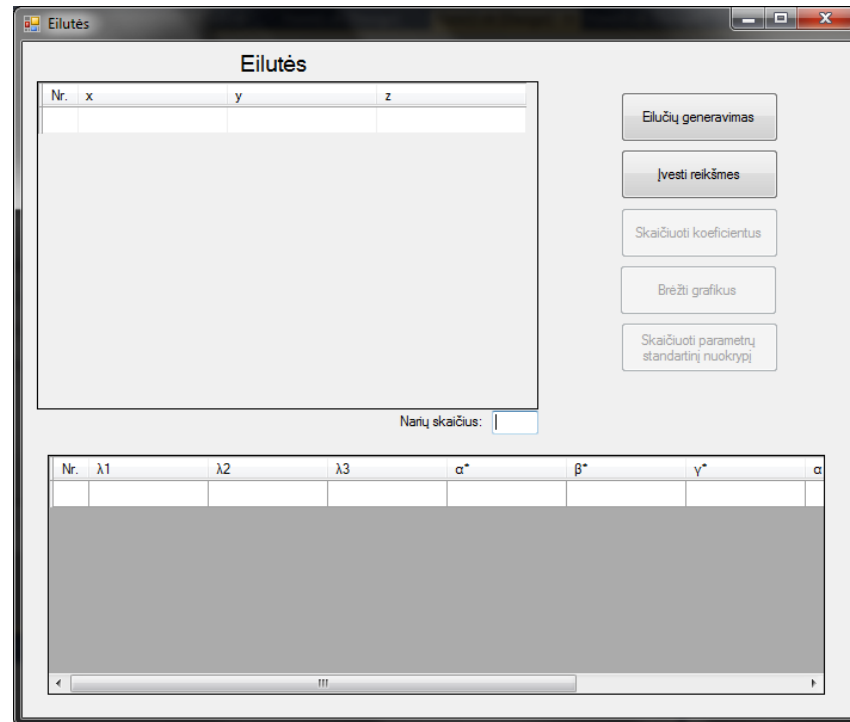
Antrasis programos langas:



2.14 pav. Antrasis programos langas

Šiame programos lange pirmiausia reikia įvesti trečios eilės kvadratinės matricos parametų reikšmes ir paspausti mygtuką „Įvesti“. Tada aktyvuojamas mygtukas „Skaičiuoti parametrus“, kuris pirmiausiai randa tikrines reikšmes pagal QR metodą, iš pradžių įvestą matricą perskaičiuojant į Hasenbergo matricą (algoritmas pritaikytas iš [16]). Tikrinės reikšmės surikiuojamos ir tada atliekami skaičiavimai, aprašyti 1.9, 1.10 ir 1.11 skyreliuose. Apskaičiuoti parametrai bei tarpinės matricos išspausdinamos tekstiniame lauke. Atlikus šiuos skaičiavimus, aktyvuojamas mygtukas „patikrinimas“, kuris pagal gautas tarpines matricas atkuria pradinę matricą bei apskaičiuoja gautas paklaidas. Visi rezultatai išspausdinami patikrinimo tekstiniame lauke.

Langas, skirtas eilučių analizei, atrodo taip:



2.15 pav. Trečiasis programos langas

Šiame programos lange pirmiausiai reikia gauti tris eilutes, iš kurių bus sudaroma trečios eilės kvadratinė matricų seka. Eilutės galima gauti vieno iš dviejų aktyvių mygtukų pagalba. Mygtukas „Eilučių generavimas“ atidaro ketvirtąjį programos langą (žr. 2.16 pav.), o „Įvesti“ reikšmes mygtukas atidaro penktąjį programos langą (žr. 2.17 pav.). Tada gautos reikšmės surašomos į lentelę, randama Lagranžo skirtumų matricų seka ir aktyvuojamas mygtukas „Skaičiuoti koeficientus“, kuris randa tikrines reikšmes bei matricų algebrinius–struktūrinius parametrus pagal darbe aprašytus metodus. Gautos parametų reikšmės surašomos į lentelę ir aktyvuojami likę du mygtukai. „Brėžti grafikus“ mygtukas atidaro šeštąjį programos langą (žr. 2.18 pav.), o mygtukas „Skaičiuoti parametų standartinį nuokrypį“ atidaro vienuoliktąjį programos langą (žr. 2.23 pav.). Programos langas, skirtas eilučių generavimui, atrodo taip:

Eilučių generavimas pagal šias funkcijas:

$$x = \cos(\rho_1 t + \varphi_1)$$

$$y = \cos(\rho_2 t + \varphi_2)$$

$$z = \cos(\rho_3 t + \varphi_3)$$

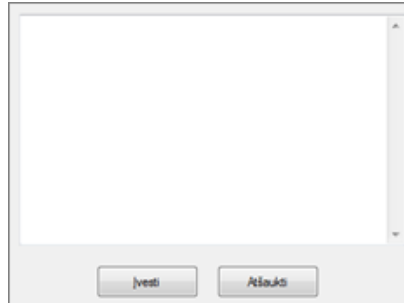
Koeficientų reikšmės:

$\rho_1 =$ $\varphi_1 =$ Narių skaičius:
 $\rho_2 =$ $\varphi_2 =$ Žingsnis:
 $\rho_3 =$ $\varphi_3 =$

2.16 pav. Ketvirtasis programos langas

Šiame lange galima pasirinkti pateiktų funkcijų parametrų reikšmes: dažnius ρ_1, ρ_2, ρ_3 , fazinius postūmius $\varphi_1, \varphi_2, \varphi_3$, eilutės narių skaičių, bei laiko intervalo žingsnį. Mygtukas „Generuoti x, y, z “ apskaičiuoja eilutės narius pagal pateiktas funkcijas, o mygtukas „Atšaukti“ uždaro šį programos langą.

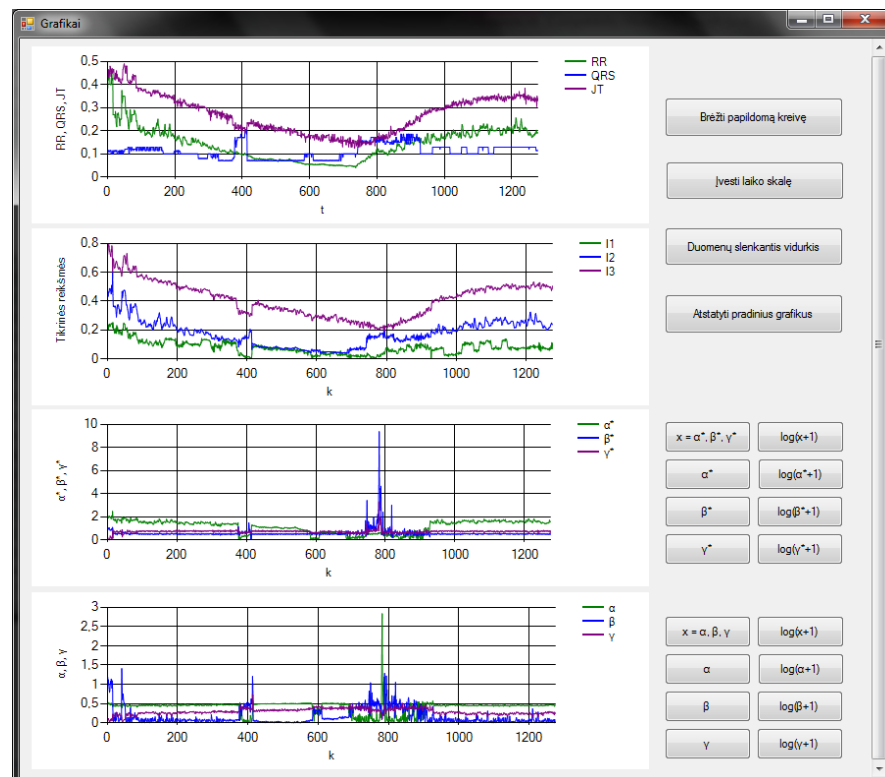
EKG duomenims įvesti naudojamas šis programos langas:



2.17 pav. Penktasis programos langas

Į šiame lange pateiktą tekstinį lauką įvedamos ranka arba įkopijuojamos (pavyzdžiui iš Microsoft Excel lentelės) EKG reikšmių eilutės. Mygtuku „Įvesti“ reikšmės įkeliamos į trečiojo programos lango (žr. 2.15 pav.) lentelę ir apskaičiuojama Lagranžo skirtumų matricių seka, o mygtuku „Atšaukti“ uždaromas šis programos langas ir reikšmės neįvedamos.

Grafiniam EKG duomenų bei gautų parametrų nagrinėjimui skirtas langas:



2.18 pav. Šeštasis programos langas

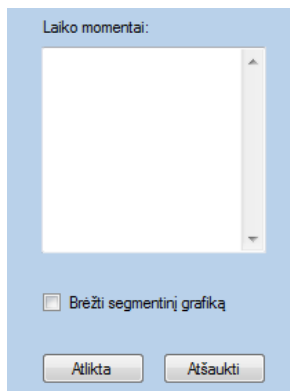
Atidarius pavaizduotą programos langą, automatiškai nubrėžiami keturi grafikai: įvesti elektrokardiogramos parametrai, Lagranžo skirtumų matricų tikrinės reikšmės bei algebrinių- struktūrinių parametrų α^* , β^* , γ^* ir α , β , γ reikšmės. Mygtukai „Brėžti papildomą kreivę“, „Įvesti laiko skalę“ ir „Duomenų slenkantis vidurkis“ atitinkamai atidaro septintąjį (žr. 2.19 pav.), aštuntąjį (žr. 2.20 pav.) ir devintąjį (žr. 2.21 pav.) programos langus. Mygtuko „Atstatyti pradinis grafikus“ pagalba gaunami grafikai, kurie buvo automatiškai nubrėžti atidarius šį langą. Trečiojo ir ketvirtojo grafiko dešinėje pusėje yra po 8 mygtukus, kurių pagalba grafikuose brėžiami atitinkami parametrai kartu arba po vieną, atvaizduojant jų realias arba logaritmuotas reikšmes.

Prie pasirinktų grafikų šeštajame programos lange (žr. 2.18 pav.) norint atvaizduoti papildomą kreivę, naudojamas šis programos langas:

2.19 pav. Septintasis programos langas

Pavaizduotame programos lange galima apskaičiuoti vieno iš trijų EKG parametrų slenkantį standartinį nuokrypį naudojantis mygtukų „ $\sigma(RR)$ “, „ $\sigma(QRS)$ “ ir „ $\sigma(JT)$ “ pagalba. Apskaičiuotos reikšmės pateikiamos tekstiniame programos lango lauke. Į pateiktą tekstinį lauką galima suvesti ir kitos norimos kreivės reikšmes. Tada pasirinkus, kuriuose iš keturių grafikų brėžti norimą kreivę spaudžiamas vienas iš trijų mygtukų: „Brėžti kreivę“ mygtukas atvaizduoja reikšmes pasirinktuose grafikuose ir uždaro programos langą, mygtukas „Ištrinti kreivę“ pašalina iš grafikų papildomą kreivę, jeigu tokia jau buvo nubrėžta bei ištrina reikšmes iš tekstinio lauko, o mygtukas „Atšaukti“ uždaro programos langą.

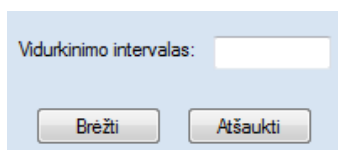
Norit pakeisti matavimo skalę abscisių ašyje, atidaromas šis programos langas:



2.20 pav. Aštuntasis programos langas

Į šiame lange esantį tekstinį lauką suvedamos reikšmės, nusakančios suvestų EKG parametrų matavimų laiko momentus. Kai suvedamos reikšmės, spaudžiamas mygtukas „Atlikta“ ir gaunami grafikai, kuriuose parametrai atvaizduojami jau pagal laiko momentus. Norint, kad grafikuose būtų atvaizduotas reikšmių vidurkis kiekvienos minutės laikotarpiu, reikia pažymėti varnelę, prie užrašo „Brėžti segmentinį grafiką“ - tokiu būdu gaunamas stulpelinis grafikas. Mygtukas „Atšaukti“ uždaro programos langą.

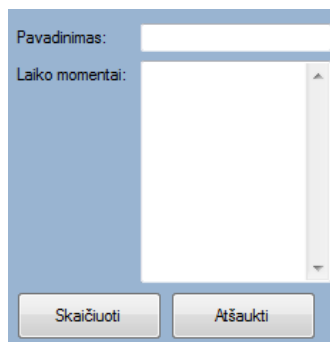
Norint grafikuose atvaizduoti parametrų slenkantį vidurkį, naudojamas šis programos langas:



2.21 pav. Devintasis programos langas

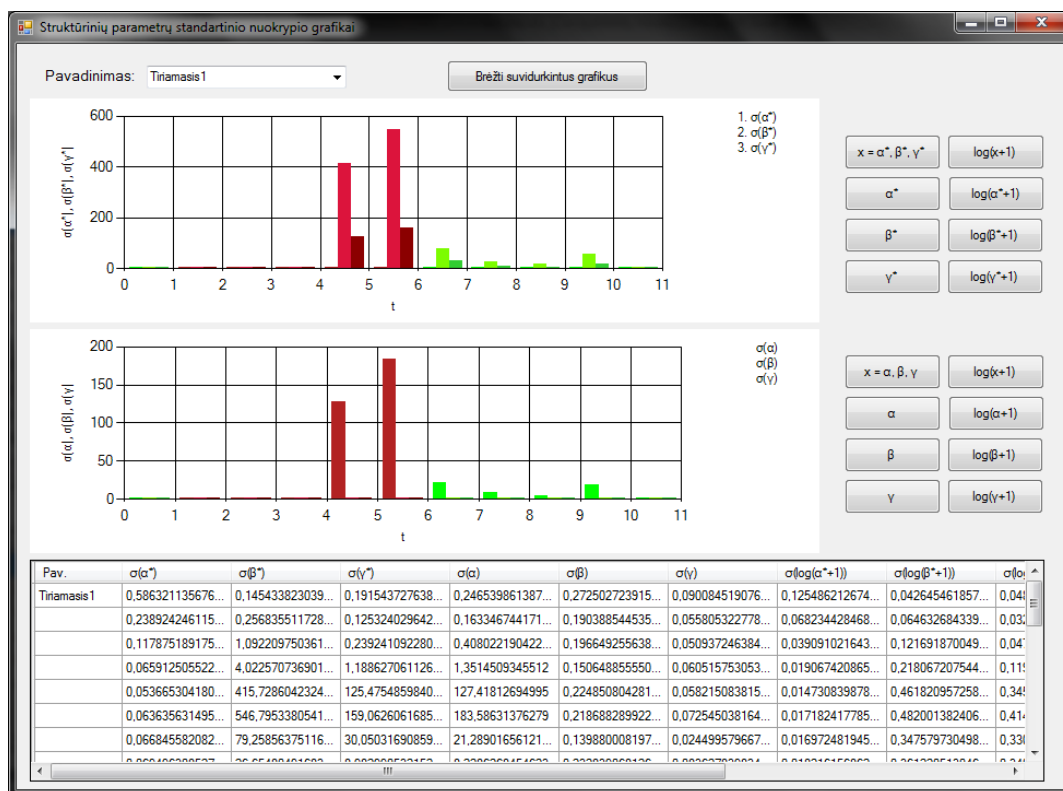
Pateikto programos lango tekstiniame lauke įvedamas vidurkinimo intervalas ir pagal jo reikšmę, mygtuko „Brėžti“ paspaudimu, grafikuose atvaizduojamas parametrų slenkantis vidurkis. Jeigu atidarius šį programos langą nusprendžiama tokių grafikų nebrėžti, spaudžiamas mygtukas „Atšaukti“.

Norint apskaičiuoti iš elektrokardiogramos duomenų gautų parametrų reikšmių standartinius nuokrypius, naudojamas šis programos langas:



2.22 pav. Dešimtas programos langas

Kadangi kiekvieną kartą įvedus naujus EKG duomenis ir iš jų gautų parametų apskaičiuvus standartinius nuokrypius, reikšmės išsaugomos palyginimo bei vidurkio ieškojimo tikslais, todėl šiame lange įvedamas norimas pavadinimas tuo metu tiriamiems duomenims (pavyzdžiui tiriamojo vardas ar numeris). Į sekantį tekstinį lauką turi būti įvedamos reikšmės, atitinkančios suvestų EKG parametų laiko momentus. Mygtuko „Atšaukti“ pagalba išjungiamas šis programos langas ir grįžtama į prieš tai buvusį programos langą (žr. 2.15 pav.), mygtuku „Skaičiuoti“ paleidžiamas vienuoliktasis programos langas:



2.23 pav. Vienuoliktasis programos langas

Atidarius šį programos langą, automatiškai nubrėžiami struktūrinių parametų standartinių nuokrypių grafikai, kiekvienos minutės laikotarpiu. Programos lango viršuje yra iškrentantis sąrašas su jau įvestų duomenų pavadinimais. Paspaudus ant atitinkamą pavadinimą turinčio laukelio, nubrėžiami pasirinkti duomenys. Mygtukas „Brėžti suvidurkintus grafikus“ apskaičiuoja visų įvestų tiriamųjų apskaičiuotų parametų vidurkį kiekvienos minutės laikotarpyje ir atvaizduoja jį grafikuose. Kiekvieno grafiko dešinėje pusėje yra po 8 mygtukus, kurių pagalba grafikuose brėžiami atitinkamų parametų standartiniai nuokrypiai, kartu arba po vieną, atvaizduojant jų realias arba logaritmuotas reikšmes. Programos lango apačioje yra lentelė su visomis apskaičiuotų standartinių nuokrypių ir jų logaritmų reikšmėmis. Lentelėje paėliui rodoma kiekvieno tiriamojo reikšmės, tokia tvarka, kokia buvo įvesti duomenys. Jeigu buvo apskaičiuotas visų tiriamųjų standartinių nuokrypių vidurkis, lentelės apačioje išspausdinami gauti rezultatai.

IŠVADOS

1. Algebrinės-struktūrinės matricų analizės pagalba atlikus trečios eilės matricų transformacijas, gavome parametrus, atspindinčius signalų dinamines sąsajas.
2. Tiriant sinchronines eilutes, struktūrinių parametrų moduliai didžiausias reikšmes įgyja, kai funkcijos viena prie kitos artėja. Taip pat pastebėjome, kad kuo didesnis atstumas tarp funkcijų, tuo pastovesnės struktūrinių parametrų modulių reikšmės.
3. Tarpusavyje lyginant parametrus, gautus matricų suvedimo į trikampę matricą metu $(\alpha^*, \beta^*, \gamma^*)$, su parametrais, gautais matricas verčiant diagonaliomis (α, β, γ) , pastebėjome, kad pastaroji parametrų grupė atkartoja pirmąją, tik su priešinga dinamika, kai pirmosios grupės reikšmės mažėja.
4. Tiriant EKG duomenis prie maksimalaus krūvio, gautos reikšmės staigiai krenta. Tai gali būti sietina su anaerobinio slenksčio pasiekimu.
5. Krūvio metu dauguma struktūrinių parametrų išreiškia panašų charakterį – jų sklaida apie vidurkį didėja, didėjant krūviui.
6. Iš elektrokardiogramos duomenų gauti algebriniai-struktūriniai parametrai bei jų sklaida apie vidurkį atkartoja fiziologines sportininkų būsenas ir apibūdina būsenų kitimo specifiką jėgos, greitumo bei ištvermės sportuose.

PADĖKOS

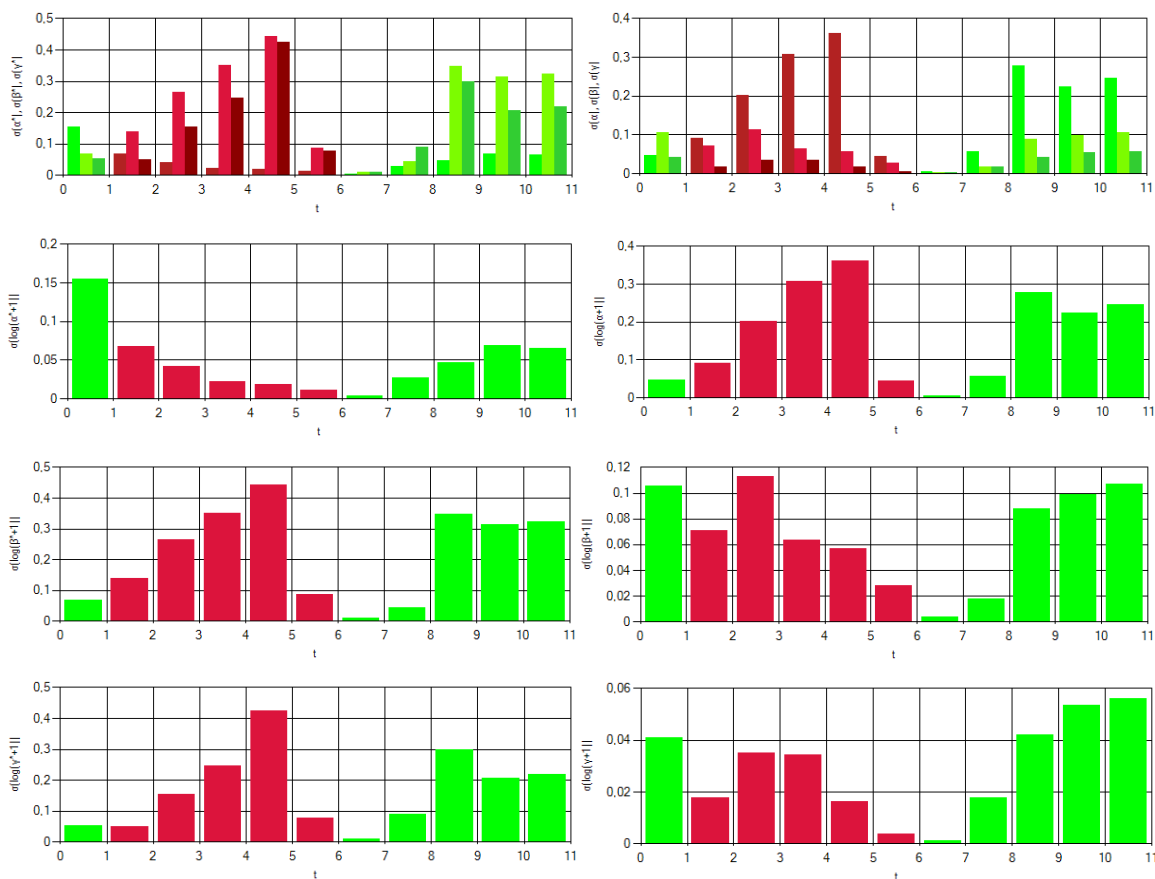
Dėkoju Kauno technologijos universiteto profesoriui Zenonui Navickui už patarimus, matematinės žinias ir pagalbą įsisavinant algebrinės-struktūrinės analizės metodus. Taip pat dėkoju Lietuvos sveikatos mokslų universiteto profesoriui Alfonsui Vainorui už pagalbą gaunant elektrokardiogramų matavimų duomenis bei konsultavimą medicinos tematika, sprendžiant kardiologinius uždavinius.

LITERATŪRA

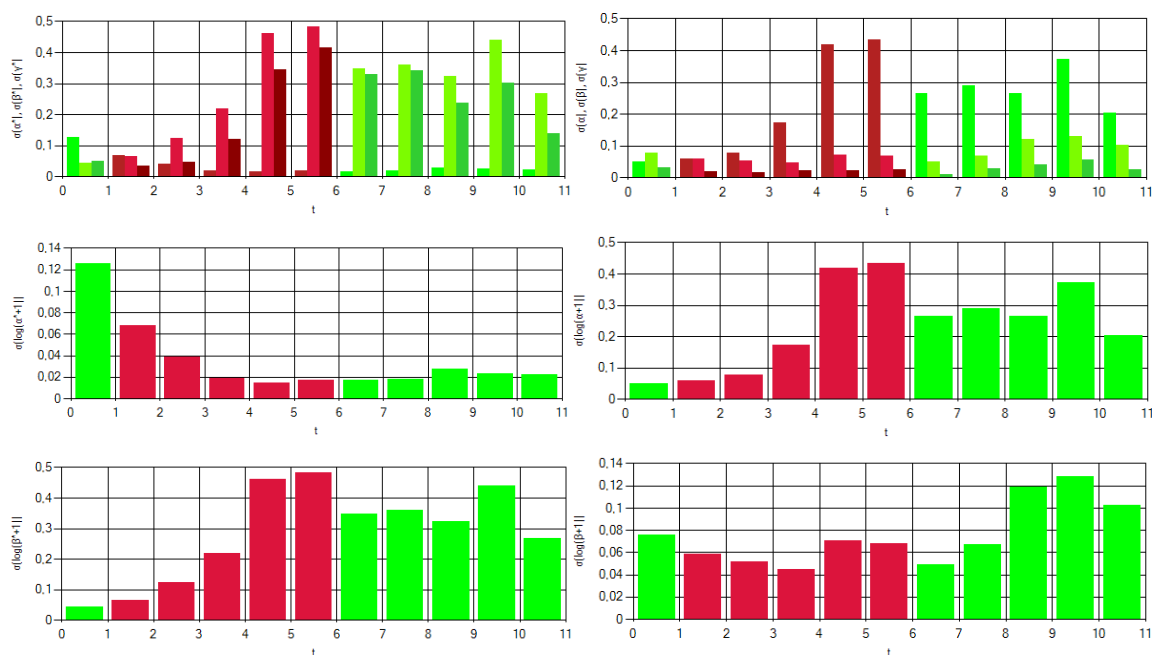
1. Ashley EA, Niebauer J. *Cardiology Explained*. London: Remedica, 2004, p. 19.
2. Bernstein D. S. *Matrix Mathematics*. New Jersey, Princeton University Press, 2009.
3. Berškienė K. Elektrokardiografinių signalų parametrų dinaminių sąsajų analizė // Daktaro disertacija: technologijos mokslai, elektros ir elektronikos inžinerija. Kauno technologijos universitetas, Kaunas, 2010.
4. Birko G., MacLane S. *A Survey of Modern Algebra*, A K Peters, Wellesley, MA, 1996.
5. *Complex Systems Science in Biomedicine: Topics in Biomedical Engineering: International Book Series*. Ed. Deisboeck, T. S; Kresh, J. Y. Springer, 2006.
6. Horn, Roger A.; Johnson, Charles R. *Matrix Analysis*. Cambridge University Press, 1990.
7. Kresh, J. Y.; Izrailyan, I., Wechsler A.S. *The heart as a Complex Adaptive System* [interaktyvus] [žiūrėta 2014-05-10]. Prieiga per internetą:
http://www.pages.drexel.edu/~jk45/Heart_CAS/heart_co.htm
8. Markienė, Z. O. Elektrokardiografija [interaktyvus] [žiūrėta 2014-05-11]. Prieiga per internetą:
http://ekg.lt/index.php?id=3_0
9. Markienė, Z. O. *Klinikinė elektrokardiografija: monografija*. 2000.
10. Miller, J. H.; Page, E.S. *Complex Adaptive Systems: An Introduction to Computational Models of Social Life*. 2007.
11. Myers, W. T. *Anatomy trains*. Edinburgh, Churchill Livingstone, 2001.
12. Navickas Z., Kriaunavičienė A., Vainoras A., Analysis relationship between some electrocardiographic parameters. // *Matematika ir matematinis modeliavimas / Kauno Technologijos Universitetas*. Kaunas: Technologija. 2012, t. 7-8, p. 1-6.
13. Navickas Z., Ragulskis M., Vainoras A., Šmidaitė R., The explosive divergence in iterative maps of matrices. // *Communications in Nonlinear Science and Numerical Simulation*. Amsterdam: Elsevier Science. ISSN 1007-5704, t. 17, p. 4430-4438.
14. Navickas Z., Šmidaitė R., Vainoras A., Ragulskis M., The logistic maps of matrices. // *Discrete and Continuous Dynamical systems – Series B*. Springfield : American Institute of Mathematical Sciences. ISSN 1531-3492. 2011, t. 16, nr. 3, p 927-944.
15. Petrulionienė Ž. *Koronarinė širdies liga. Rizikos veiksniai, klinikiniai simptomai ir gydymas*. Vilnius: UAB "Vaistų žinios", 2010, p. 5.
16. Plukas K. *Skaitiniai metodai ir algoritmai: vadovėlis aukštųjų mokyklų studentams*. Kaunas, 2001. p. 244-259.

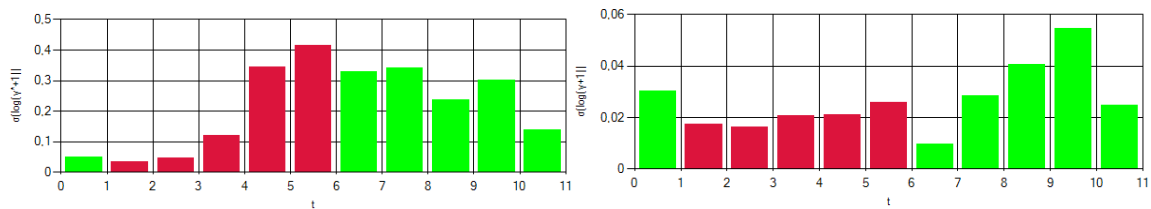
IŠ EKG DUOMENŲ GAUTŲ STRUKTŪRINIŲ PARAMETRŲ STANDARTINIŲ NUOKRYPIŲ GRAFIKAI

Tiriamasis nr.1

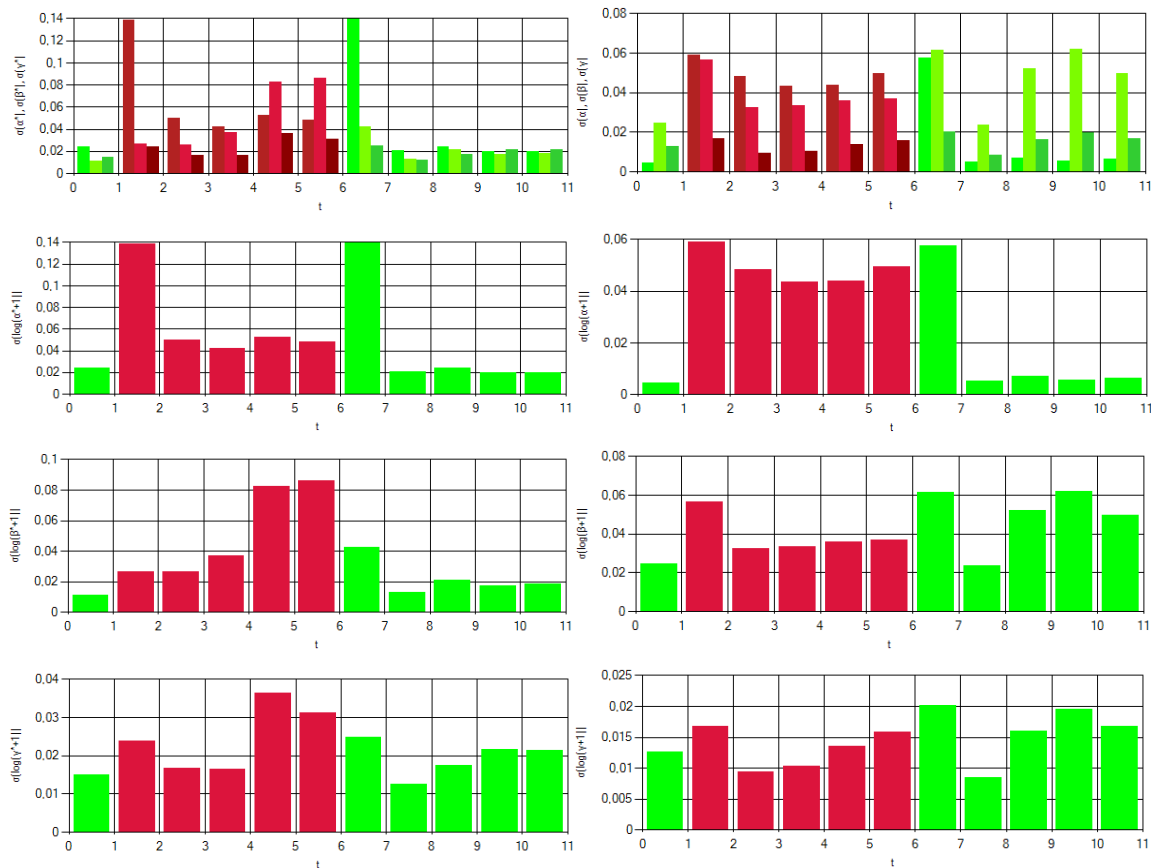


Tiriamasis nr. 2

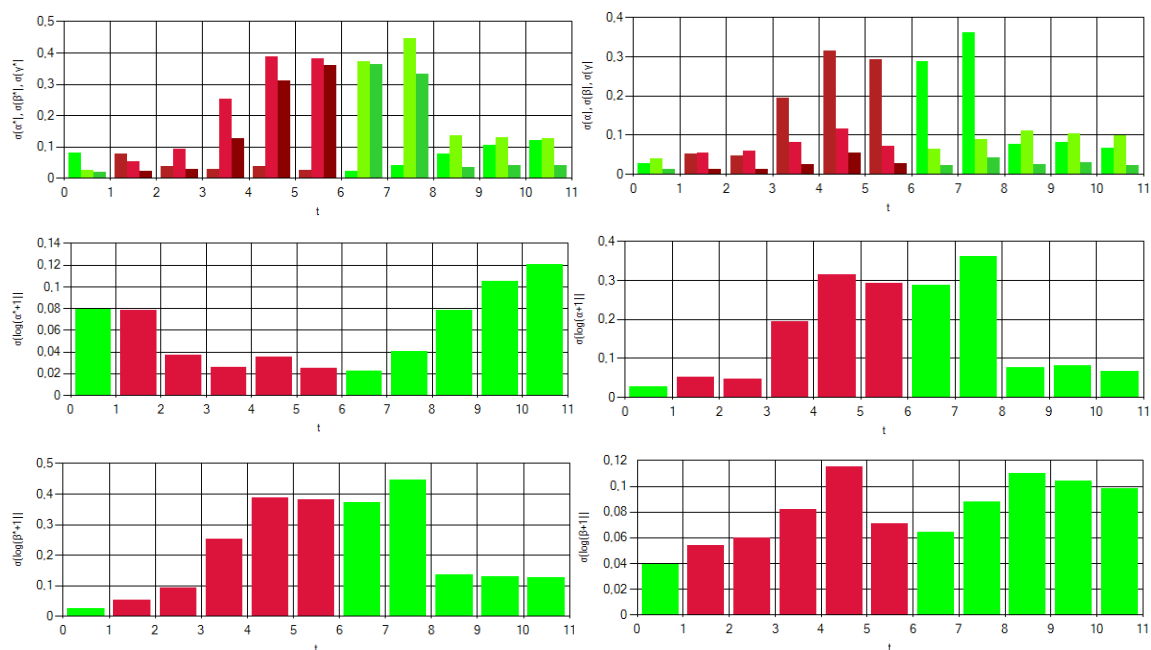


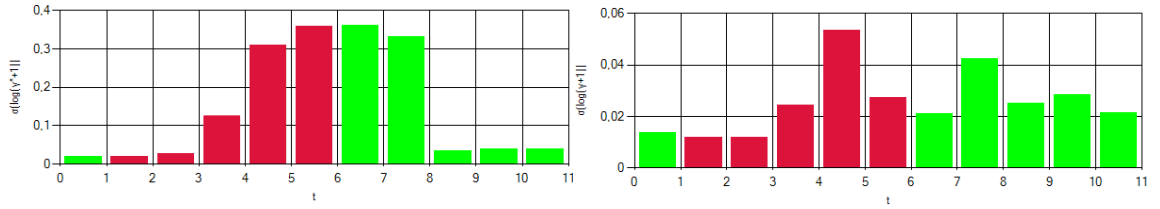


Tiriamasis nr. 3

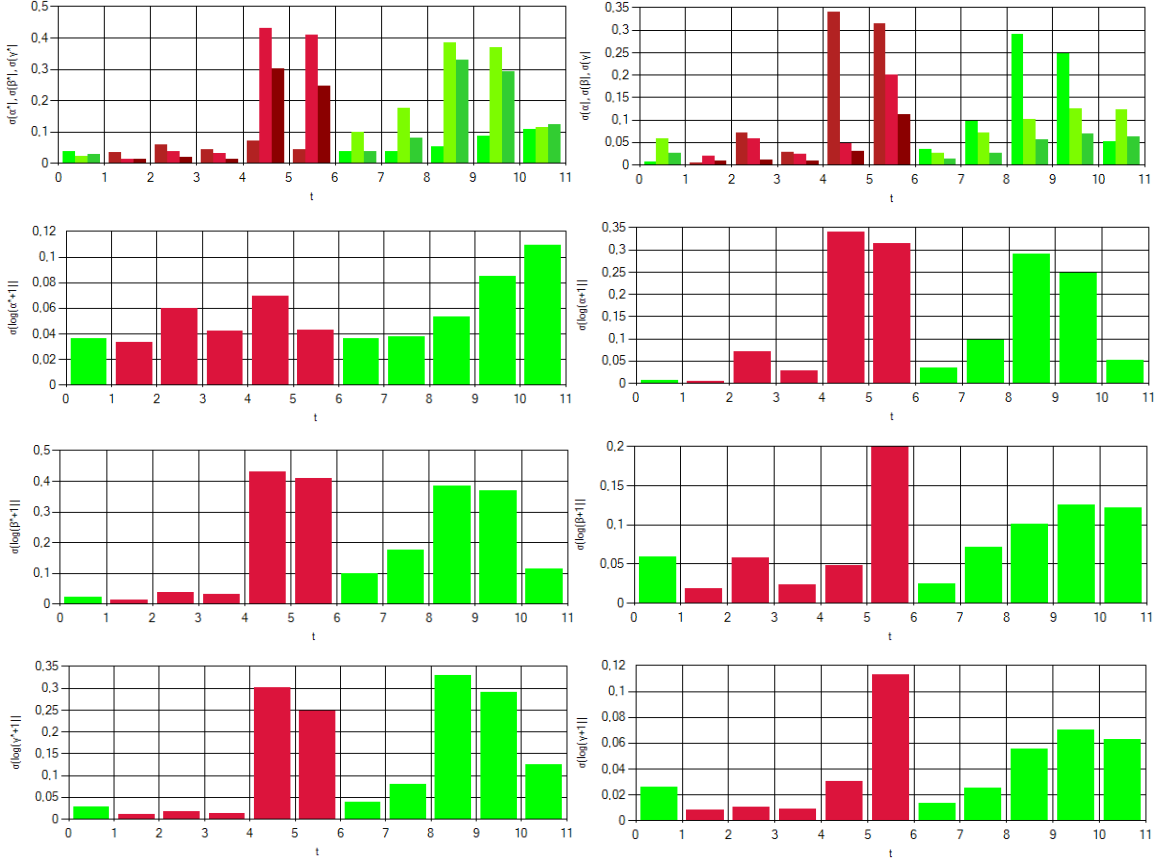


Tiriamasis nr. 4

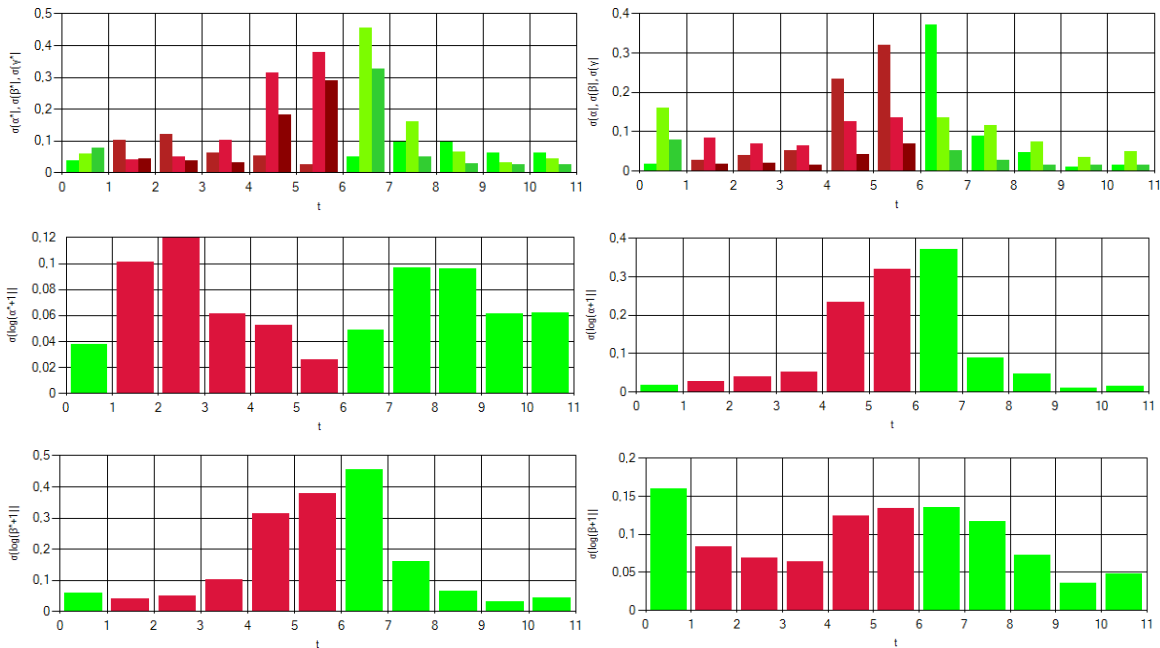


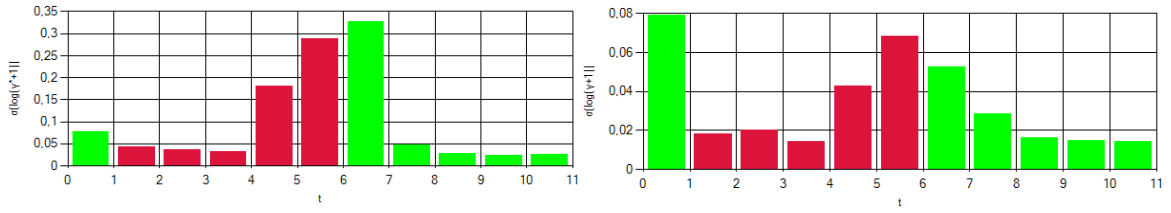


Tiriamasis nr. 5

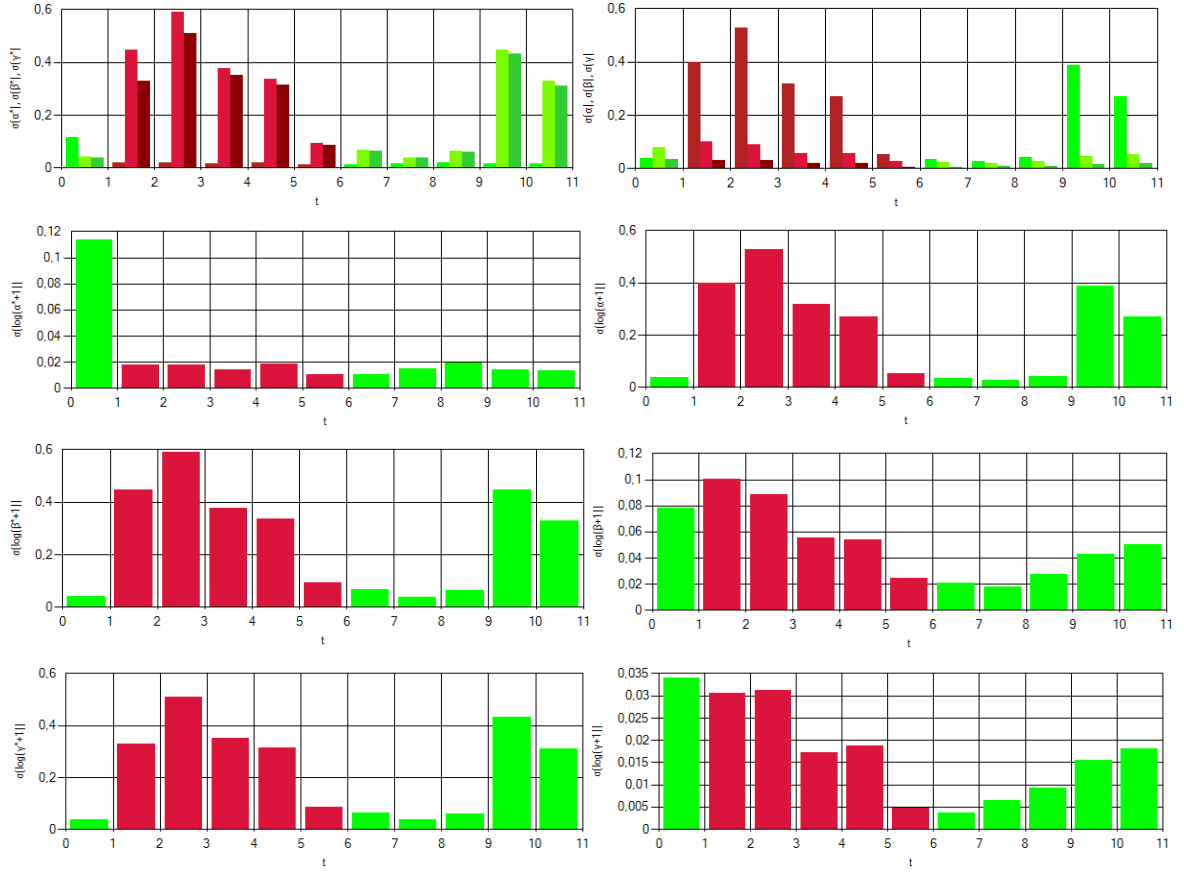


Tiriamasis nr. 6

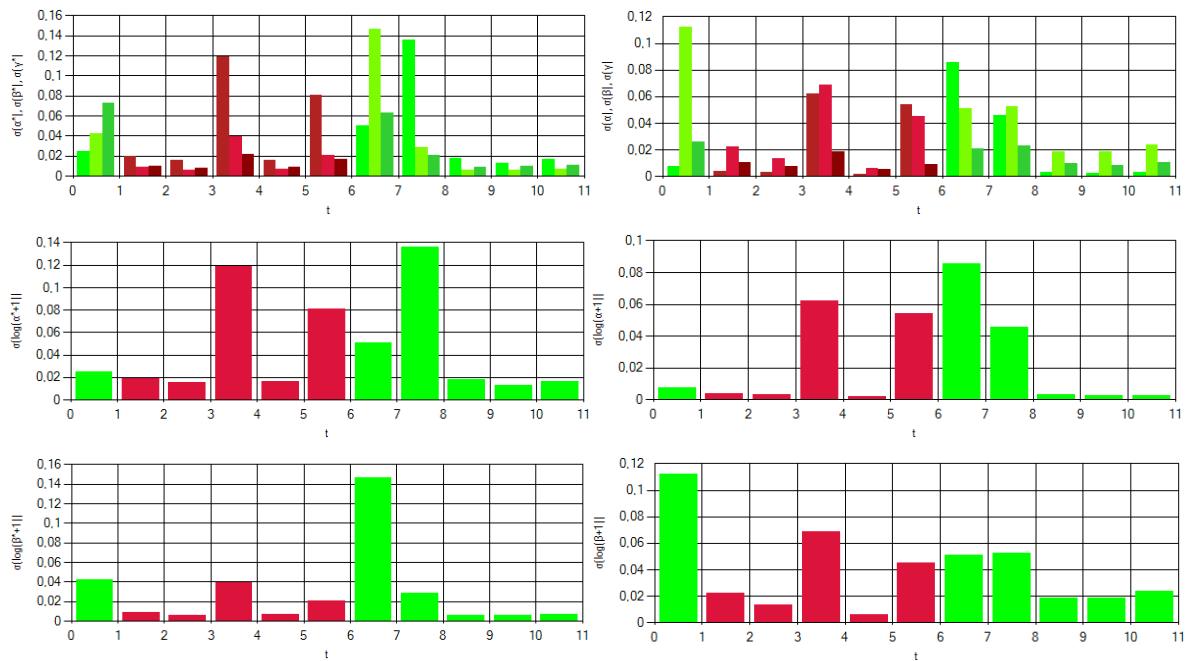


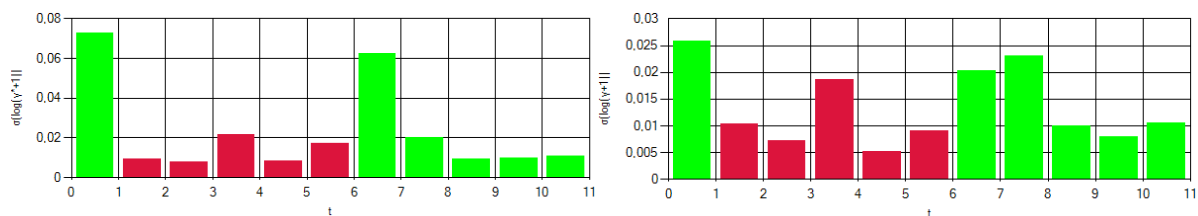


Tiriamasis nr. 7

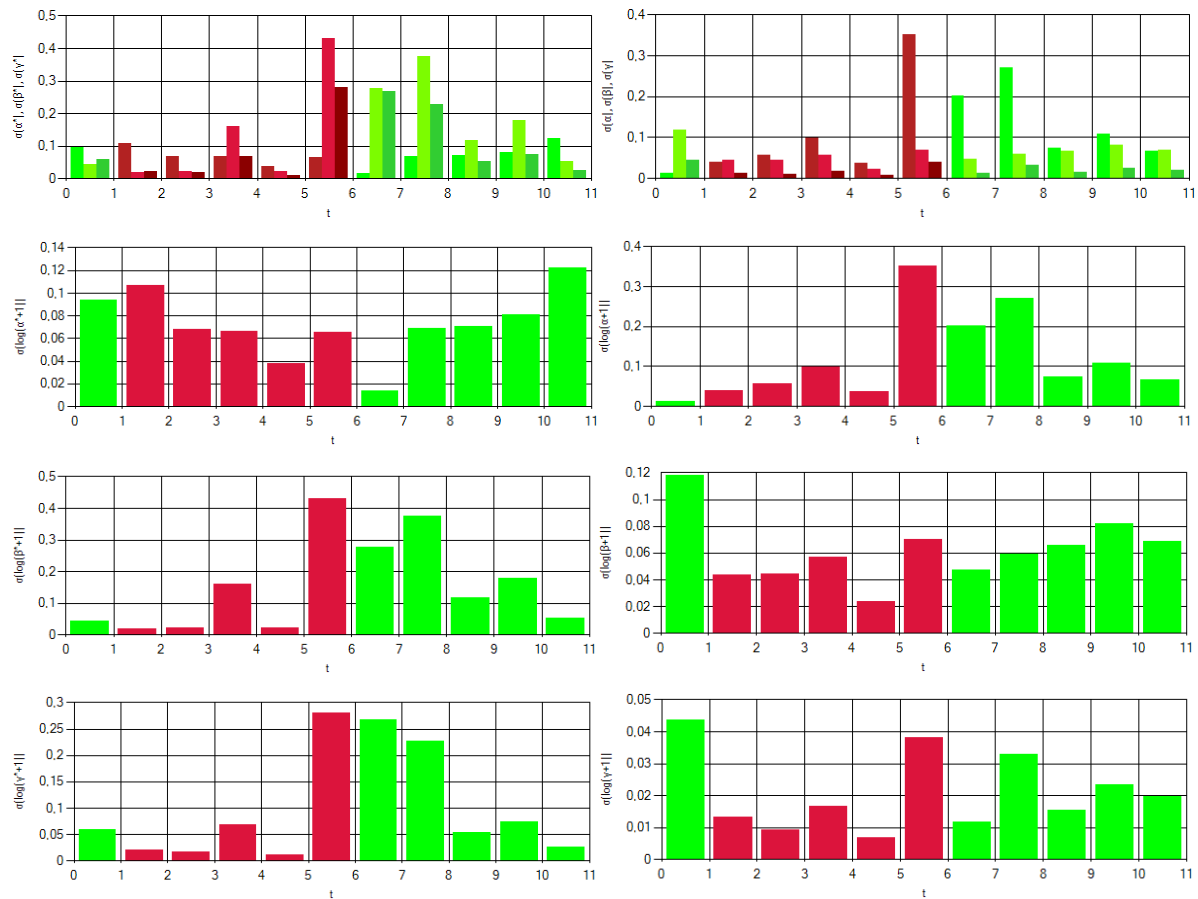


Tiriamasis nr. 8

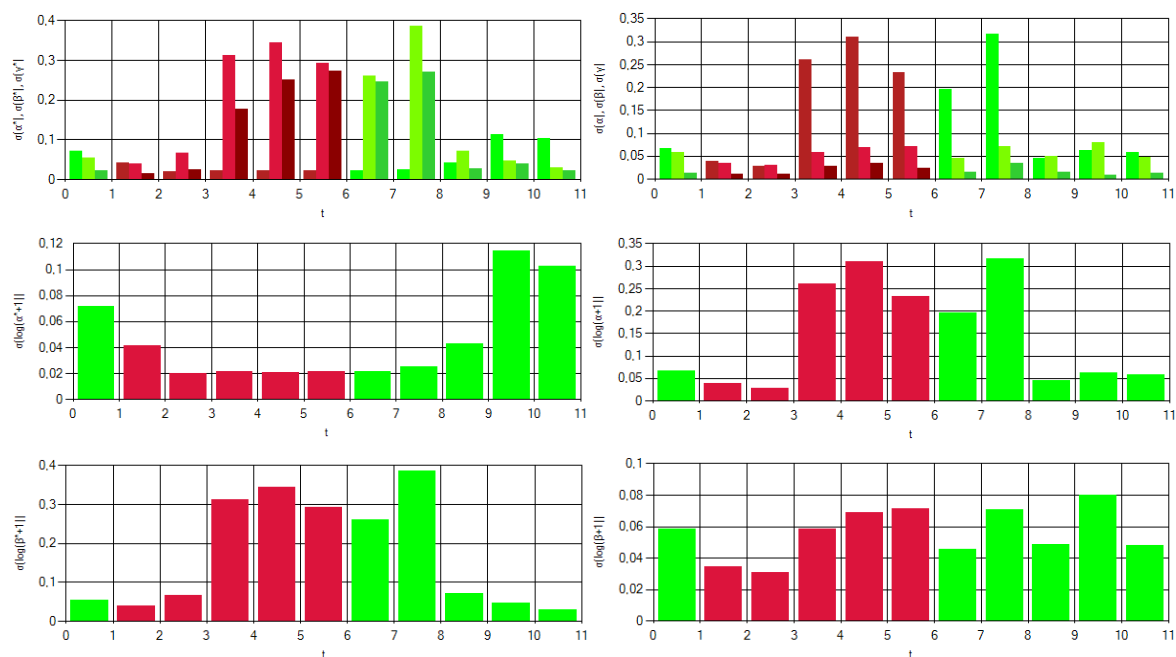


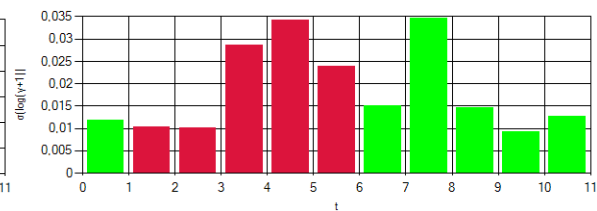
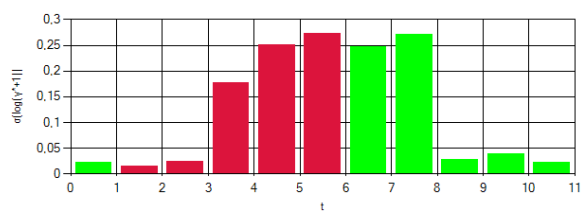


Tiriamasis nr. 9

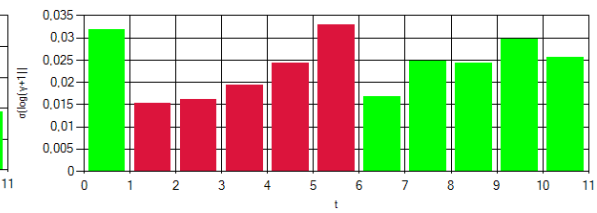
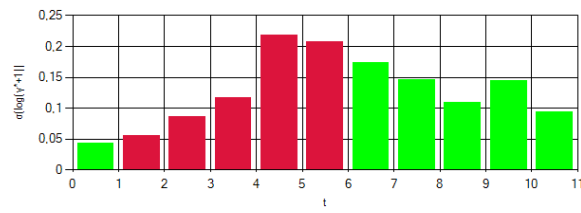
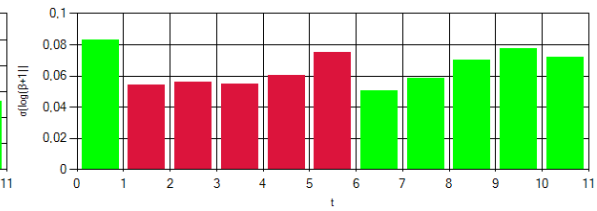
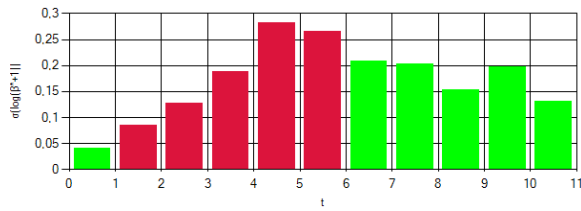
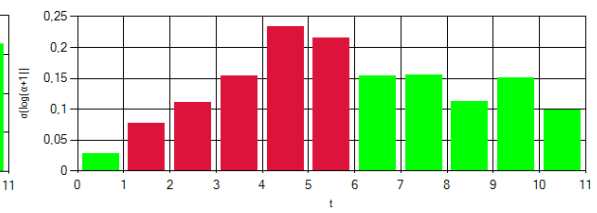
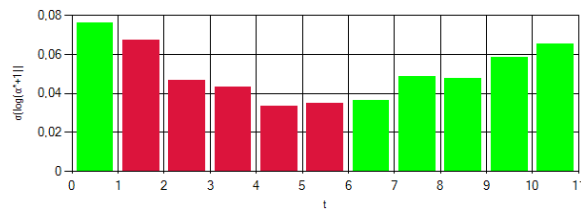
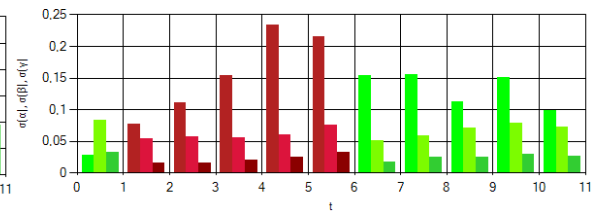
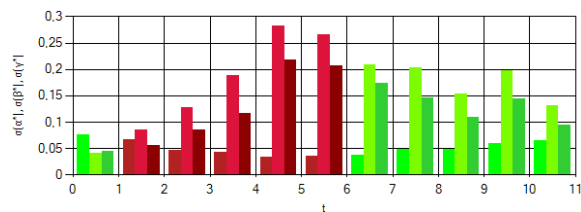


Tiriamasis nr. 10





Suvidurkinti rezultatai



PROGRAMOS KODAS

```

Public Class pradine
    Dim mk As Integer
    Public Function Tikrines(ByVal mk, ByVal matr) As Double(,,)
        Dim n As Integer
        Dim wr(2) As Double
        Dim wi(2) As Double
        Dim iter(2) As Integer
        Dim m As Integer
        Dim it As Integer
        Dim p As Double
        Dim q As Double
        Dim r As Double
        Dim s As Double
        Dim xt As Double
        Dim yt As Double
        Dim zt As Double
        Dim eps As Double
        Dim t1 As Double
        Dim t2 As Double
        Dim machepts As Double
        Dim rksm(mk, 2, 1) As Double
        Dim h(mk, 2, 2) As Double

        For ss = 0 To mk
            n = 3
            For i = 0 To 2
                For j = 0 To 2
                    h(ss, i, j) = matr(ss, i, j)
                Next
            Next
            'Pasiverčiam matricą į Hasenbergo viršutinę matricą
            Dim dgk As Double
            Dim g As Double

            If Math.Abs(h(ss, 1, 0)) < Math.Abs(h(ss, 2, 0)) Then
                For i = 0 To 2
                    g = h(ss, 1, i)
                    h(ss, 1, i) = h(ss, 2, i)
                    h(ss, 2, i) = g
                Next
                For i = 0 To 2
                    g = h(ss, i, 1)
                    h(ss, i, 1) = h(ss, i, 2)
                    h(ss, i, 2) = g
                Next
            End If
            If h(ss, 1, 0) <> 0 Then
                dgk = h(ss, 2, 0) / h(ss, 1, 0)
                For i = 0 To 2
                    h(ss, 2, i) = h(ss, 2, i) - h(ss, 1, i) * dgk
                Next
                For i = 0 To 2
                    h(ss, i, 1) = h(ss, i, 1) + h(ss, i, 2) * dgk
                Next
            End If

            'mašininio epsilon apskaičiavimas
            machepts = 1
            While 1 + machepts > 1
                machepts = machepts / 2
            End While
            machepts = machepts * 2
        Next
    End Function
End Class

```

```

For i = 0 To 2
    wr(i) = 0
    wi(i) = 0
Next

While n > 0
    it = 0
    If n > 2 Then
        'skaičiavimo tikslumo nustatymas
        eps = macheps * (Math.Abs(h(ss, 1, 1)) + Math.Abs(h(ss, 2, 2)))
        While ((it < 30) And (Math.Abs(h(ss, 2, 1)) > eps) And (Math.Abs(h(ss, 1, 0)) > eps))
            'dvigubas qr metodo žingsnis
            it = it + 1
            'matricos Q*H*Q perskaičiavimas į viršutinę Hasenbergo matricą
            h(ss, 2, 0) = 0
            For l = 1 To n - 1
                If l = 1 Then
                    If (it = 10) Or (it = 20) Then
                        t1 = Math.Abs(h(ss, n - 1, n - 2)) + Math.Abs(h(ss, n - 2, n - 3))
                        p = (h(ss, 0, 0) * h(ss, 0, 0) - 1.5 * t1 * h(ss, 0, 0)
                            + t1 * t1) / (h(ss, 1, 0) + h(ss, 0, 1))
                        q = h(ss, 0, 0) + h(ss, 1, 1) - 1.5 * t1
                        r = h(ss, 2, 1)
                    Else
                        t1 = h(ss, n - 1, n - 1) - h(ss, 0, 0)
                        t2 = h(ss, n - 2, n - 2) - h(ss, 0, 0)
                        p = (t1 * t2 - h(ss, n - 1, n - 2) * h(ss, n - 2, n - 1))
                            / (h(ss, 1, 0) + h(ss, 0, 1))
                        q = h(ss, 1, 1) - h(ss, 0, 0) - t1 - t2
                        r = h(ss, 2, 1)
                    End If
                Else
                    p = h(ss, l - 1, l - 2)
                    q = h(ss, l, l - 2)
                    r = 0
                End If
                'Vektoriaus normavimas
                xt = Math.Abs(p) + Math.Abs(q) + Math.Abs(r)
                If xt <> 0 Then
                    p = p / xt
                    q = q / xt
                    r = r / xt
                    s = Math.Sqrt(p * p + q * q + r * r)
                    If p < 0 Then
                        s = -s
                    End If
                    If l <> 1 Then
                        h(ss, l - 1, l - 2) = -s * xt
                    End If
                    'Atspindžio matricos formavimas
                    'vektorius u:
                    p = p + s
                    xt = p / s
                    yt = q / s
                    zt = r / s
                    'vektorius v:
                    q = q / p
                    r = r / p
                    'Sandaugos H = Q*H apskaičiavimas
                    For j = 1 To n
                        p = h(ss, l - 1, j - 1) + q * h(ss, l, j - 1)
                        If l <> n - 1 Then
                            p = p + r * h(ss, l + 1, j - 1)
                            h(ss, l + 1, j - 1) = h(ss, l + 1, j - 1) - p * zt
                        End If
                        h(ss, l, j - 1) = h(ss, l, j - 1) - p * yt
                        h(ss, l - 1, j - 1) = h(ss, l - 1, j - 1) - p * xt
                    Next
                End If
            Next
        End While
    End If
    n = n - 1
End While

```

```

        m = n
        'Sandaugos H = H*Q apskaičiavimas
        For i = 1 To m
            p = xt * h(ss, i - 1, l - 1) + yt * h(ss, i - 1, l)
            If l <> n - 1 Then
                p = p + zt * h(ss, i - 1, l + 1)
                h(ss, i - 1, l + 1) = h(ss, i - 1, l + 1) - p * r
            End If
            h(ss, i - 1, l) = h(ss, i - 1, l) - p * q
            h(ss, i - 1, l - 1) = h(ss, i - 1, l - 1) - p
        Next
    End If
Next
    eps = macheps * (Math.Abs(h(ss, n-2, n-2)) + Math.Abs(h(ss, n-1, n-1)))
End While
If it > 30 Then
    MsgBox("it > 30")
End If
End If

If n = 1 Then
    'Rasta viena tikrinė reikšmė
    wr(n - 1) = h(ss, n - 1, n - 1)
    wi(n - 1) = 0
    iter(n - 1) = it
    n = n - 1
ElseIf Math.Abs(h(ss, n - 1, n - 2)) < eps Then
    'Rasta viena tikrinė reikšmė
    wr(n - 1) = h(ss, n - 1, n - 1)
    wi(n - 1) = 0
    iter(n - 1) = it
    n = n - 1
ElseIf n = 2 Then
    'Rastos dvi tikrinės reikšmės
    p = (h(ss, n - 2, n - 2) + h(ss, n - 1, n - 1)) / 2
    q = h(ss, n - 1, n - 1) * h(ss, n-2, n-2) - h(ss, n-2, n-1) * h(ss, n-1, n-2)
    yt = p * p - q
    t1 = Math.Sqrt(Math.Abs(yt))
    iter(n - 1) = -it
    iter(n - 2) = it
    If yt < 0 Then
        wr(n - 1) = p
        wi(n - 1) = t1
        wr(n - 2) = p
        wi(n - 2) = -t1
    Else
        wr(n - 1) = p + t1
        wr(n - 2) = p - t1
    End If
    n = n - 2
ElseIf Math.Abs(h(ss, n - 2, n - 3)) < eps Then
    'Rastos dvi tikrinės reikšmės
    p = (h(ss, n - 2, n - 2) + h(ss, n - 1, n - 1)) / 2
    q = h(ss, n-1, n-1) * h(ss, n-2, n-2) - h(ss, n-2, n-1) * h(ss, n-1, n-2)
    yt = p * p - q
    t1 = Math.Sqrt(Math.Abs(yt))
    iter(n - 1) = -it
    iter(n - 2) = it
    If yt < 0 Then
        wr(n - 1) = p
        wi(n - 1) = t1
        wr(n - 2) = p
        wi(n - 2) = -t1
    Else
        wr(n - 1) = p + t1
        wr(n - 2) = p - t1
    End If
    n = n - 2

```

```

        Else
            n = n - 1
        End If
    End While

    rksm(ss, 0, 0) = wr(0)
    rksm(ss, 0, 1) = wi(0)
    rksm(ss, 1, 0) = wr(1)
    rksm(ss, 1, 1) = wi(1)
    rksm(ss, 2, 0) = wr(2)
    rksm(ss, 2, 1) = wi(2)
Next
Tikrines = rksm
End Function

Public Function rikiuoti2(ByVal mk, ByVal tikrine) As Double(,,)
    Dim fu(mk, 2, 1) As Double
    Dim pr(1) As Double
    Dim an(1) As Double
    Dim tr(1) As Double
    Dim sk1 As Double
    Dim sk2 As Double
    Dim sk3 As Double
    Dim rd1 As Double
    Dim kd1 As Double
    Dim rd2 As Double
    Dim kd2 As Double
    Dim rd3 As Double
    Dim kd3 As Double

    fu = tikrine
    'Surikiuojam pirmas reikšmes
    For i = 0 To mk
        'priskiriam tikrines reikšmes
        For j = 0 To 1
            pr(j) = fu(i, 0, j)
            an(j) = fu(i, 1, j)
            tr(j) = fu(i, 2, j)
        Next
        'randam absoliučią reikšmę
        rd1 = fu(i, 0, 0)
        kd1 = fu(i, 0, 1)
        If rd1 >= 0 Then
            sk1 = Math.Sqrt(rd1 * rd1 + kd1 * kd1)
        Else
            sk1 = -Math.Sqrt(rd1 * rd1 + kd1 * kd1)
        End If
        rd2 = fu(i, 1, 0)
        kd2 = fu(i, 1, 1)
        If rd2 >= 0 Then
            sk2 = Math.Sqrt(rd2 * rd2 + kd2 * kd2)
        Else
            sk2 = -Math.Sqrt(rd2 * rd2 + kd2 * kd2)
        End If
        rd3 = fu(i, 2, 0)
        kd3 = fu(i, 2, 1)
        If rd3 >= 0 Then
            sk3 = Math.Sqrt(rd3 * rd3 + kd3 * kd3)
        Else
            sk3 = -Math.Sqrt(rd3 * rd3 + kd3 * kd3)
        End If
        'Surikiuojam tikrines reikšmes
        '1 atvejis - Kai visos reikšmės skirtingos
        If sk1 <> sk2 And sk1 <> sk3 And sk2 <> sk3 Then
            If sk1 < sk2 And sk1 < sk3 Then
                'Kai pirma mažiausia
                fu(i, 0, 0) = pr(0)
                fu(i, 0, 1) = pr(1)
            End If
        End If
    Next
End Function

```

```

If sk2 < sk3 Then
    fu(i, 1, 0) = an(0)
    fu(i, 1, 1) = an(1)
    fu(i, 2, 0) = tr(0)
    fu(i, 2, 1) = tr(1)
Else
    fu(i, 1, 0) = tr(0)
    fu(i, 1, 1) = tr(1)
    fu(i, 2, 0) = an(0)
    fu(i, 2, 1) = an(1)
End If
ElseIf sk2 < sk1 And sk2 < sk3 Then
    'Kai antra mažiausia
    fu(i, 0, 0) = an(0)
    fu(i, 0, 1) = an(1)
    If sk1 < sk3 Then
        fu(i, 1, 0) = pr(0)
        fu(i, 1, 1) = pr(1)
        fu(i, 2, 0) = tr(0)
        fu(i, 2, 1) = tr(1)
    Else
        fu(i, 1, 0) = tr(0)
        fu(i, 1, 1) = tr(1)
        fu(i, 2, 0) = pr(0)
        fu(i, 2, 1) = pr(1)
    End If
Else
    'Kai trečia mažiausia
    fu(i, 0, 0) = tr(0)
    fu(i, 0, 1) = tr(1)
    If sk1 < sk2 Then
        fu(i, 1, 0) = pr(0)
        fu(i, 1, 1) = pr(1)
        fu(i, 2, 0) = an(0)
        fu(i, 2, 1) = an(1)
    Else
        fu(i, 1, 0) = an(0)
        fu(i, 1, 1) = an(1)
        fu(i, 2, 0) = pr(0)
        fu(i, 2, 1) = pr(1)
    End If
End If
ElseIf sk1 = sk2 And sk2 = sk3 Then '2 atvejis - kai visos vienodos
    If kd1 <= kd2 And kd1 <= kd3 Then
        fu(i, 0, 0) = pr(0)
        fu(i, 0, 1) = pr(1)
        If kd2 < kd3 Then
            fu(i, 1, 0) = an(0)
            fu(i, 1, 1) = an(1)
            fu(i, 2, 0) = tr(0)
            fu(i, 2, 1) = tr(1)
        Else
            fu(i, 1, 0) = tr(0)
            fu(i, 1, 1) = tr(1)
            fu(i, 2, 0) = an(0)
            fu(i, 2, 1) = an(1)
        End If
    ElseIf kd2 < kd1 And kd2 < kd3 Then
        fu(i, 0, 0) = an(0)
        fu(i, 0, 1) = an(1)
        If kd1 < kd3 Then
            fu(i, 1, 0) = pr(0)
            fu(i, 1, 1) = pr(1)
            fu(i, 2, 0) = tr(0)
            fu(i, 2, 1) = tr(1)
        Else
            fu(i, 1, 0) = tr(0)
            fu(i, 1, 1) = tr(1)
        End If
    End If

```

```

        fu(i, 2, 0) = pr(0)
        fu(i, 2, 1) = pr(1)
    End If
Else
    fu(i, 0, 0) = tr(0)
    fu(i, 0, 1) = tr(1)
    If kd1 < kd2 Then
        fu(i, 1, 0) = pr(0)
        fu(i, 1, 1) = pr(1)
        fu(i, 2, 0) = an(0)
        fu(i, 2, 1) = an(1)
    Else
        fu(i, 1, 0) = an(0)
        fu(i, 1, 1) = an(1)
        fu(i, 2, 0) = pr(0)
        fu(i, 2, 1) = pr(1)
    End If
End If
ElseIf sk1 = sk2 Then 'kai pirmos dvi vienodos, trečia - ne.
    If sk1 < sk3 Then ' kai jos mažesnės už trečią
        If kd1 <= kd2 Then
            fu(i, 0, 0) = pr(0)
            fu(i, 0, 1) = pr(1)
            fu(i, 1, 0) = an(0)
            fu(i, 1, 1) = an(1)
            fu(i, 2, 0) = tr(0)
            fu(i, 2, 1) = tr(1)
        Else
            fu(i, 0, 0) = an(0)
            fu(i, 0, 1) = an(1)
            fu(i, 1, 0) = pr(0)
            fu(i, 1, 1) = pr(1)
            fu(i, 2, 0) = tr(0)
            fu(i, 2, 1) = tr(1)
        End If
    Else 'kai jos didesnės už trečią
        If kd1 <= kd2 Then
            fu(i, 0, 0) = tr(0)
            fu(i, 0, 1) = tr(1)
            fu(i, 1, 0) = pr(0)
            fu(i, 1, 1) = pr(1)
            fu(i, 2, 0) = an(0)
            fu(i, 2, 1) = an(1)
        Else
            fu(i, 0, 0) = tr(0)
            fu(i, 0, 1) = tr(1)
            fu(i, 1, 0) = an(0)
            fu(i, 1, 1) = an(1)
            fu(i, 2, 0) = pr(0)
            fu(i, 2, 1) = pr(1)
        End If
    End If
ElseIf sk1 = sk3 Then ' kai pirma ir trečia lygios
    If sk1 < sk2 Then ' kai jos mažesnės už antrą
        If kd1 <= kd3 Then
            fu(i, 0, 0) = pr(0)
            fu(i, 0, 1) = pr(1)
            fu(i, 1, 0) = tr(0)
            fu(i, 1, 1) = tr(1)
            fu(i, 2, 0) = an(0)
            fu(i, 2, 1) = an(1)
        Else
            fu(i, 0, 0) = tr(0)
            fu(i, 0, 1) = tr(1)
            fu(i, 1, 0) = pr(0)
            fu(i, 1, 1) = pr(1)
            fu(i, 2, 0) = an(0)
            fu(i, 2, 1) = an(1)
        End If
    End If
End If

```



```

    End If
Else 'kai jos didesnės už antrą
    If kd1 <= kd3 Then
        fu(i, 0, 0) = an(0)
        fu(i, 0, 1) = an(1)
        fu(i, 1, 0) = pr(0)
        fu(i, 1, 1) = pr(1)
        fu(i, 2, 0) = tr(0)
        fu(i, 2, 1) = tr(1)
    Else
        fu(i, 0, 0) = an(0)
        fu(i, 0, 1) = an(1)
        fu(i, 1, 0) = tr(0)
        fu(i, 1, 1) = tr(1)
        fu(i, 2, 0) = pr(0)
        fu(i, 2, 1) = pr(1)
    End If
End If
ElseIf sk2 = sk3 Then ' kai antra ir trečia lygios
    If sk2 < sk1 Then ' kai jos mažesnės už pirmą
        If kd2 <= kd3 Then
            fu(i, 0, 0) = an(0)
            fu(i, 0, 1) = an(1)
            fu(i, 1, 0) = tr(0)
            fu(i, 1, 1) = tr(1)
            fu(i, 2, 0) = pr(0)
            fu(i, 2, 1) = pr(1)
        Else
            fu(i, 0, 0) = tr(0)
            fu(i, 0, 1) = tr(1)
            fu(i, 1, 0) = an(0)
            fu(i, 1, 1) = an(1)
            fu(i, 2, 0) = pr(0)
            fu(i, 2, 1) = pr(1)
        End If
    Else 'kai jos didesnės už pirmą
        If kd2 <= kd3 Then
            fu(i, 0, 0) = pr(0)
            fu(i, 0, 1) = pr(1)
            fu(i, 1, 0) = an(0)
            fu(i, 1, 1) = an(1)
            fu(i, 2, 0) = tr(0)
            fu(i, 2, 1) = tr(1)
        Else
            fu(i, 0, 0) = pr(0)
            fu(i, 0, 1) = pr(1)
            fu(i, 1, 0) = tr(0)
            fu(i, 1, 1) = tr(1)
            fu(i, 2, 0) = an(0)
            fu(i, 2, 1) = an(1)
        End If
    End If
End If
Next
rikiuoti2 = fu
End Function

Public Function daugyba(ByVal daug1, ByVal daug2) As Double(,)
    Dim pir(2, 2) As Double
    Dim ant(2, 2) As Double
    pir = daug1
    ant = daug2
    Dim funk(2, 2) As Double

    funk(0, 0) = pir(0, 0) * ant(0, 0) + pir(0, 1) * ant(1, 0) + pir(0, 2) * ant(2, 0)
    funk(0, 1) = pir(0, 0) * ant(0, 1) + pir(0, 1) * ant(1, 1) + pir(0, 2) * ant(2, 1)
    funk(0, 2) = pir(0, 0) * ant(0, 2) + pir(0, 1) * ant(1, 2) + pir(0, 2) * ant(2, 2)
    funk(1, 0) = pir(1, 0) * ant(0, 0) + pir(1, 1) * ant(1, 0) + pir(1, 2) * ant(2, 0)

```

```

funk(1, 1) = pir(1, 0) * ant(0, 1) + pir(1, 1) * ant(1, 1) + pir(1, 2) * ant(2, 1)
funk(1, 2) = pir(1, 0) * ant(0, 2) + pir(1, 1) * ant(1, 2) + pir(1, 2) * ant(2, 2)
funk(2, 0) = pir(2, 0) * ant(0, 0) + pir(2, 1) * ant(1, 0) + pir(2, 2) * ant(2, 0)
funk(2, 1) = pir(2, 0) * ant(0, 1) + pir(2, 1) * ant(1, 1) + pir(2, 2) * ant(2, 1)
funk(2, 2) = pir(2, 0) * ant(0, 2) + pir(2, 1) * ant(1, 2) + pir(2, 2) * ant(2, 2)

```

```
daugyba = funk
```

```
End Function
```

```
Private Function kdaugyba(ByVal kdaug1, ByVal kdaug2) As Double(,,)
```

```
Dim kpir(2, 2, 1) As Double
```

```
Dim kant(2, 2, 1) As Double
```

```
kpir = kdaug1
```

```
kant = kdaug2
```

```
Dim kfunk(2, 2, 1) As Double
```

```
'Realiosios dalys
```

```

kfunk(0, 0, 0) = kpir(0, 0, 0) * kant(0, 0, 0) - kpir(0, 0, 1) * kant(0, 0, 1) +
    kpir(0, 1, 0) * kant(1, 0, 0) - kpir(0, 1, 1) * kant(1, 0, 1) +
    kpir(0, 2, 0) * kant(2, 0, 0) - kpir(0, 2, 1) * kant(2, 0, 1)
kfunk(0, 1, 0) = kpir(0, 0, 0) * kant(0, 1, 0) - kpir(0, 0, 1) * kant(0, 1, 1) +
    kpir(0, 1, 0) * kant(1, 1, 0) - kpir(0, 1, 1) * kant(1, 1, 1) +
    kpir(0, 2, 0) * kant(2, 1, 0) - kpir(0, 2, 1) * kant(2, 1, 1)
kfunk(0, 2, 0) = kpir(0, 0, 0) * kant(0, 2, 0) - kpir(0, 0, 1) * kant(0, 2, 1) +
    kpir(0, 1, 0) * kant(1, 2, 0) - kpir(0, 1, 1) * kant(1, 2, 1) +
    kpir(0, 2, 0) * kant(2, 2, 0) - kpir(0, 2, 1) * kant(2, 2, 1)
kfunk(1, 0, 0) = kpir(1, 0, 0) * kant(0, 0, 0) - kpir(1, 0, 1) * kant(0, 0, 1) +
    kpir(1, 1, 0) * kant(1, 0, 0) - kpir(1, 1, 1) * kant(1, 0, 1) +
    kpir(1, 2, 0) * kant(2, 0, 0) - kpir(1, 2, 1) * kant(2, 0, 1)
kfunk(1, 1, 0) = kpir(1, 0, 0) * kant(0, 1, 0) - kpir(1, 0, 1) * kant(0, 1, 1) +
    kpir(1, 1, 0) * kant(1, 1, 0) - kpir(1, 1, 1) * kant(1, 1, 1) +
    kpir(1, 2, 0) * kant(2, 1, 0) - kpir(1, 2, 1) * kant(2, 1, 1)
kfunk(1, 2, 0) = kpir(1, 0, 0) * kant(0, 2, 0) - kpir(1, 0, 1) * kant(0, 2, 1) +
    kpir(1, 1, 0) * kant(1, 2, 0) - kpir(1, 1, 1) * kant(1, 2, 1) +
    kpir(1, 2, 0) * kant(2, 2, 0) - kpir(1, 2, 1) * kant(2, 2, 1)
kfunk(2, 0, 0) = kpir(2, 0, 0) * kant(0, 0, 0) - kpir(2, 0, 1) * kant(0, 0, 1) +
    kpir(2, 1, 0) * kant(1, 0, 0) - kpir(2, 1, 1) * kant(1, 0, 1) +
    kpir(2, 2, 0) * kant(2, 0, 0) - kpir(2, 2, 1) * kant(2, 0, 1)
kfunk(2, 1, 0) = kpir(2, 0, 0) * kant(0, 1, 0) - kpir(2, 0, 1) * kant(0, 1, 1) +
    kpir(2, 1, 0) * kant(1, 1, 0) - kpir(2, 1, 1) * kant(1, 1, 1) +
    kpir(2, 2, 0) * kant(2, 1, 0) - kpir(2, 2, 1) * kant(2, 1, 1)
kfunk(2, 2, 0) = kpir(2, 0, 0) * kant(0, 2, 0) - kpir(2, 0, 1) * kant(0, 2, 1) +
    kpir(2, 1, 0) * kant(1, 2, 0) - kpir(2, 1, 1) * kant(1, 2, 1) +
    kpir(2, 2, 0) * kant(2, 2, 0) - kpir(2, 2, 1) * kant(2, 2, 1)

```

```
'Kompleksinés dalys
```

```

kfunk(0, 0, 1) = kpir(0, 0, 0) * kant(0, 0, 1) + kpir(0, 0, 1) * kant(0, 0, 0) +
    kpir(0, 1, 0) * kant(1, 0, 1) + kpir(0, 1, 1) * kant(1, 0, 0) +
    kpir(0, 2, 0) * kant(2, 0, 1) + kpir(0, 2, 1) * kant(2, 0, 0)
kfunk(0, 1, 1) = kpir(0, 0, 0) * kant(0, 1, 1) + kpir(0, 0, 1) * kant(0, 1, 0) +
    kpir(0, 1, 0) * kant(1, 1, 1) + kpir(0, 1, 1) * kant(1, 1, 0) +
    kpir(0, 2, 0) * kant(2, 1, 1) + kpir(0, 2, 1) * kant(2, 1, 0)
kfunk(0, 2, 1) = kpir(0, 0, 0) * kant(0, 2, 1) + kpir(0, 0, 1) * kant(0, 2, 0) +
    kpir(0, 1, 0) * kant(1, 2, 1) + kpir(0, 1, 1) * kant(1, 2, 0) +
    kpir(0, 2, 0) * kant(2, 2, 1) + kpir(0, 2, 1) * kant(2, 2, 0)
kfunk(1, 0, 1) = kpir(1, 0, 0) * kant(0, 0, 1) + kpir(1, 0, 1) * kant(0, 0, 0) +
    kpir(1, 1, 0) * kant(1, 0, 1) + kpir(1, 1, 1) * kant(1, 0, 0) +
    kpir(1, 2, 0) * kant(2, 0, 1) + kpir(1, 2, 1) * kant(2, 0, 0)
kfunk(1, 1, 1) = kpir(1, 0, 0) * kant(0, 1, 1) + kpir(1, 0, 1) * kant(0, 1, 0) +
    kpir(1, 1, 0) * kant(1, 1, 1) + kpir(1, 1, 1) * kant(1, 1, 0) +
    kpir(1, 2, 0) * kant(2, 1, 1) + kpir(1, 2, 1) * kant(2, 1, 0)
kfunk(1, 2, 1) = kpir(1, 0, 0) * kant(0, 2, 1) + kpir(1, 0, 1) * kant(0, 2, 0) +
    kpir(1, 1, 0) * kant(1, 2, 1) + kpir(1, 1, 1) * kant(1, 2, 0) +
    kpir(1, 2, 0) * kant(2, 2, 1) + kpir(1, 2, 1) * kant(2, 2, 0)
kfunk(2, 0, 1) = kpir(2, 0, 0) * kant(0, 0, 1) + kpir(2, 0, 1) * kant(0, 0, 0) +
    kpir(2, 1, 0) * kant(1, 0, 1) + kpir(2, 1, 1) * kant(1, 0, 0) +
    kpir(2, 2, 0) * kant(2, 0, 1) + kpir(2, 2, 1) * kant(2, 0, 0)
kfunk(2, 1, 1) = kpir(2, 0, 0) * kant(0, 1, 1) + kpir(2, 0, 1) * kant(0, 1, 0) +
    kpir(2, 1, 0) * kant(1, 1, 1) + kpir(2, 1, 1) * kant(1, 1, 0) +

```

```

    kpir(2, 2, 0) * kant(2, 1, 1) + kpir(2, 2, 1) * kant(2, 1, 0)
kfunk(2, 2, 1) = kpir(2, 0, 0) * kant(0, 2, 1) + kpir(2, 0, 1) * kant(0, 2, 0) +
    kpir(2, 1, 0) * kant(1, 2, 1) + kpir(2, 1, 1) * kant(1, 2, 0) +
    kpir(2, 2, 0) * kant(2, 2, 1) + kpir(2, 2, 1) * kant(2, 2, 0)

    kdaugyba = kfunk
End Function

Public Function alfabetagama(ByVal mk, ByVal w, ByVal lmd) As Double(,,)
    Dim ava As Double
    Dim bva As Double
    Dim ask As Double
    Dim bsk As Double
    Dim laik1 As Double
    Dim laik2 As Double
    Dim laik3 As Double
    Dim albega(2, mk, 1) As Double

    For i = 0 To mk
        'alfa ir beta vardiklių realiosios dalys
        ava = w(i, 0, 1) * w(i, 1, 2) - w(i, 1, 1) * w(i, 0, 2) + w(i, 0, 2) * lmd(i, 0, 0)
        bva = w(i, 1, 1) * w(i, 0, 2) - w(i, 0, 1) * w(i, 1, 2) - w(i, 0, 2) * lmd(i, 2, 0)
        'jeigu skaičiai bus realūs
        If lmd(i, 0, 1) = 0 And lmd(i, 1, 1) = 0 And lmd(i, 2, 1) = 0 Then
            'jeigu visi vardikliai nelygūs nuliui
            If ava <> 0 And bva <> 0 And w(i, 0, 2) <> 0 Then
                'alfa skaičiavimas
                albega(0, i, 0) = (w(i, 0, 0) * w(i, 1, 2) - w(i, 1, 0) * w(i, 0, 2)
                    - w(i, 1, 2) * lmd(i, 0, 0)) / ava

                albega(0, i, 1) = 0
                'beta skaičiavimas
                albega(1, i, 0) = (w(i, 2, 2) * w(i, 0, 1) - w(i, 2, 1) * w(i, 0, 2)
                    - w(i, 0, 1) * lmd(i, 2, 0)) / bva

                albega(1, i, 1) = 0
                'gama skaičiavimas
                albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0)
                    - w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / (2 * w(i, 0, 2))
                albega(2, i, 1) = 0
                'jei tik trečias vardiklis lygus nuliui:
            ElseIf ava <> 0 And bva <> 0 And w(i, 0, 2) = 0 Then
                albega(0, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 1)
                albega(0, i, 1) = 0
                albega(1, i, 0) = (lmd(i, 2, 0) - w(i, 2, 2)) / w(i, 1, 2)
                albega(1, i, 1) = 0
                albega(2, i, 0) = (w(i, 1, 0) + albega(0, i, 0) * (w(i, 0, 0) - w(i, 1, 1))
                    - albega(0, i, 0) * albega(0, i, 0) * w(i, 0, 1)) / w(i, 1, 2)
                    + (albega(0, i, 0) * albega(1, i, 0)) / 2
                albega(2, i, 1) = 0
                'jei tik antras vardiklis lygus nuliui
            ElseIf ava <> 0 And bva = 0 And w(i, 0, 2) <> 0 Then
                albega(0, i, 0) = (w(i, 0, 0) * w(i, 1, 2) - w(i, 1, 0) * w(i, 0, 2)
                    - w(i, 1, 2) * lmd(i, 0, 0)) / ava

                albega(0, i, 1) = 0
                laik1 = w(i, 0, 2) * albega(0, i, 0) + w(i, 1, 2)
                If laik1 = 0 Then
                    albega(0, i, 0) = -w(i, 1, 2) / w(i, 0, 2)
                    albega(1, i, 0) = 0
                    albega(1, i, 1) = 0
                    albega(2, i, 0) = (lmd(i, 2, 0) - w(i, 2, 2)) / w(i, 0, 2)
                    albega(2, i, 1) = 0
                Else
                    albega(1, i, 0) = (w(i, 1, 1) - lmd(i, 1, 0) + w(i, 0, 1)
                        * albega(0, i, 0)) / laik1

                    albega(1, i, 1) = 0
                    albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0) -
                        w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / (2 * w(i, 0, 2))
                    albega(2, i, 1) = 0
                End If
            End If
        End For
    End Function

```

```

'jei tik pirmas vardiklis lygus nuliui
ElseIf ava = 0 And bva <> 0 And w(i, 0, 2) <> 0 Then
'beta skaičiavimas
albega(1, i, 0) = (w(i, 2, 2) * w(i, 0, 1) - w(i, 2, 1) * w(i, 0, 2)
- w(i, 0, 1) * lmd(i, 2, 0)) / bva
albega(1, i, 1) = 0

'tikrinam ar alfa vardiklis lygus nuliui
laik1 = w(i, 0, 2) * albega(1, i, 0) - w(i, 0, 1)
If laik1 = 0 Then
albega(0, i, 0) = 0
albega(0, i, 1) = 0
albega(1, i, 0) = w(i, 0, 1) / w(i, 0, 2)
albega(1, i, 1) = 0
albega(2, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 2)
albega(2, i, 1) = 0
Else
'alfa skaičiavimas
albega(0, i, 0) = (w(i, 1, 1) - lmd(i, 1, 0) - w(i, 1, 2)
* albega(1, i, 0)) / laik1
albega(0, i, 1) = 0
'gama skaičiavimas
albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0)
- w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / (2 * w(i, 0, 2))
albega(2, i, 1) = 0
End If
ElseIf ava = 0 And bva = 0 And w(i, 0, 2) <> 0 Then
albega(0, i, 0) = 0
albega(1, i, 0) = 0
albega(0, i, 1) = 0
albega(1, i, 1) = 0
albega(2, i, 0) = 0
albega(2, i, 1) = 0
ElseIf ava = 0 And bva = 0 And w(i, 0, 2) = 0 Then
'pirmas atvejis
If w(i, 0, 2) = 0 And w(i, 0, 1) = 0 And w(i, 1, 2) <> 0 Then
albega(1, i, 0) = (w(i, 1, 1) - lmd(i, 1, 0)) / w(i, 1, 2)
albega(1, i, 1) = 0
laik1 = w(i, 0, 0) * w(i, 0, 0) - w(i, 0, 0) * w(i, 1, 1) - w(i, 1, 2)
* w(i, 2, 1) + w(i, 1, 1) * w(i, 2, 2) - w(i, 0, 0) * w(i, 2, 2)
If laik1 = 0 Then
albega(0, i, 0) = 0
Else
albega(0, i, 0) = (w(i, 1, 0) * w(i, 2, 2) - w(i, 1, 2) * w(i, 2, 0)
- w(i, 0, 0) * w(i, 1, 0)) / laik1
End If
albega(0, i, 1) = 0
albega(2, i, 0) = (w(i, 1, 0) + albega(0, i, 0) * (w(i, 0, 0) - w(i, 1, 1)))
/ w(i, 1, 2) + (albega(0, i, 0) * albega(1, i, 0)) / 2
albega(2, i, 1) = 0
'antras atvejis
ElseIf w(i, 0, 2) = 0 And w(i, 0, 1) <> 0 And w(i, 1, 2) = 0 Then
albega(0, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 1)
albega(0, i, 1) = 0
laik1 = w(i, 0, 0) * w(i, 2, 2) - w(i, 0, 0) * w(i, 1, 1) + w(i, 0, 1)
* w(i, 1, 0) + w(i, 1, 1) * w(i, 2, 2) - w(i, 2, 2) * w(i, 2, 2)
If laik1 = 0 Then
albega(1, i, 0) = 0
Else
albega(1, i, 0) = (w(i, 0, 0) * w(i, 2, 1) - w(i, 0, 1) * w(i, 2, 0)
- w(i, 2, 1) * w(i, 2, 2)) / laik1
End If
albega(1, i, 1) = 0
albega(2, i, 0) = ((w(i, 2, 2) - w(i, 1, 1)) * albega(1, i, 0) - w(i, 2, 1))
/ w(i, 0, 1) - (albega(0, i, 0) * albega(1, i, 0)) / 2
albega(2, i, 1) = 0
'trečias atvejis
ElseIf w(i, 0, 2) = 0 And w(i, 0, 1) = 0 And w(i, 1, 2) = 0 Then

```

```

laik1 = w(i, 1, 1) - w(i, 0, 0)
laik2 = w(i, 2, 2) - w(i, 1, 1)
laik3 = w(i, 2, 2) - w(i, 0, 0)
If laik1 <> 0 And laik2 <> 0 And laik3 <> 0 Then
    albeg(0, i, 0) = w(i, 1, 0) / laik1
    albeg(0, i, 1) = 0
    albeg(1, i, 0) = w(i, 2, 1) / laik2
    albeg(1, i, 1) = 0
    albeg(2, i, 0) = (0.5 * albeg(0, i, 0) * albeg(1, i, 0) * (w(i, 0, 0)
        - 2 * w(i, 1, 1) + w(i, 2, 2)) + w(i, 2, 0)
        - albeg(0, i, 0) * w(i, 2, 1) + albeg(1, i, 0) * w(i, 1, 0)) / laik3
    albeg(2, i, 1) = 0
ElseIf laik1 = 0 And laik2 <> 0 And laik3 <> 0 Then
    albeg(0, i, 0) = 0
    albeg(0, i, 1) = 0
    albeg(1, i, 0) = w(i, 2, 1) / laik2
    albeg(1, i, 1) = 0
    albeg(2, i, 0) = (w(i, 2, 0) + w(i, 1, 0) * albeg(1, i, 0)) / laik3
    albeg(2, i, 1) = 0
ElseIf laik1 <> 0 And laik2 = 0 And laik3 <> 0 Then
    albeg(0, i, 0) = w(i, 1, 0) / laik1
    albeg(0, i, 1) = 0
    albeg(1, i, 0) = 0
    albeg(1, i, 1) = 0
    albeg(2, i, 0) = (w(i, 2, 0) - w(i, 2, 1) * albeg(0, i, 0)) / laik3
    albeg(2, i, 1) = 0
ElseIf laik1 <> 0 And laik2 <> 0 And laik3 = 0 Then
    albeg(0, i, 0) = w(i, 1, 0) / laik1
    albeg(0, i, 1) = 0
    albeg(1, i, 0) = w(i, 2, 1) / laik2
    albeg(1, i, 1) = 0
    albeg(2, i, 0) = 0
    albeg(2, i, 1) = 0
ElseIf laik1 = 0 And laik2 = 0 And laik3 = 0 Then
    'turi tenkinti lygybę a32*alfa-a21*beta = a31
    If w(i, 2, 0) <> 0 And w(i, 2, 1) <> 0 Then
        albeg(0, i, 0) = w(i, 2, 0) / w(i, 2, 1)
        albeg(1, i, 0) = 0
    ElseIf w(i, 2, 0) <> 0 And w(i, 1, 0) <> 0 Then
        albeg(0, i, 0) = 0
        albeg(1, i, 0) = -w(i, 2, 0) / w(i, 1, 0)
    Else
        albeg(0, i, 0) = 0
        albeg(1, i, 0) = 0
    End If
    albeg(0, i, 1) = 0
    albeg(1, i, 1) = 0
    albeg(2, i, 0) = 0
    albeg(2, i, 1) = 0
End If
End If
End If
'jei skaičiai kompleksiniai
Else
    'jeigu visi vardikliai nelygūs nuliui
    If (ava <> 0 Or lmd(i,0,1)<>0) And (bva<>0 Or lmd(i,2,1)<>0) And w(i,0,2)<>0 Then
        'alfa skaičiavimas
        ask = w(i, 0, 0) * w(i, 1, 2) - w(i, 1, 0) * w(i, 0, 2) - w(i, 1, 2) * lmd(i, 0, 0)
        laik1 = w(i, 0, 2) * lmd(i, 0, 1)
        laik2 = ava * ava + laik1 * laik1
        'realioji alfa dalis
        albeg(0, i, 0) = (ask * ava - w(i, 1, 2) * laik1 * lmd(i, 0, 1)) / laik2
        'kompleksinė alfa dalis
        albeg(0, i, 1) = -((ava * w(i, 1, 2) + ask * w(i, 0, 2)) * lmd(i, 0, 1)) / laik2

        'beta skaičiavimas
        bsk = w(i, 2, 2) * w(i, 0, 1) - w(i, 2, 1) * w(i, 0, 2) - w(i, 0, 1) * lmd(i, 2, 0)
        laik1 = w(i, 0, 2) * lmd(i, 2, 1)

```

```

laik2 = bva * bva + laik1 * laik1
'realioji beta dalis
albega(1, i, 0) = (bsk * bva + w(i, 0, 1) * laik1 * lmd(i, 2, 1)) / laik2
'kompleksinė beta dalis
albega(1, i, 1) = ((bsk * w(i, 0, 2) - bva * w(i, 0, 1)) * lmd(i, 2, 1)) / laik2

'gama skaičiavimas
laik1 = 2 * w(i, 0, 2)
'realioji gama dalis
albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0)
- w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / laik1
'kompleksinė gama dalis
albega(2, i, 1) = (lmd(i, 2, 1) - lmd(i, 0, 1) - w(i, 0, 1) * albega(0, i, 1)
- w(i, 1, 2) * albega(1, i, 1)) / laik1

'jei tik trečias vardiklis lygus nuliui:
ElseIf ava <> 0 And bva <> 0 And w(i, 0, 2) = 0 Then
albega(0, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 1)
albega(0, i, 1) = -lmd(i, 0, 1) / w(i, 0, 1)
albega(1, i, 0) = (lmd(i, 2, 0) - w(i, 2, 2)) / w(i, 1, 2)
albega(1, i, 1) = lmd(i, 2, 1) / w(i, 1, 2)
'realioji gama dalis
laik1 = (w(i, 1, 0) + albega(0, i, 0) * (w(i, 0, 0) - w(i, 1, 1)) -
(albega(0, i, 0) * albega(0, i, 0) - albega(0, i, 1) * albega(0, i, 1)) * w(i, 0, 1))
laik2 = albega(0, i, 0) * albega(1, i, 0) - albega(0, i, 1) * albega(1, i, 1)
albega(2, i, 0) = laik1 / w(i, 1, 2) + laik2 / 2
'kompleksinė gama dalis
laik1 = albega(0, i, 1) * (w(i, 0, 0) - w(i, 1, 1)) - 2 * albega(0, i, 0)
* albega(0, i, 1) * w(i, 0, 1)
laik2 = albega(0, i, 0) * albega(1, i, 1) + albega(0, i, 1) * albega(1, i, 0)
albega(2, i, 1) = laik1 / w(i, 1, 2) + laik2 / 2
'jei tik antras vardiklis lygus nuliui
ElseIf ava <> 0 And bva = 0 And w(i, 0, 2) <> 0 Then
'alfa skaičiavimas
ask = w(i, 0, 0) * w(i, 1, 2) - w(i, 1, 0) * w(i, 0, 2) - w(i, 1, 2) * lmd(i, 0, 0)
laik1 = w(i, 0, 2) * lmd(i, 0, 1)
laik2 = ava * ava + laik1 * laik1
'realioji alfa dalis
albega(0, i, 0) = (ask * ava - w(i, 1, 2) * laik1 * lmd(i, 0, 1)) / laik2
'kompleksinė alfa dalis
albega(0, i, 1) = -((ava * w(i, 1, 2) + ask * w(i, 0, 2)) * lmd(i, 0, 1)) / laik2
'tikrinam ar beta vidurkis lygus nuliui
laik1 = w(i, 0, 2) * albega(0, i, 0) - w(i, 1, 2)
laik2 = w(i, 0, 2) * albega(0, i, 1)
bsk = laik1 * laik1 + laik2 * laik2
If bsk = 0 Then
albega(0, i, 0) = -w(i, 1, 2) / w(i, 0, 2)
albega(0, i, 1) = 0
albega(1, i, 0) = 0
albega(1, i, 1) = 0
albega(2, i, 0) = (lmd(i, 2, 0) - w(i, 2, 2)) / w(i, 0, 2)
albega(2, i, 1) = lmd(i, 2, 1) / w(i, 0, 2)
Else
'beta skaičiavimas
laik3 = w(i, 1, 1) - lmd(i, 1, 0) - w(i, 0, 1) * albega(0, i, 0)
Dim laik4 = lmd(i, 1, 1) + w(i, 0, 1) * albega(0, i, 1)
albega(1, i, 0) = (laik1 * laik3 - laik2 * laik4) / bsk
albega(1, i, 1) = (laik1 * (-laik4) - laik2 * laik3) / bsk
'gama skaičiavimas
laik1 = 2 * w(i, 0, 2)
'realioji gama dalis
albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0)
- w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / laik1
'kompleksinė gama dalis
albega(2, i, 1) = (lmd(i, 2, 1) - lmd(i, 0, 1) - w(i, 0, 1)
* albega(0, i, 1) - w(i, 1, 2) * albega(1, i, 1)) / laik1
End If

```

```

'jei tik pirmas vardiklis lygus nuliui
ElseIf ava = 0 And bva <> 0 And w(i, 0, 2) <> 0 Then
'beta skaičiavimas
bsk = w(i, 2, 2) * w(i, 0, 1) - w(i, 2, 1) * w(i, 0, 2) - w(i, 0, 1) * lmd(i, 2, 0)
laik1 = w(i, 0, 2) * lmd(i, 2, 1)
laik2 = bva * bva + laik1 * laik1
'realioji beta dalis
albega(1, i, 0) = (bsk * bva + w(i, 0, 1) * laik1 * lmd(i, 2, 1)) / laik2
'kompleksinė beta dalis
albega(1, i, 1) = ((bsk * w(i, 0, 2) - bva * w(i, 0, 1)) * lmd(i, 2, 1)) / laik2
'tikrinam ar alfa vardiklis lygus nuliui
laik1 = w(i, 0, 2) * albega(1, i, 0) - w(i, 0, 1)
laik2 = w(i, 0, 2) * albega(1, i, 1)
bsk = laik1 * laik1 + laik2 * laik2
If bsk = 0 Then
albega(0, i, 0) = 0
albega(0, i, 1) = 0
albega(1, i, 0) = w(i, 0, 1) / w(i, 0, 2)
albega(1, i, 1) = 0
albega(2, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 2)
albega(2, i, 1) = -lmd(i, 0, 1) / w(i, 0, 2)
Else
'alfa skaičiavimas
laik3 = w(i, 1, 1) - lmd(i, 1, 0) - w(i, 1, 2) * albega(1, i, 0)
Dim laik4 = lmd(i, 1, 1) + w(i, 1, 2) * albega(1, i, 1)
albega(0, i, 0) = (laik1 * laik3 - laik2 * laik4) / bsk
albega(0, i, 1) = (laik1 * (-laik4) - laik2 * laik3) / bsk
'gama skaičiavimas
laik1 = 2 * w(i, 0, 2)
'realioji gama dalis
albega(2, i, 0) = (lmd(i, 2, 0) - lmd(i, 0, 0) - w(i, 2, 2) + w(i, 0, 0)
- w(i, 0, 1) * albega(0, i, 0) - w(i, 1, 2) * albega(1, i, 0)) / laik1
'kompleksinė gama dalis
albega(2, i, 1) = (lmd(i, 2, 1) - lmd(i, 0, 1) - w(i, 0, 1)
* albega(0, i, 1) - w(i, 1, 2) * albega(1, i, 1)) / laik1
End If
ElseIf ava = 0 And bva = 0 And w(i, 0, 2) <> 0 Then
albega(0, i, 0) = 0
albega(1, i, 0) = 0
albega(0, i, 1) = 0
albega(1, i, 1) = 0
albega(2, i, 0) = 0
albega(2, i, 1) = 0
ElseIf ava = 0 And bva = 0 And w(i, 0, 2) = 0 Then
'pirmas atvejis
If w(i, 0, 2) = 0 And w(i, 0, 1) = 0 And w(i, 1, 2) <> 0 Then
albega(1, i, 0) = (w(i, 1, 1) - lmd(i, 1, 0)) / w(i, 1, 2)
albega(1, i, 1) = -lmd(i, 1, 1) / w(i, 1, 2)
laik1 = w(i, 0, 0) * w(i, 0, 0) - w(i, 0, 0) * w(i, 1, 1) - w(i, 1, 2)
* w(i, 2, 1) + w(i, 1, 1) * w(i, 2, 2) - w(i, 0, 0) * w(i, 2, 2)
If laik1 = 0 Then
albega(0, i, 0) = 0
Else
albega(0, i, 0) = (w(i, 1, 0) * w(i, 2, 2) - w(i, 1, 2) * w(i, 2, 0)
- w(i, 0, 0) * w(i, 1, 0)) / laik1
End If
albega(0, i, 1) = 0
albega(2, i, 0) = (w(i, 1, 0) + albega(0, i, 0) * (w(i, 0, 0) - w(i, 1, 1)))
/ w(i, 1, 2) + (albega(0, i, 0) * albega(1, i, 0)) / 2
albega(2, i, 1) = (albega(0, i, 0) * albega(1, i, 1)) / 2
'antras atvejis
ElseIf w(i, 0, 2) = 0 And w(i, 0, 1) <> 0 And w(i, 1, 2) = 0 Then
albega(0, i, 0) = (w(i, 0, 0) - lmd(i, 0, 0)) / w(i, 0, 1)
albega(0, i, 1) = -lmd(i, 0, 0) / w(i, 0, 1)
laik1 = w(i, 0, 0) * w(i, 2, 2) - w(i, 0, 0) * w(i, 1, 1) + w(i, 0, 1)
* w(i, 1, 0) + w(i, 1, 1) * w(i, 2, 2) - w(i, 2, 2) * w(i, 2, 2)
If laik1 = 0 Then
albega(1, i, 0) = 0

```

```

Else
    albeg(1, i, 0) = (w(i, 0, 0) * w(i, 2, 1) - w(i, 0, 1) * w(i, 2, 0)
                    - w(i, 2, 1) * w(i, 2, 2)) / laik1
End If
albeg(1, i, 1) = 0
albeg(2, i, 0) = ((w(i, 2, 2) - w(i, 1, 1)) * albeg(1, i, 0) - w(i, 2, 1))
                / w(i, 0, 1) - (albeg(0, i, 0) * albeg(1, i, 0)) / 2
albeg(2, i, 1) = -(albeg(0, i, 0) * albeg(1, i, 1)) / 2
'trečias atvejis
ElseIf w(i, 0, 2) = 0 And w(i, 0, 1) = 0 And w(i, 1, 2) = 0 Then
    laik1 = w(i, 1, 1) - w(i, 0, 0)
    laik2 = w(i, 2, 2) - w(i, 1, 1)
    laik3 = w(i, 2, 2) - w(i, 0, 0)
    If laik1 <> 0 And laik2 <> 0 And laik3 <> 0 Then
        albeg(0, i, 0) = w(i, 1, 0) / laik1
        albeg(0, i, 1) = 0
        albeg(1, i, 0) = w(i, 2, 1) / laik2
        albeg(1, i, 1) = 0
        albeg(2, i, 0) = (0.5 * albeg(0, i, 0) * albeg(1, i, 0) * (w(i, 0, 0)
                        - 2 * w(i, 1, 1) + w(i, 2, 2)) + w(i, 2, 0) - albeg(0, i, 0)
                        * w(i, 2, 1) + albeg(1, i, 0) * w(i, 1, 0)) / laik3
        albeg(2, i, 1) = 0
    ElseIf laik1 = 0 And laik2 <> 0 And laik3 <> 0 Then
        albeg(0, i, 0) = 0
        albeg(0, i, 1) = 0
        albeg(1, i, 0) = w(i, 2, 1) / laik2
        albeg(1, i, 1) = 0
        albeg(2, i, 0) = (w(i, 2, 0) + w(i, 1, 0) * albeg(1, i, 0)) / laik3
        albeg(2, i, 1) = 0
    ElseIf laik1 <> 0 And laik2 = 0 And laik3 <> 0 Then
        albeg(0, i, 0) = w(i, 1, 0) / laik1
        albeg(0, i, 1) = 0
        albeg(1, i, 0) = 0
        albeg(1, i, 1) = 0
        albeg(2, i, 0) = (w(i, 2, 0) - w(i, 2, 1) * albeg(0, i, 0)) / laik3
        albeg(2, i, 1) = 0
    ElseIf laik1 <> 0 And laik2 <> 0 And laik3 = 0 Then
        albeg(0, i, 0) = w(i, 1, 0) / laik1
        albeg(0, i, 1) = 0
        albeg(1, i, 0) = w(i, 2, 1) / laik2
        albeg(1, i, 1) = 0
        albeg(2, i, 0) = 0
        albeg(2, i, 1) = 0
    ElseIf laik1 = 0 And laik2 = 0 And laik3 = 0 Then
        'turi tenkinti lygybę a32*alfa-a21*beta = a31
        If w(i, 2, 0) <> 0 And w(i, 2, 1) <> 0 Then
            albeg(0, i, 0) = w(i, 2, 0) / w(i, 2, 1)
            albeg(1, i, 0) = 0
        ElseIf w(i, 2, 0) <> 0 And w(i, 1, 0) <> 0 Then
            albeg(0, i, 0) = 0
            albeg(1, i, 0) = -w(i, 2, 0) / w(i, 1, 0)
        Else
            albeg(0, i, 0) = 0
            albeg(1, i, 0) = 0
        End If
        albeg(0, i, 1) = 0
        albeg(1, i, 1) = 0
        albeg(2, i, 0) = 0
        albeg(2, i, 1) = 0
    End If
End If
End If
End If
Next
alfabetagama = albeg
End Function

```



```

Public Function matricaT(ByVal mk, ByVal alfa, ByVal beta, ByVal gama) As Double(,,)
    Dim t(mk, 2, 2, 1) As Double
    For i = 0 To mk
        t(i, 0, 0, 0) = 1
        t(i, 1, 1, 0) = 1
        t(i, 2, 2, 0) = 1
        t(i, 1, 0, 0) = alfa(i, 0)
        t(i, 1, 0, 1) = alfa(i, 1)
        t(i, 2, 1, 0) = beta(i, 0)
        t(i, 2, 1, 1) = beta(i, 1)
        t(i, 2, 0, 0) = 0.5 * (alfa(i, 0) * beta(i, 0) - alfa(i, 1) * beta(i, 1)) + gama(i, 0)
        t(i, 2, 0, 1) = 0.5 * (alfa(i, 0) * beta(i, 1) + alfa(i, 1) * beta(i, 0)) + gama(i, 1)
    Next
    matricaT = t
End Function

Public Function matricaTa(ByVal mk, ByVal alfa, ByVal beta, ByVal gama) As Double(,,)
    Dim ta(mk, 2, 2, 1) As Double
    For i = 0 To mk
        ta(i, 0, 0, 0) = 1
        ta(i, 1, 1, 0) = 1
        ta(i, 2, 2, 0) = 1
        ta(i, 1, 0, 0) = -alfa(i, 0)
        ta(i, 1, 0, 1) = -alfa(i, 1)
        ta(i, 2, 1, 0) = -beta(i, 0)
        ta(i, 2, 1, 1) = -beta(i, 1)
        ta(i, 2, 0, 0) = 0.5 * (alfa(i, 0) * beta(i, 0) - alfa(i, 1) * beta(i, 1)) - gama(i, 0)
        ta(i, 2, 0, 1) = 0.5 * (alfa(i, 0) * beta(i, 1) + alfa(i, 1) * beta(i, 0)) - gama(i, 1)
    Next
    matricaTa = ta
End Function

Public Function abcreiksmes(ByVal mk, ByVal alfa, ByVal beta, ByVal gama, ByVal w, ByVal t,
ByVal ta) As Double(,,)
    Dim tarp(2, 2) As Double
    Dim daug1(2, 2) As Double
    Dim daug2(2, 2) As Double
    Dim ktarp(2, 2, 1) As Double
    Dim kdaug1(2, 2, 1) As Double
    Dim kdaug2(2, 2, 1) As Double
    Dim wn(mk, 2, 2, 1) As Double

    'Daugyba T*W*T^-1
    For i = 0 To mk
        If alfa(i, 1) = 0 And beta(i, 1) = 0 And gama(i, 1) = 0 Then
            'T*W
            For j = 0 To 2
                For l = 0 To 2
                    daug1(j, l) = t(i, j, l, 0)
                    daug2(j, l) = w(i, j, l)
                Next
            Next
            tarp = daugyba(daug1, daug2)
            'T*W*T^-1
            daug1 = tarp
            For j = 0 To 2
                For l = 0 To 2
                    daug2(j, l) = ta(i, j, l, 0)
                Next
            Next
            tarp = daugyba(daug1, daug2)

            For j = 0 To 2
                For l = 0 To 2
                    wn(i, j, l, 0) = tarp(j, l)
                    wn(i, j, l, 1) = 0
                Next
            Next
        End If
    Next
End Function

```

```

Else
    'T*W
    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                kdaug1(j, l, m) = t(i, j, l, m)
            Next
        Next
    Next

    For j = 0 To 2
        For l = 0 To 2
            kdaug2(j, l, 0) = w(i, j, l)
            kdaug2(j, l, 1) = 0
        Next
    Next
    ktarp = kdaugyba(kdaug1, kdaug2)

    'Sandauga T*W*T^-1
    kdaug1 = ktarp
    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                kdaug2(j, l, m) = ta(i, j, l, m)
            Next
        Next
    Next
    ktarp = kdaugyba(kdaug1, kdaug2)

    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                wn(i, j, l, m) = ktarp(j, l, m)
            Next
        Next
    Next
End If
Next
abcreiksmes = wn
End Function

Public Function alfbetgam(ByVal mk, ByVal lmd, ByVal a, ByVal b, ByVal c) As Double(,,)
    Dim vard As Double
    Dim v(1) As Double
    Dim d(4, 1) As Double
    Dim la1(1) As Double
    Dim albg(2, mk, 1) As Double
    Dim sk(1) As Double

    For i = 0 To mk
        'Pirmoji situacija (jei skirtingos tikrinės reikšmės)
        If (lmd(i, 0, 0) <> lmd(i, 1, 0) Or lmd(i, 0, 1) <> lmd(i, 1, 1)) And (lmd(i, 0, 0) <>
lmd(i, 2, 0) Or lmd(i, 0, 1) <> lmd(i, 2, 1)) And (lmd(i, 1, 0) <> lmd(i, 2, 0) Or lmd(i, 1, 1) <>
lmd(i, 2, 1)) Then
            'alfa skaičiavimas
            If a(i, 1) = 0 And lmd(i, 0, 1) = 0 And lmd(i, 1, 1) = 0 Then
                albg(0, i, 0) = a(i, 0) / (lmd(i, 0, 0) - lmd(i, 1, 0))
                albg(0, i, 1) = 0
            Else
                v(0) = lmd(i, 0, 0) - lmd(i, 1, 0)
                v(1) = lmd(i, 0, 1) - lmd(i, 1, 1)
                vard = v(0) * v(0) + v(1) * v(1)
                albg(0, i, 0) = (a(i, 0) * v(0) + a(i, 1) * v(1)) / vard
                albg(0, i, 1) = (a(i, 1) * v(0) - a(i, 0) * v(1)) / vard
            End If
            'beta skaičiavimas
            If b(i, 1) = 0 And lmd(i, 1, 1) = 0 And lmd(i, 2, 1) = 0 Then
                albg(1, i, 0) = b(i, 0) / (lmd(i, 1, 0) - lmd(i, 2, 0))
            End If
        End If
    Next
End Function

```

```

    albg(1, i, 1) = 0
Else
    v(0) = lmd(i, 1, 0) - lmd(i, 2, 0)
    v(1) = lmd(i, 1, 1) - lmd(i, 2, 1)
    vard = v(0) * v(0) + v(1) * v(1)
    albg(1, i, 0) = (b(i, 0) * v(0) + b(i, 1) * v(1)) / vard
    albg(1, i, 1) = (b(i, 1) * v(0) - b(i, 0) * v(1)) / vard
End If
'gama skaičiavimas
If a(i, 1) = 0 And b(i, 1) = 0 And c(i, 1) = 0 And lmd(i, 0, 1) = 0
    And lmd(i, 1, 1) = 0 And lmd(i, 2, 1) = 0 Then
    d(0, 0) = albg(0, i, 0) * albg(1, i, 0) * (2 * lmd(i, 1, 0) - lmd(i, 0, 0)
        - lmd(i, 2, 0))
    d(0, 1) = 2 * (-c(i, 0) + albg(1, i, 0) * a(i, 0) - albg(0, i, 0) * b(i, 0))
    vard = 2 * (lmd(i, 2, 0) - lmd(i, 0, 0))
    albg(2, i, 0) = (d(0, 0) + d(0, 1)) / vard
    albg(2, i, 1) = 0
Else
    d(0, 0) = albg(0, i, 0) * albg(1, i, 0) - albg(0, i, 1) * albg(1, i, 1)
    d(0, 1) = albg(0, i, 0) * albg(1, i, 1) + albg(0, i, 1) * albg(1, i, 0)
    d(1, 0) = 2 * lmd(i, 1, 0) - lmd(i, 0, 0) - lmd(i, 2, 0)
    d(1, 1) = 2 * lmd(i, 1, 1) - lmd(i, 0, 1) - lmd(i, 2, 1)
    d(2, 0) = d(0, 0) * d(1, 0) - d(0, 1) * d(1, 1)
    d(2, 1) = d(0, 0) * d(1, 1) + d(0, 1) * d(1, 0)
    '+
    d(0, 0) = albg(1, i, 0) * a(i, 0) - albg(1, i, 1) * a(i, 1)
    d(0, 1) = albg(1, i, 0) * a(i, 1) + albg(1, i, 1) * a(i, 0)
    d(1, 0) = albg(0, i, 0) * b(i, 0) - albg(0, i, 1) * b(i, 1)
    d(1, 1) = albg(0, i, 0) * b(i, 1) + albg(0, i, 1) * b(i, 0)
    d(3, 0) = 2 * (-c(i, 0) + d(0, 0) - d(1, 0))
    d(3, 1) = 2 * (-c(i, 1) + d(0, 1) - d(1, 1))
    '=
    d(4, 0) = d(2, 0) + d(3, 0)
    d(4, 1) = d(2, 1) + d(3, 1)
    'vardiklis
    v(0) = 2 * (lmd(i, 2, 0) - lmd(i, 0, 0))
    v(1) = 2 * (lmd(i, 2, 1) - lmd(i, 0, 1))
    vard = v(0) * v(0) + v(1) * v(1)
    albg(2, i, 0) = (d(4, 0) * v(0) + d(4, 1) * v(1)) / vard
    albg(2, i, 1) = (d(4, 1) * v(0) - d(4, 0) * v(1)) / vard
End If
'jei lmd1=lmd2<>lmd3
ElseIf lmd(i, 0, 0) = lmd(i, 1, 0) And lmd(i, 0, 1) = lmd(i, 1, 1)
    And lmd(i, 0, 0) <> lmd(i, 2, 0) And lmd(i, 0, 1) <> lmd(i, 2, 1) Then
    albg(0, i, 0) = 0
    albg(0, i, 1) = 0
    v(0) = lmd(i, 0, 0) - lmd(i, 2, 0)
    v(1) = lmd(i, 0, 1) - lmd(i, 2, 1)
    vard = v(0) * v(0) + v(1) * v(1)
    'beta
    albg(1, i, 0) = (b(i, 0) * v(0) + b(i, 1) * v(1)) / vard
    albg(1, i, 1) = (b(i, 1) * v(0) - b(i, 0) * v(1)) / vard
    'gama
    sk(0) = c(i, 0) - albg(1, i, 0) * a(i, 0) + albg(1, i, 1) * a(i, 1)
    sk(1) = c(i, 1) - albg(1, i, 0) * a(i, 1) - albg(1, i, 1) * a(i, 0)
    albg(2, i, 0) = (sk(0) * v(0) + sk(1) * v(1)) / vard
    albg(2, i, 1) = (sk(1) * v(0) - sk(0) * v(1)) / vard
    'jei lmd1<>lmd2=lmd3
ElseIf lmd(i, 2, 0) = lmd(i, 1, 0) And lmd(i, 2, 1) = lmd(i, 1, 1)
    And lmd(i, 0, 0) <> lmd(i, 2, 0) And lmd(i, 0, 1) <> lmd(i, 2, 1) Then
    albg(1, i, 0) = 0
    albg(1, i, 1) = 0
    v(0) = lmd(i, 0, 0) - lmd(i, 1, 0)
    v(1) = lmd(i, 0, 1) - lmd(i, 1, 1)
    vard = v(0) * v(0) + v(1) * v(1)
    'alfa
    albg(0, i, 0) = (a(i, 0) * v(0) + a(i, 1) * v(1)) / vard
    albg(0, i, 1) = (a(i, 1) * v(0) - a(i, 0) * v(1)) / vard

```

```

'gama
v(0) = albg(0, i, 1) * b(i, 1) - albg(0, i, 0) * b(i, 0) - c(i, 0)
v(1) = -albg(0, i, 1) * b(i, 0) - albg(0, i, 0) * b(i, 1) - c(i, 1)
albg(2, i, 0) = (sk(0) * v(0) + sk(1) * v(1)) / vard
albg(2, i, 1) = (sk(1) * v(0) - sk(0) * v(1)) / vard
'jei visos tikrinės reikšmės yra lygios
Else
albg(2, i, 0) = 0
albg(2, i, 1) = 0
If a(i, 0) <> 0 Or a(i, 1) <> 0 Then
albg(0, i, 0) = 0
albg(0, i, 1) = 0
vard = a(i, 0) * a(i, 0) + a(i, 1) * a(i, 1)
albg(1, i, 0) = (a(i, 0) * c(i, 0) + a(i, 1) * c(i, 1)) / vard
albg(1, i, 1) = (a(i, 0) * c(i, 1) - a(i, 1) * c(i, 0)) / vard
ElseIf b(i, 0) <> 0 Or b(i, 1) <> 0 Then
albg(1, i, 0) = 0
albg(1, i, 1) = 0
vard = -(b(i, 0) * b(i, 0) + b(i, 1) * b(i, 1))
albg(0, i, 0) = (b(i, 0) * c(i, 0) + b(i, 1) * c(i, 1)) / vard
albg(0, i, 1) = (b(i, 0) * c(i, 1) - b(i, 1) * c(i, 0)) / vard
Else
albg(0, i, 0) = 0
albg(0, i, 1) = 0
albg(1, i, 0) = 0
albg(1, i, 1) = 0
End If
End If
Next
alfbetgam = albg
End Function

Public Function matricaK(ByVal mk, ByVal alf, ByVal bet, ByVal gam) As Double(,,,)
Dim K(mk, 2, 2, 1) As Double
Dim v(1) As Double
'K matrica
For i = 0 To mk
'Jeigu realūs skaičiai
If alf(i, 1) = 0 And bet(i, 1) = 0 And gam(i, 1) = 0 Then
For j = 0 To 2
For l = 0 To 2
K(i, j, l, 1) = 0
If j = l Then
K(i, j, l, 0) = 1
End If
Next
Next
K(i, 0, 1, 0) = alf(i, 0)
K(i, 1, 2, 0) = bet(i, 0)
K(i, 0, 2, 0) = (alf(i, 0) * bet(i, 0)) / 2 + gam(i, 0)
'Jeigu kompleksiniai skaičiai
Else
K(i, 0, 0, 0) = 1
K(i, 1, 1, 0) = 1
K(i, 2, 2, 0) = 1

v(0) = (alf(i, 0) * bet(i, 0) - alf(i, 1) * bet(i, 1)) / 2
v(1) = (alf(i, 0) * bet(i, 1) + alf(i, 1) * bet(i, 0)) / 2

For j = 0 To 1
K(i, 0, 1, j) = alf(i, j)
K(i, 1, 2, j) = bet(i, j)
K(i, 0, 2, j) = v(j) + gam(i, j)
Next
End If
Next
matricaK = K
End Function

```

```

Public Function matricaaK(ByVal mk, ByVal alf, ByVal bet, ByVal gam) As Double(,,)
    Dim aK(mk, 2, 2, 1) As Double
    Dim v(1) As Double

    'K matrica
    For i = 0 To mk
        'Jeigu realūs skaičiai
        If alf(i, 1) = 0 And bet(i, 1) = 0 And gam(i, 1) = 0 Then
            For j = 0 To 2
                For l = 0 To 2
                    aK(i, j, l, 1) = 0
                    If j = 1 Then
                        aK(i, j, l, 0) = 1
                    End If
                Next
            Next
            aK(i, 0, 1, 0) = -alf(i, 0)
            aK(i, 1, 2, 0) = -bet(i, 0)
            aK(i, 0, 2, 0) = (alf(i, 0) * bet(i, 0)) / 2 - gam(i, 0)

            'Jeigu kompleksiniai skaičiai
        Else
            aK(i, 0, 0, 0) = 1
            aK(i, 1, 1, 0) = 1
            aK(i, 2, 2, 0) = 1

            v(0) = (alf(i, 0) * bet(i, 0) - alf(i, 1) * bet(i, 1)) / 2
            v(1) = (alf(i, 0) * bet(i, 1) + alf(i, 1) * bet(i, 0)) / 2

            For j = 0 To 1
                aK(i, 0, 1, j) = -alf(i, j)
                aK(i, 1, 2, j) = -bet(i, j)
                aK(i, 0, 2, j) = v(j) - gam(i, j)
            Next
        End If
    Next
    matricaaK = aK
End Function

Public Function daugyba_3m_4d(ByVal mk, ByVal lmd, ByVal pim, ByVal anm, ByVal trm) As Double(,,)
    Dim daug1(2, 2) As Double
    Dim daug2(2, 2) As Double
    Dim tarp(2, 2) As Double
    Dim kdaug1(2, 2, 1) As Double
    Dim kdaug2(2, 2, 1) As Double
    Dim ktarp(2, 2, 1) As Double
    Dim rez(mk, 2, 2, 1) As Double

    'pim*anm*trm
    For i = 0 To mk
        If lmd(i, 0, 1) = 0 And lmd(i, 1, 1) = 0 And lmd(i, 2, 1) = 0 Then
            'pim*anm
            daug1 = tarp
            For j = 0 To 2
                For l = 0 To 2
                    daug1(j, l) = pim(i, j, l, 0)
                    daug2(j, l) = anm(i, j, l, 0)
                Next
            Next
            tarp = daugyba(daug1, daug2)

            'pim*anm*trm
            daug1 = tarp
            For j = 0 To 2
                For l = 0 To 2
                    daug2(j, l) = trm(i, j, l, 0)
                Next
            Next
        End If
    Next
    rez = kdaug1(daug1, kdaug2, ktarp)
End Function

```

```

        Next
    Next
    tarp = daugyba(daug1, daug2)

    For j = 0 To 2
        For l = 0 To 2
            rez(i, j, l, 0) = tarp(j, l)
            rez(i, j, l, 1) = 0
        Next
    Next
Else
    'pim*anm
    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                kdaug1(j, l, m) = pim(i, j, l, m)
                kdaug2(j, l, m) = anm(i, j, l, m)
            Next
        Next
    Next
    ktarp = kdaugyba(kdaug1, kdaug2)
    'Ka*Lm*K
    kdaug1 = ktarp
    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                kdaug2(j, l, m) = trm(i, j, l, m)
            Next
        Next
    Next
    ktarp = kdaugyba(kdaug1, kdaug2)

    For j = 0 To 2
        For l = 0 To 2
            For m = 0 To 1
                rez(i, j, l, m) = ktarp(j, l, m)
            Next
        Next
    Next
End If
Next
daugyba_3m_4d = rez
End Function

Private Sub Btn_1matr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_1matr.Click
    viena_matrica.Show()
End Sub

Private Sub Btn_eilutes_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_eilutes.Click
    eilutes.Show()
End Sub
End Class

Public Class viena_matrica
    Inherits System.Windows.Forms.Form
    Public mycaller As pradine

    Dim mk As Integer = 0
    Dim Am(2, 2) As Double
    Dim Bm(mk, 2, 2, 1) As Double
    Dim lm(mk, 2, 1) As Double
    Dim alfa(mk, 1) As Double
    Dim beta(mk, 1) As Double
    Dim gama(mk, 1) As Double
    Dim a(mk, 1) As Double
    Dim b(mk, 1) As Double

```

```

Dim c(mk, 1) As Double
Dim t(mk, 2, 2, 1) As Double
Dim ta(mk, 2, 2, 1) As Double
Dim alf(mk, 1) As Double
Dim bet(mk, 1) As Double
Dim gam(mk, 1) As Double
Dim Km(mk, 2, 2, 1) As Double
Dim aKm(mk, 2, 2, 1) As Double

'Vienos matricos įvedimas
Private Sub Btn_ivmatr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_ivmatr.Click
    Dim tekst As String
    tekst = Replace(a00.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(0, 0) = CDb1(Val(tekst))
    End If

    tekst = Replace(a01.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(0, 1) = CDb1(Val(tekst))
    End If

    tekst = Replace(a02.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(0, 2) = CDb1(Val(tekst))
    End If

    tekst = Replace(a10.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(1, 0) = CDb1(Val(tekst))
    End If

    tekst = Replace(a11.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(1, 1) = CDb1(Val(tekst))
    End If

    tekst = Replace(a12.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(1, 2) = CDb1(Val(tekst))
    End If

    tekst = Replace(a20.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(2, 0) = CDb1(Val(tekst))
    End If

    tekst = Replace(a21.Text, ",", ".")
    If tekst = Nothing Or IsNumeric(tekst) = False Then
        MessageBox.Show("Įveskite tinkamas reikšmes")
    Else
        Am(2, 1) = CDb1(Val(tekst))
    End If

```

```

tekst = Replace(a22.Text, ",", ".")
If tekst = Nothing Or IsNumeric(tekst) = False Then
    MessageBox.Show("Įveskite tinkamas reikšmes")
Else
    Am(2, 2) = CDb1(Val(tekst))
End If

Btn_vm_koef.Enabled = True
End Sub

'Koeficientų skaičiavimas
Private Sub Btn_vm_koef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_vm_koef.Click
    Dim an(1, 2, 2) As Double
    For i = 0 To 2
        For j = 0 To 2
            an(i, j) = Am(i, j)
        Next
    Next

    'Tikrinės reikšmės
    Dim lmda(0, 2, 1) As Double
    lmda = pradine.Tikrines(mk, an)

    'Tikrinių reikšmių rikiavimas
    lm = pradine.rikiuoti2(mk, lmda)
    Text_vm_rez.Text = "Tikrinės reikšmės:" & vbCrLf
    Text_vm_rez.Text = Text_vm_rez.Text & lm(0, 0, 0) & " + " & lm(0, 0, 1) & "i" & vbCrLf &
    lm(0, 1, 0) & " + " & lm(0, 1, 1) & "i" & vbCrLf & lm(0, 2, 0) & " + " & lm(0, 2, 1) & "i" & vbCrLf

    'Koeficientai alfa*, beta*, gama*
    Dim nlm(2, 2, 1) As Double
    nlm = pradine.alfabetagama(mk, an, lm)
    For i = 0 To 1
        alfa(0, i) = nlm(0, mk, i)
        beta(0, i) = nlm(1, mk, i)
        gama(0, i) = nlm(2, mk, i)
    Next

    Text_vm_rez.Text = Text_vm_rez.Text & vbCrLf & "Parametrų  $\alpha^*$ ,  $\beta^*$ ,  $\gamma^*$  reikšmės:" & vbCrLf &
    alfa(0, 0) & " + " & alfa(0, 1) & "i" & vbCrLf &
    beta(0, 0) & " + " & beta(0, 1) & "i" & vbCrLf &
    gama(0, 0) & " + " & gama(0, 1) & "i" & vbCrLf

    t = pradine.matricaT(mk, alfa, beta, gama)
    ta = pradine.matricaTa(mk, alfa, beta, gama)
    Bm = pradine.abcreiksmes(mk, alfa, beta, gama, an, t, ta)
    For j = 0 To 1
        a(0, j) = Bm(0, 0, 1, j)
        b(0, j) = Bm(0, 1, 2, j)
        c(0, j) = Bm(0, 0, 2, j)
    Next

    Text_vm_rez.Text = Text_vm_rez.Text & vbCrLf & "Parametrų a, b, c reikšmės:" & vbCrLf & a(0,
0) & " + " & a(0, 1) & "i" & vbCrLf &
    b(0, 0) & " + " & b(0, 1) & "i" & vbCrLf &
    c(0, 0) & " + " & c(0, 1) & "i" & vbCrLf
    Text_vm_rez.Text = Text_vm_rez.Text & vbCrLf & "Matrica T:" & vbCrLf &
    t(0, 0, 0, 0) & " + " & t(0, 0, 0, 1) & "i" & t(0, 0, 1, 0) & " + " & t(0, 0, 1, 1) &
    "i" & t(0, 0, 2, 0) & " + " & t(0, 0, 2, 1) & "i" & vbCrLf &
    t(0, 1, 0, 0) & " + " & t(0, 1, 0, 1) & "i" & t(0, 1, 1, 0) & " + " & t(0, 1, 1, 1) &
    "i" & t(0, 1, 2, 0) & " + " & t(0, 1, 2, 1) & "i" & vbCrLf &
    t(0, 2, 0, 0) & " + " & t(0, 2, 0, 1) & "i" & t(0, 2, 1, 0) & " + " & t(0, 2, 1, 1) & "i"
& t(0, 2, 2, 0) & " + " & t(0, 2, 2, 1) & "i" & vbCrLf & vbCrLf &
    "Matrica T^-1:" & vbCrLf &
    ta(mk, 0, 0, 0) & " + " & ta(mk, 0, 0, 1) & "i" & ta(mk, 0, 1, 0) & " + " & ta(mk, 0,
1, 1) & "i" & ta(mk, 0, 2, 0) & " + " & ta(mk, 0, 2, 1) & "i" & vbCrLf &

```



```

    ta(mk, 1, 0, 0) & " + " & ta(mk, 1, 0, 1) & "i " & ta(mk, 1, 1, 0) & " + " & ta(mk, 1, 1,
1) & "i " & ta(mk, 1, 2, 0) & " + " & ta(mk, 1, 2, 1) & "i" & vbCrLf &
    ta(mk, 2, 0, 0) & " + " & ta(mk, 2, 0, 1) & "i " & ta(mk, 2, 1, 0) & " + " & ta(mk, 2, 1,
1) & "i " & ta(mk, 2, 2, 0) & " + " & ta(mk, 2, 2, 1) & "i" & vbCrLf & vbCrLf &
    "Matrica B = T*A*T^-1:" & vbCrLf & Bm(mk, 0, 0, 0) & " + " & Bm(mk, 0, 0, 1) & "i " &
Bm(mk, 0, 1, 0) & " + " & Bm(mk, 0, 1, 1) & "i " & Bm(mk, 0, 2, 0) & " + " & Bm(mk, 0, 2, 1) & "i"
& vbCrLf &
    Bm(mk, 1, 0, 0) & " + " & Bm(mk, 1, 0, 1) & "i " & Bm(mk, 1, 1, 0) & " + " & Bm(mk, 1,
1, 1) & "i " & Bm(mk, 1, 2, 0) & " + " & Bm(mk, 1, 2, 1) & "i" & vbCrLf &
    Bm(mk, 2, 0, 0) & " + " & Bm(mk, 2, 0, 1) & "i " & Bm(mk, 2, 1, 0) & " + " & Bm(mk, 2, 1,
1) & "i " & Bm(mk, 2, 2, 0) & " + " & Bm(mk, 2, 2, 1) & "i" & vbCrLf

'randam alfa, beta, gama reikšmes
Dim ll(2, 1, 1) As Double
ll = pradine.alfbetgam(mk, lm, a, b, c)
For j = 0 To 1
    alf(mk, j) = ll(0, 0, j)
    bet(mk, j) = ll(1, 0, j)
    gam(mk, j) = ll(2, 0, j)
Next

Text_vm_rez.Text = Text_vm_rez.Text & vbCrLf & "Parametru α, β, γ reikšmės:" & vbCrLf &
alf(mk, 0) & " + " & alf(mk, 1) & "i" & vbCrLf &
    bet(mk, 0) & " + " & bet(mk, 1) & "i" & vbCrLf &
    gam(mk, 0) & " + " & gam(mk, 1) & "i" & vbCrLf

Km = pradine.matricaK(mk, alf, bet, gam)
aKm = pradine.matricaak(mk, alf, bet, gam)

Text_vm_rez.Text = Text_vm_rez.Text & vbCrLf & "K" & vbCrLf &
    Km(mk, 0, 0, 0) & " + " & Km(mk, 0, 0, 1) & "i " & Km(mk, 0, 1, 0) & " + " &
Km(mk, 0, 1, 1) & "i " & Km(mk, 0, 2, 0) & " + " & Km(mk, 0, 2, 1) & "i" & vbCrLf &
    Km(mk, 1, 0, 0) & " + " & Km(mk, 1, 0, 1) & "i " & Km(mk, 1, 1, 0) & " + " &
Km(mk, 1, 1, 1) & "i " & Km(mk, 1, 2, 0) & " + " & Km(mk, 1, 2, 1) & "i" & vbCrLf &
    Km(mk, 2, 0, 0) & " + " & Km(mk, 2, 0, 1) & "i " & Km(mk, 2, 1, 0) & " + " &
Km(mk, 2, 1, 1) & "i " & Km(mk, 2, 2, 0) & " + " & Km(mk, 2, 2, 1) & "i" & vbCrLf &
    vbCrLf & "K^-1" & vbCrLf &
    aKm(mk, 0, 0, 0) & " + " & aKm(mk, 0, 0, 1) & "i " & aKm(mk, 0, 1, 0) & " + " &
aKm(mk, 0, 1, 1) & "i " & aKm(mk, 0, 2, 0) & " + " & aKm(mk, 0, 2, 1) & "i" & vbCrLf &
    aKm(mk, 1, 0, 0) & " + " & aKm(mk, 1, 0, 1) & "i " & aKm(mk, 1, 1, 0) & " + " &
aKm(mk, 1, 1, 1) & "i " & aKm(mk, 1, 2, 0) & " + " & aKm(mk, 1, 2, 1) & "i" & vbCrLf &
    aKm(mk, 2, 0, 0) & " + " & aKm(mk, 2, 0, 1) & "i " & aKm(mk, 2, 1, 0) & " + " &
aKm(mk, 2, 1, 1) & "i " & aKm(mk, 2, 2, 0) & " + " & aKm(mk, 2, 2, 1) & "i" & vbCrLf
    Btn_patikra.Enabled = True
End Sub

Private Sub Btn_patikra_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_patikra.Click
    Dim nAm(mk, 2, 2, 1) As Double
    nAm = pradine.daugyba_3m_4d(mk, lm, ta, Bm, t)
    Text_vm_patikr.Text = "Matrica A* = T^-1*B*T:" & vbCrLf &
    nAm(mk, 0, 0, 0) & " + " & nAm(mk, 0, 0, 1) & "i " & nAm(mk, 0, 1, 0) & " + " & nAm(mk,
0, 1, 1) & "i " & nAm(mk, 0, 2, 0) & " + " & nAm(mk, 0, 2, 1) & "i" & vbCrLf &
    nAm(mk, 1, 0, 0) & " + " & nAm(mk, 1, 0, 1) & "i " & nAm(mk, 1, 1, 0) & " + " & nAm(mk,
1, 1, 1) & "i " & nAm(mk, 1, 2, 0) & " + " & nAm(mk, 1, 2, 1) & "i" & vbCrLf &
    nAm(mk, 2, 0, 0) & " + " & nAm(mk, 2, 0, 1) & "i " & nAm(mk, 2, 1, 0) & " + " & nAm(mk,
2, 1, 1) & "i " & nAm(mk, 2, 2, 0) & " + " & nAm(mk, 2, 2, 1) & "i" & vbCrLf

    Dim Lmm(mk, 2, 2, 1) As Double
    Lmm = pradine.daugyba_3m_4d(mk, lm, Km, Bm, aKm)
    Text_vm_patikr.Text = Text_vm_patikr.Text & vbCrLf & "Matrica Lm = K*B*K^-1:" & vbCrLf &
    Lmm(mk, 0, 0, 0) & " + " & Lmm(mk, 0, 0, 1) & "i " & Lmm(mk, 0, 1, 0) & " + " & Lmm(mk,
0, 1, 1) & "i " & Lmm(mk, 0, 2, 0) & " + " & Lmm(mk, 0, 2, 1) & "i" & vbCrLf &
    Lmm(mk, 1, 0, 0) & " + " & Lmm(mk, 1, 0, 1) & "i " & Lmm(mk, 1, 1, 0) & " + " & Lmm(mk,
1, 1, 1) & "i " & Lmm(mk, 1, 2, 0) & " + " & Lmm(mk, 1, 2, 1) & "i" & vbCrLf &
    Lmm(mk, 2, 0, 0) & " + " & Lmm(mk, 2, 0, 1) & "i " & Lmm(mk, 2, 1, 0) & " + " & Lmm(mk,
2, 1, 1) & "i " & Lmm(mk, 2, 2, 0) & " + " & Lmm(mk, 2, 2, 1) & "i" & vbCrLf

```

```

'Patikrinam ar B* = K^-1*Lm*K
Dim nBm(mk, 2, 2, 1) As Double
nBm = pradine.daugyba_3m_4d(mk, lm, aKm, Lmm, Km)
Text_vm_patikr.Text = Text_vm_patikr.Text & vbCrLf & "Matrica B* = K^-1*Lm*K:" & vbCrLf &
nBm(mk, 0, 0, 0) & " + " & nBm(mk, 0, 0, 1) & "i " & nBm(mk, 0, 1, 0) & " + " & nBm(mk,
0, 1, 1) & "i " & nBm(mk, 0, 2, 0) & " + " & nBm(mk, 0, 2, 1) & "i" & vbCrLf &
nBm(mk, 1, 0, 0) & " + " & nBm(mk, 1, 0, 1) & "i " & nBm(mk, 1, 1, 0) & " + " & nBm(mk,
1, 1, 1) & "i " & nBm(mk, 1, 2, 0) & " + " & nBm(mk, 1, 2, 1) & "i" & vbCrLf &
nBm(mk, 2, 0, 0) & " + " & nBm(mk, 2, 0, 1) & "i " & nBm(mk, 2, 1, 0) & " + " & nBm(mk,
2, 1, 1) & "i " & nBm(mk, 2, 2, 0) & " + " & nBm(mk, 2, 2, 1) & "i" & vbCrLf

'Patikrinam ar A = T^-1*K^-1*Lm*K*T
Dim nnAm(mk, 2, 2, 1) As Double
nnAm = pradine.daugyba_3m_4d(mk, lm, ta, Bm, t)
Text_vm_patikr.Text = Text_vm_patikr.Text & vbCrLf & "Matrica A** = T^-1*K^-1*Lm*K*T:" &
vbCrLf &
nnAm(mk, 0, 0, 0) & " + " & nnAm(mk, 0, 0, 1) & "i " & nnAm(mk, 0, 1, 0) & " + " &
nnAm(mk, 0, 1, 1) & "i " & nnAm(mk, 0, 2, 0) & " + " & nnAm(mk, 0, 2, 1) & "i" & vbCrLf &
nnAm(mk, 1, 0, 0) & " + " & nnAm(mk, 1, 0, 1) & "i " & nnAm(mk, 1, 1, 0) & " + " &
nnAm(mk, 1, 1, 1) & "i " & nnAm(mk, 1, 2, 0) & " + " & nnAm(mk, 1, 2, 1) & "i" & vbCrLf &
nnAm(mk, 2, 0, 0) & " + " & nnAm(mk, 2, 0, 1) & "i " & nnAm(mk, 2, 1, 0) & " + " &
nnAm(mk, 2, 1, 1) & "i " & nnAm(mk, 2, 2, 0) & " + " & nnAm(mk, 2, 2, 1) & "i" & vbCrLf

'Paklaidų matrica
Dim skirt(2, 2, 1)
For i = 0 To 2
    For j = 0 To 2
        skirt(i, j, 0) = Math.Abs(Am(i, j) - nnAm(0, i, j, 0))
        skirt(i, j, 1) = Math.Abs(nnAm(0, i, j, 1))
    Next
Next
Text_vm_patikr.Text = Text_vm_patikr.Text & vbCrLf & "Paklaidų matrica:" & vbCrLf &
skirt(0, 0, 0) & " + " & skirt(0, 0, 1) & "i " & skirt(0, 1, 0) & " + " & skirt(0, 1, 1) &
"i " & skirt(0, 2, 0) & " + " & skirt(0, 2, 1) & "i" & vbCrLf &
skirt(1, 0, 0) & " + " & skirt(1, 0, 1) & "i " & skirt(1, 1, 0) & " + " & skirt(1, 1, 1) &
"i " & skirt(1, 2, 0) & " + " & skirt(1, 2, 1) & "i" & vbCrLf &
skirt(2, 0, 0) & " + " & skirt(2, 0, 1) & "i " & skirt(2, 1, 0) & " + " & skirt(2, 1, 1) &
"i " & skirt(2, 2, 0) & " + " & skirt(2, 2, 1) & "i" & vbCrLf
End Sub
End Class
Imports System.Windows.Forms.DataVisualization.Charting

Public Class eilutes
    Inherits System.Windows.Forms.Form
    Public mycaller As pradine
    Public pubmk As Integer
    Dim pmk As Integer
    Dim mk As Integer = 3000
    Dim xx(mk) As Double
    Dim yy(mk) As Double
    Dim zz(mk) As Double
    Dim w(mk, 2, 2) As Double
    Dim lmd(mk, 2, 1) As Double
    Dim alfas(mk, 1) As Double
    Dim betas(mk, 1) As Double
    Dim gamas(mk, 1) As Double
    Dim alf(mk, 1) As Double
    Dim bet(mk, 1) As Double
    Dim gam(mk, 1) As Double
    Dim sintetines As Boolean = False

    'Grafikam skirtos reikšmės
    Public pxx(mk) As Double
    Public pyy(mk) As Double
    Public pzz(mk) As Double
    Public l1(mk) As Double
    Public l2(mk) As Double
    Public l3(mk) As Double

```

```

Public aalf(mk) As Double
Public avbet(mk) As Double
Public avgam(mk) As Double
Public lalf(mk) As Double
Public lavbet(mk) As Double
Public lavgam(mk) As Double
Public aalfas(mk) As Double
Public avbetas(mk) As Double
Public avgamas(mk) As Double
Public lalfas(mk) As Double
Public lavbetas(mk) As Double
Public lavgamas(mk) As Double

Private Sub Btn_iv_eilut_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_iv_eilut.Click
    Pan_ivest_eil.Visible = True
    Text_eilutes.Clear()
End Sub

Private Sub Btn_ivesti_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_ivesti.Click
    lentele_xyz.Rows.Clear()
    'Nuskaitome duomenis i lentele
    Dim visas_tekstas As String = Text_eilutes.Text
    visas_tekstas = Replace(visas_tekstas, ",", ".")

    If visas_tekstas = Nothing Then
        Return
    End If

    Dim eilutes() As String = visas_tekstas.Split(vbCrLf)
    Dim k As Integer = 0
    pmk = UBound(eilutes) - 1
    For i = 0 To pmk + 1
        If eilutes(i).Length < 3 Then
            Continue For
        End If
        Dim nariai() As String = eilutes(i).Split(vbTab)
        If nariai.Length < 3 Then
            Continue For
        ElseIf IsNumeric(nariai(0)) = False Or IsNumeric(nariai(1)) = False Or
IsNumeric(nariai(2)) = False Then
            Continue For
        Else
            lentele_xyz.Rows.Add()
            lentele_xyz.Rows(k).Cells(0).Value = k + 1
            lentele_xyz.Rows(k).Cells(1).Value = CDb1(Val(nariai(0)))
            lentele_xyz.Rows(k).Cells(2).Value = CDb1(Val(nariai(1)))
            lentele_xyz.Rows(k).Cells(3).Value = CDb1(Val(nariai(2)))
        End If
        k = k + 1
    Next

    n_sk.Text = k
    pmk = k - 1
    mk = pmk - 2
    If mk < 0 Then
        Btn_eil_koef.Enabled = False
        Btn_grafikai.Enabled = False
        Return
    End If

    'Įdedame reikšmes i masyva
    For i = 0 To pmk
        xx(i) = lentele_xyz.Rows(i).Cells(1).Value
        yy(i) = lentele_xyz.Rows(i).Cells(2).Value
        zz(i) = lentele_xyz.Rows(i).Cells(3).Value
    Next

```

```

'Sudarome matricas
mk = pmk - 2

For i = 1 To pmk - 1
    w(i - 1, 0, 0) = xx(i)
    w(i - 1, 0, 1) = xx(i + 1) - yy(i + 1)
    w(i - 1, 0, 2) = zz(i + 1) - xx(i + 1)
    w(i - 1, 1, 0) = xx(i - 1) - yy(i - 1)
    w(i - 1, 1, 1) = yy(i)
    w(i - 1, 1, 2) = yy(i + 1) - zz(i + 1)
    w(i - 1, 2, 0) = zz(i - 1) - xx(i - 1)
    w(i - 1, 2, 1) = yy(i - 1) - zz(i - 1)
    w(i - 1, 2, 2) = zz(i)
Next

sintetines = False
Btn_eil_koef.Enabled = True
Pan_ivest_eil.Visible = False
End Sub

Private Sub gener_xyz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
gener_xyz.Click
    lentele_xyz.Rows.Clear()
    If IsNumeric(n_sk_box.Text) = False Then
        Btn_eil_koef.Enabled = False
        Btn_grafikai.Enabled = False
        Return
    End If
    pmk = CInt(Int(n_sk_box.Text)) - 1
    If pmk <= 0 Then
        Btn_eil_koef.Enabled = False
        Btn_grafikai.Enabled = False
        Return
    End If

    mk = pmk - 2
    n_sk.Text = pmk + 1

    Dim ro1 As Double
    Dim ro2 As Double
    Dim ro3 As Double
    Dim fi1 As Double
    Dim fi2 As Double
    Dim fi3 As Double
    Dim k As Double

    k = Cdbl(Val(Replace(hbox.Text, ",", ".")))

    ro1 = Cdbl(Val(Replace(ro1box.Text, ",", ".")))
    fi1 = Cdbl(Val(Replace(fi1box.Text, ",", ".")))
    ro2 = Cdbl(Val(Replace(ro2box.Text, ",", ".")))
    fi2 = Cdbl(Val(Replace(fi2box.Text, ",", ".")))
    ro3 = Cdbl(Val(Replace(ro3box.Text, ",", ".")))
    fi3 = Cdbl(Val(Replace(fi3box.Text, ",", ".")))

'Sintetinių eilučių generavimas ir surašymas į lentelę
For i = 0 To pmk
    xx(i) = Math.Cos(ro1 * k * i + fi1)
    yy(i) = Math.Cos(ro2 * k * i + fi2)
    zz(i) = Math.Cos(ro3 * k * i + fi3)
    lentele_xyz.Rows.Add()
    lentele_xyz.Rows(i).Cells(0).Value = i + 1
    lentele_xyz.Rows(i).Cells(1).Value = xx(i)
    lentele_xyz.Rows(i).Cells(2).Value = yy(i)
    lentele_xyz.Rows(i).Cells(3).Value = zz(i)
Next

```

```

'Kauno matricų skaičiavimas
For i = 1 To pmk - 1
    w(i - 1, 0, 0) = xx(i)
    w(i - 1, 0, 1) = xx(i + 1) - yy(i + 1)
    w(i - 1, 0, 2) = zz(i + 1) - xx(i + 1)
    w(i - 1, 1, 0) = xx(i - 1) - yy(i - 1)
    w(i - 1, 1, 1) = yy(i)
    w(i - 1, 1, 2) = yy(i + 1) - zz(i + 1)
    w(i - 1, 2, 0) = zz(i - 1) - xx(i - 1)
    w(i - 1, 2, 1) = yy(i - 1) - zz(i - 1)
    w(i - 1, 2, 2) = zz(i)
Next

sintetines = True
Btn_eil_koef.Enabled = True
Pan_gener.Visible = False
End Sub

Public Sub Btn_eil_koef_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_eil_koef.Click
    lent_koef.Rows.Clear()

    'Tikrinės reikšmės
    Dim nlmd(mk, 2, 1) As Double
    nlmd = pradine.Tikrines(mk, w)
    'Rikiavimas

    lmd = pradine.rikiuoti2(mk, nlmd)
    'Spausdinimas
    For i = 0 To mk
        lent_koef.Rows.Add()
        lent_koef.Rows(i).Cells(0).Value = i + 1
        For j = 0 To 2
            If lmd(i, j, 1) = 0 Then
                lent_koef.Rows(i).Cells(j + 1).Value = lmd(i, j, 0)
            Else
                lent_koef.Rows(i).Cells(j + 1).Value=lmd(i,j,0).ToString+"&"+lmd(i,j,1).ToString+"&i"
            End If
        Next
    Next

    'Koeficientų alfa*, beta*, gama* apskaičiavimas
    Dim nlm(2, 2, 1) As Double
    nlm = pradine.alfabetagama(mk, w, lmd)

    For j = 0 To mk
        For i = 0 To 1
            alfas(j, i) = nlm(0, j, i)
            betas(j, i) = nlm(1, j, i)
            gamas(j, i) = nlm(2, j, i)
        Next
    Next

    'Koeficientų alfa*, beta*, gama* surašymas į lentelę
    spausdinti(4, alfas)
    spausdinti(5, betas)
    spausdinti(6, gamas)

    'Koeficientai a,b,c
    Dim a(mk, 1) As Double
    Dim b(mk, 1) As Double
    Dim c(mk, 1) As Double
    Dim t(mk, 2, 2, 1) As Double
    Dim ta(mk, 2, 2, 1) As Double
    Dim Bm(mk, 2, 2, 1) As Double
    t = pradine.matricaT(mk, alfas, betas, gamas)
    ta = pradine.matricaTa(mk, alfas, betas, gamas)
    Bm = pradine.abcreiksmes(mk, alfas, betas, gamas, w, t, ta)
    For i = 0 To mk

```

```

    For j = 0 To 1
        a(i, j) = Bm(i, 0, 1, j)
        b(i, j) = Bm(i, 1, 2, j)
        c(i, j) = Bm(i, 0, 2, j)
    Next
Next

'Koefficientų alfa, beta, gama apskaičiavimas
Dim lkl(2, 1, 1) As Double
lkl = pradine.alfbetgam(mk, lmd, a, b, c)
For i = 0 To mk
    For j = 0 To 1
        alf(i, j) = lkl(0, i, j)
        bet(i, j) = lkl(1, i, j)
        gam(i, j) = lkl(2, i, j)
    Next
Next

'Koefficientų alfa, beta, gama surašymas į lentelę
spausdinti(7, alf)
spausdinti(8, bet)
spausdinti(9, gam)

'grafikams ir dispersijoms skirtos reikšmės
For i = 0 To mk
    l1(i) = Math.Sqrt(lmd(i, 0, 0) * lmd(i, 0, 0) + lmd(i, 0, 1) * lmd(i, 0, 1))
    l2(i) = Math.Sqrt(lmd(i, 1, 0) * lmd(i, 1, 0) + lmd(i, 1, 1) * lmd(i, 1, 1))
    l3(i) = Math.Sqrt(lmd(i, 2, 0) * lmd(i, 2, 0) + lmd(i, 2, 1) * lmd(i, 2, 1))

    aalf(i) = Math.Sqrt(alf(i, 0) * alf(i, 0) + alf(i, 1) * alf(i, 1))
    avbet(i) = Math.Sqrt(bet(i, 0) * bet(i, 0) + bet(i, 1) * bet(i, 1))
    avgam(i) = Math.Sqrt(gam(i, 0) * gam(i, 0) + gam(i, 1) * gam(i, 1))

    lavalf(i) = Math.Log10(1 + aalf(i))
    lavbet(i) = Math.Log10(1 + avbet(i))
    lavgam(i) = Math.Log10(1 + avgam(i))

    aalfas(i) = Math.Sqrt(alfas(i, 0) * alfas(i, 0) + alfas(i, 1) * alfas(i, 1))
    avbetas(i) = Math.Sqrt(betas(i, 0) * betas(i, 0) + betas(i, 1) * betas(i, 1))
    avgamas(i) = Math.Sqrt(gamas(i, 0) * gamas(i, 0) + gamas(i, 1) * gamas(i, 1))

    lavalfas(i) = Math.Log10(1 + aalfas(i))
    lavbetas(i) = Math.Log10(1 + avbetas(i))
    lavgamas(i) = Math.Log10(1 + avgamas(i))
Next

Btn_grafikai.Enabled = True
If sintetines = False Then
    Btn_dispersijos.Enabled = True
    grafikai.Btn_laikai.Enabled = True
Else
    grafikai.Btn_laikai.Enabled = False
End If
End Sub

'Procedūra, atliekanti spausdinimą į tam tikrą lentelės stulpelį
Public Sub spausdinti(ByVal kel, ByVal kas)
    For i = 0 To mk
        If kas(i, 1) = 0 Then
            lent_koef.Rows(i).Cells(kel).Value = kas(i, 0)
        Else
            lent_koef.Rows(i).Cells(kel).Value = kas(i, 0).ToString + "+" + kas(i, 1).ToString + "i"
        End If
    Next
End Sub

Public Sub Btn_grafikai_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_grafikai.Click

```

```

pxx = xx
pyy = yy
pzz = zz
With grafikai.Chart1
    .Series.Clear()
    .ChartAreas(0).AxisX.Minimum = 0
    .ChartAreas(0).AxisX.Title = "t"
    If sintetines = False Then
        .ChartAreas(0).AxisY.Title = "RR, DQRS, DJT"
        .Series.Add("RR")
        .Series.Add("DQRS")
        .Series.Add("DJT")
        grafikai.Check_prad.Text = "RR, DQRS, DJT"
    Else
        .ChartAreas(0).AxisY.Title = "x(t), y(t), z(t)"
        .Series.Add("x")
        .Series.Add("y")
        .Series.Add("z")
        grafikai.Check_prad.Text = "x(t), y(t), z(t)"
    End If
    For i = 0 To pmk
        .Series(0).Points.AddXY(i + 1, xx(i))
        .Series(1).Points.AddXY(i + 1, yy(i))
        .Series(2).Points.AddXY(i + 1, zz(i))
    Next
    .ChartAreas(0).RecalculateAxesScale()
    For i = 0 To 2
        .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
    Next
End With
grafikai.Show()
End Sub

Private Sub Btn_cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_cancel.Click
    Pan_invest_eil.Visible = False
End Sub

'-----dispersijos-----
Public disp(9, 10, 11) As Double
Public dnr As Integer
Public dname(9) As String
Public mink As Integer
Public min(mk) As Double

Private Function stdev(ByVal masyvas, ByVal minutes, ByVal stulp_kiekis)
    Dim mas = masyvas
    Dim suma As Double = 0
    Dim m As Integer = mk
    Dim sm As Integer = stulp_kiekis + 1
    Dim vid(sm) As Double
    Dim elem As Integer = 0

    Dim kmin As Integer = 1
    Dim vard As Integer = 0

    For i = 0 To m
        If minutes(i) < kmin Then
            suma = suma + mas(i)
            vard = vard + 1
        Else
            vid(elem) = suma / vard
            elem = elem + 1
            kmin = kmin + 1
            suma = mas(i)
            vard = 1
        End If
    Next

```

```

vid(elem) = suma / vard

'sqrt dispersijos skaičiavimas
Dim skirt As Double
Dim disp(elem) As Double
kmin = 1

suma = 0
vard = 0
Dim delem As Integer = 0

For i = 0 To m
  If minutes(i) < kmin Then
    skirt = mas(i) - vid(delem)
    suma = suma + skirt * skirt
    vard = vard + 1
  Else
    disp(delem) = Math.Sqrt(suma / vard)
    delem = delem + 1
    skirt = mas(i) - vid(delem)
    suma = skirt * skirt
    kmin = kmin + 1
    vard = 1
  End If
  disp(delem) = Math.Sqrt(suma / vard)
Next

stdev = disp
End Function

Private Sub Btn_dispersijos_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_dispersijos.Click
  Panel_disp.Visible = True
End Sub

Private Sub Btn_disp_cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_disp_cancel.Click
  Panel_disp.Visible = False
End Sub

Private Sub Btn_dispsk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_dispsk.Click
  Dim visas_tekstas As String = disp_laik_box.Text
  visas_tekstas = Replace(visas_tekstas, ",", ".")

  If visas_tekstas = Nothing Then
    Return
  End If

  Dim nariai() As String = visas_tekstas.Split(vbCrLf)

  Dim k As Integer = 0
  Dim pmk = UBound(nariai)
  For i = 0 To pmk
    If nariai(i) = Nothing Then
      Continue For
    ElseIf IsNumeric(nariai(i)) = False Then
      Continue For
    Else
      min(k) = CDb1(Val(nariai(i)))
    End If
    k = k + 1
  Next
  Dim minkiek = k - 1

  If minkiek <> mk + 2 Then
    MessageBox.Show("Įveskite tinkamą kiekį laiko momentų")
  End If
End Sub

```



```

    Return
End If

mink = CInt(Val(min(mk + 2))) - 1

If disp_name_box.Text = Nothing Then
    MsgBox.Show("Įveskite pavadinimą")
    Return
Else
    dname(dnr) = disp_name_box.Text
End If

If mink > 10 Then
    MsgBox.Show("Įveskite tinkamą minučių kiekį")
    Return
End If

Dim ldisp(mink) As Double
'pirmų trijų koeficientų dipersijos
ldisp = stdev(avalfas, min, mink)
For i = 0 To mink
    disp(dnr, i, 0) = ldisp(i)
Next
ldisp = stdev(avbetas, min, mink)
For i = 0 To mink
    disp(dnr, i, 1) = ldisp(i)
Next
ldisp = stdev(avgamas, min, mink)
For i = 0 To mink
    disp(dnr, i, 2) = ldisp(i)
Next
'kitų trijų koeficientų dispersijos
ldisp = stdev(avalf, min, mink)
For i = 0 To mink
    disp(dnr, i, 3) = ldisp(i)
Next
ldisp = stdev(avbet, min, mink)
For i = 0 To mink
    disp(dnr, i, 4) = ldisp(i)
Next
ldisp = stdev(avgam, min, mink)
For i = 0 To mink
    disp(dnr, i, 5) = ldisp(i)
Next

'sulogaritmintų pirmų trijų koeficientų dipersijos
ldisp = stdev(lavalfas, min, mink)
For i = 0 To mink
    disp(dnr, i, 6) = ldisp(i)
Next
ldisp = stdev(lavbetas, min, mink)
For i = 0 To mink
    disp(dnr, i, 7) = ldisp(i)
Next
ldisp = stdev(lavgamas, min, mink)
For i = 0 To mink
    disp(dnr, i, 8) = ldisp(i)
Next
'sulogaritmintų kitų trijų koeficientų dispersijos
ldisp = stdev(lavalf, min, mink)
For i = 0 To mink
    disp(dnr, i, 9) = ldisp(i)
Next
ldisp = stdev(lavbet, min, mink)
For i = 0 To mink
    disp(dnr, i, 10) = ldisp(i)
Next
ldisp = stdev(lavgam, min, mink)

```

```

    For i = 0 To mink
        disp(dnr, i, 11) = ldisp(i)
    Next

    disp_name_box.Clear()
    disp_laik_box.Clear()
    dnr = dnr + 1
    dispersija.Show()
    Panel_disp.Visible = False
End Sub

Private Sub Btn_gen_eil_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_gen_eil.Click
    Pan_gener.Visible = True
End Sub
Private Sub Btn_cancel_gen_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_cancel_gen.Click
    Pan_gener.Visible = False
End Sub
End Class
Imports System.Windows.Forms.DataVisualization.Charting

Public Class grafikai
    Inherits System.Windows.Forms.Form
    Public mycaller As eilutes
    Dim mk As Integer = eilutes.n_sk.Text - 3
    Dim newline(mk + 2) As Double
    Dim npav As String
    Dim nmk As Integer
    Dim arpapildoma = False
    Dim arslenkantys = False
    Dim slmk As Integer

    'slenkantys vidurkiaiai
    Dim slxx(mk + 2) As Double
    Dim slyy(mk + 2) As Double
    Dim slzz(mk + 2) As Double
    Dim sll1(mk) As Double
    Dim sll2(mk) As Double
    Dim sll3(mk) As Double
    Dim slalfas(mk) As Double
    Dim slbetas(mk) As Double
    Dim slgamas(mk) As Double
    Dim slalf(mk) As Double
    Dim slbet(mk) As Double
    Dim slgam(mk) As Double
    Dim sllalfas(mk) As Double
    Dim sllbetas(mk) As Double
    Dim sllgamas(mk) As Double
    Dim sllalf(mk) As Double
    Dim sllbet(mk) As Double
    Dim sllgam(mk) As Double

    'laikai
    Dim lmom As Integer = 20
    Dim minutes(mk + 2) As Double
    Dim elementai(lmom) As Integer
    Dim laikoskale As Boolean = False

    'segmentiniai vidurkiaiai
    Dim segxx(lmom) As Double
    Dim segyy(lmom) As Double
    Dim segzz(lmom) As Double
    Dim segl1(lmom) As Double
    Dim segl2(lmom) As Double
    Dim segl3(lmom) As Double
    Dim segalfas(lmom) As Double
    Dim segbetas(lmom) As Double

```

```

Dim seggamas(lmom) As Double
Dim segalf(lmom) As Double
Dim segbet(lmom) As Double
Dim seggam(lmom) As Double
Dim seglalfas(lmom) As Double
Dim seglbetas(lmom) As Double
Dim seglgamas(lmom) As Double
Dim seglalf(lmom) As Double
Dim seglbet(lmom) As Double
Dim seglgam(lmom) As Double

Private Sub brezti_tris(ByVal pirma, ByVal antra, ByVal trecia, ByVal chart, ByVal xtit, ByVal
ytit, ByVal ptit, ByVal atit, ByVal ttit, ByVal m, ByVal br_disp)
    With chart
        .Series.Clear()
        .ChartAreas(0).AxisX.Minimum = 0
        .ChartAreas(0).AxisX.Title = xtit
        .ChartAreas(0).AxisY.Title = ytit
        .Series.Add(ptit)
        .Series.Add(atit)
        .Series.Add(ttit)

    If laikoskale = False Then
        For i = 0 To m
            .Series(0).Points.AddXY(i + 1, pirma(i))
            .Series(1).Points.AddXY(i + 1, antra(i))
            .Series(2).Points.AddXY(i + 1, trecia(i))
        Next
        If br_disp = True Then
            .Series.Add(npav)
            .Series(3).YAxisType = DataVisualization.Charting.AxisType.Secondary
            .ChartAreas(0).AxisY2.MajorGrid.Enabled = False
            For i = 0 To nmk
                .Series(3).Points.AddXY(i + 1, newline(i))
            Next
            For i = 0 To 3
                .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
            Next
        Else
            For i = 0 To 2
                .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
            Next
        End If
    Else
        .ChartAreas(0).AxisX.Title = "t"
        For i = 0 To m
            .Series(0).Points.AddXY(minutes(i), pirma(i))
            .Series(1).Points.AddXY(minutes(i), antra(i))
            .Series(2).Points.AddXY(minutes(i), trecia(i))
        Next
        .ChartAreas(0).AxisX.MajorGrid.Interval = 1
        If lmom = 11 Then
            Dim int1 As StripLine = New StripLine()
            int1.BackColor = Color.LightGreen
            int1.IntervalOffset = 0
            int1.StripWidth = 1
            .ChartAreas(0).AxisX.StripLines.Add(int1)

            Dim int2 As StripLine = New StripLine()
            int2.BackColor = Color.IndianRed
            int2.IntervalOffset = 1
            int2.StripWidth = 5
            .ChartAreas(0).AxisX.StripLines.Add(int2)

            Dim int3 As StripLine = New StripLine()
            int3.BackColor = Color.LightGreen
            int3.IntervalOffset = 6
            int3.StripWidth = 5
        End If
    End With
End Sub

```

```

        .ChartAreas(0).AxisX.StripLines.Add(int3)
    End If

    If br_disp = True Then
        .Series.Add(npav)
        .Series(3).YAxisType = DataVisualization.Charting.AxisType.Secondary
        .ChartAreas(0).AxisY2.MajorGrid.Enabled = False
        For i = 0 To nmk
            .Series(3).Points.AddXY(minutes(i), newline(i))
        Next
        For i = 0 To 3
            .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
        Next
    Else
        For i = 0 To 2
            .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
        Next
    End If
End If
End With
End Sub

Private Sub brezti_viena(ByVal pirma, ByVal chart, ByVal xtit, ByVal ytit, ByVal ptit, ByVal m,
ByVal br_disp)
    With chart
        .Series.Clear()
        .ChartAreas(0).AxisX.Title = xtit
        .ChartAreas(0).AxisY.Title = ytit
        .Series.Add(ptit)

        If laikoskale = False Then
            For i = 0 To m
                .Series(0).Points.AddXY(i + 1, pirma(i))
            Next

            If br_disp = True Then
                .Series.Add(npav)
                .Series(1).YAxisType = DataVisualization.Charting.AxisType.Secondary
                For i = 0 To nmk
                    .Series(1).Points.AddXY(i + 1, newline(i))
                Next
                For i = 0 To 1
                    .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
                Next
            Else
                .Series(0).ChartType = DataVisualization.Charting.SeriesChartType.Line
            End If
        Else
            .ChartAreas(0).AxisX.Title = "t"
            For i = 0 To m
                .Series(0).Points.AddXY(minutes(i), pirma(i))
            Next

            If br_disp = True Then
                .Series.Add(npav)
                .Series(1).YAxisType = DataVisualization.Charting.AxisType.Secondary
                For i = 0 To nmk
                    .Series(1).Points.AddXY(minutes(i), newline(i))
                Next
                For i = 0 To 1
                    .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Line
                Next
            Else
                .Series(0).ChartType = DataVisualization.Charting.SeriesChartType.Line
            End If
        End If
    End With
End Sub

```

```

Private Function Slenk_vidurkis(ByVal zirk, ByVal masyvas, ByVal kiek) As Double()
    Dim kiekis As Integer
    Dim z = zirk
    Dim mas = masyvas
    kiekis = kiek - z + 1
    Dim suma As Double

    Dim slvid(kiekis) As Double
    For i = 0 To kiekis
        suma = 0
        For j = 0 To z - 1
            suma = suma + mas(i + j)
        Next
        slvid(i) = suma / z
    Next
    Slenk_vidurkis = slvid
End Function

Private Function dispersija(ByVal masyvas, ByVal kiek, ByVal zirk) As Double()
    Dim kiekis As Integer
    Dim z = zirk
    Dim mas = masyvas
    kiekis = kiek - z + 1
    Dim suma As Double

    'vidurkio skaičiavimas
    Dim vid(kiekis) As Double
    For i = 0 To kiekis
        suma = 0
        For j = 0 To z - 1
            suma = suma + mas(i + j)
        Next
        vid(i) = suma / z
    Next
    'sqrt dispersijos skaičiavimas
    Dim skirt As Double
    Dim disp(kiekis) As Double

    For i = 0 To kiekis
        suma = 0
        For j = 0 To z - 1
            skirt = mas(i + j) - vid(i)
            suma = suma + skirt * skirt
        Next
        disp(i) = Math.Sqrt(suma / (z - 1))
    Next
    dispersija = disp
End Function

Private Function segmentinis(ByVal masyvas, ByVal visi_nariai, ByVal stulp_kiekis) As Double()
    Dim m As Integer = visi_nariai
    Dim mas = masyvas
    Dim sm As Integer = stulp_kiekis
    Dim suma As Double = 0
    Dim vid(sm) As Double
    Dim elem As Integer = 0
    Dim kmin As Integer = 1
    Dim vard As Integer = 0

    For i = 0 To m
        If minutes(i) < kmin Then
            suma = suma + mas(i)
            vard = vard + 1
        Else
            vid(elem) = suma / vard
            elem = elem + 1
            kmin = kmin + 1
        End If
    Next

```

```

        suma = mas(i)
        vard = 1
    End If
Next
vid(elem) = suma / vard
segmentinis = vid
End Function

Private Sub brezti_tris_seg(m(ByVal pirma, ByVal antra, ByVal trecia, ByVal chart, ByVal xtit,
ByVal ytit, ByVal ptit, ByVal atit, ByVal ttit)
    With chart
        .Series.Clear()
        .ChartAreas(0).AxisX.Title = xtit
        .ChartAreas(0).AxisY.Title = ytit
        .Series.Add(ptit)
        .Series.Add(atit)
        .Series.Add(ttit)

        For i = 0 To lmom - 1
            .Series(0).Points.AddXY(i + 0.5, pirma(i))
            .Series(1).Points.AddXY(i + 0.5, antra(i))
            .Series(2).Points.AddXY(i + 0.5, trecia(i))
        Next
        For i = 0 To 2
            .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Column
        Next
        .ChartAreas(0).AxisX.CustomLabels.Clear()
        For i = 0 To lmom - 1
            .ChartAreas(0).AxisX.CustomLabels.Add(i, i + 1, (i + 1).ToString)
        Next
        .ChartAreas(0).RecalculateAxesScale()
    End With
End Sub

Private Sub brezti_viena_seg(m(ByVal pirma, ByVal chart, ByVal xtit, ByVal ytit, ByVal ptit)
    With chart
        .Series.Clear()
        .ChartAreas(0).AxisX.Title = xtit
        .ChartAreas(0).AxisY.Title = ytit
        .Series.Add(ptit)

        For i = 0 To lmom - 1
            .Series(0).Points.AddXY(i + 0.5, pirma(i))
        Next

        .Series(0).ChartType = DataVisualization.Charting.SeriesChartType.Column
        .ChartAreas(0).AxisX.CustomLabels.Clear()
        For i = 0 To lmom - 1
            .ChartAreas(0).AxisX.CustomLabels.Add(i, i + 1, (i + 1).ToString)
        Next
        .ChartAreas(0).RecalculateAxesScale()
    End With
End Sub

Private Sub prad_duom_braiz()
    Dim pavy As String = Chart1.ChartAreas(0).AxisY.Title
    Dim pavx As String = Chart1.ChartAreas(0).AxisX.Title
    Dim pav1 As String = Chart1.Series(0).Name
    Dim pav2 As String = Chart1.Series(1).Name
    Dim pav3 As String = Chart1.Series(2).Name

    If Check_segmentinis.Checked = True Then
        brezti_tris_seg(segxx, segyy, segzz, Chart1, "t", pavy, pav1, pav2, pav3)
    ElseIf arslenkantys = False Then
        brezti_tris(eilutes.pxx, eilutes.pyy, eilutes.pzz, Chart1, pavx, pavy, pav1, pav2, pav3,
mk + 2, Check_prad.Checked)
    Else

```

```

        brezti_tris(slxx, slyy, slzz, Chart1, pavx, pavy, pav1, pav2, pav3, slmk + 2,
Check_prad.Checked)
    End If
End Sub

Private Sub tikriniu_braiz()
    If Check_segmentinis.Checked = True Then
        brezti_tris_seg1(seg11, seg12, seg13, Chart2, "t", "Tikrinės reikšmės", "11", "12",
"13")
    ElseIf arslenkantys = False Then
        brezti_tris(eilutes.l1, eilutes.l2, eilutes.l3, Chart2, "k", "Tikrinės reikšmės", "11",
"12", "13", mk, Check_tikrin.Checked)
    Else
        brezti_tris(s111, s112, s113, Chart2, "k", "Tikrinės reikšmės", "11", "12", "13", slmk,
Check_tikrin.Checked)
    End If
End Sub

Private Sub Btn_gr_abgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_gr_abgz.Click
    If Check_segmentinis.Checked = True Then
        brezti_tris_seg1(segalfas, segbetas, seggamas, Chart4, "t", "α*, β*, γ*", "α*", "β*",
"γ*")
    ElseIf arslenkantys = False Then
        brezti_tris(eilutes.avalfas, eilutes.avbetas, eilutes.avgamas, Chart4, "k", "α*, β*,
γ*", "α*", "β*", "γ*", mk, Check_abgk.Checked)
    Else
        brezti_tris(slalfas, slbetas, slgamas, Chart4, "k", "α*, β*, γ*", "α*", "β*", "γ*",
slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_labgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_gr_labgz.Click
    If Check_segmentinis.Checked = True Then
        brezti_tris_seg1(seg1alfas, seg1betas, seg1gamas, Chart4, "t", "α*, β*, γ*",
"log(α*+1)", "log(β*+1)", "log(γ*+1)")
    ElseIf arslenkantys = False Then
        brezti_tris(eilutes.lavalfas, eilutes.lavbetas, eilutes.lavgamas, Chart4, "k", "α*, β*,
γ*", "log(α*+1)", "log(β*+1)", "log(γ*+1)", mk, Check_abgk.Checked)
    Else
        brezti_tris(s11alfas, s11betas, s11gamas, Chart4, "k", "α*, β*, γ*", "log(α*+1)",
"log(β*+1)", "log(γ*+1)", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_az_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_az.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg1(segalfas, Chart4, "t", "α*", "α*")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.avalfas, Chart4, "k", "α*", "α*", mk, Check_abgk.Checked)
    Else
        brezti_viena(slalfas, Chart4, "k", "α*", "α*", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_bz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_bz.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg1(segbetas, Chart4, "t", "β*", "β*")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.avbetas, Chart4, "k", "β*", "β*", mk, Check_abgk.Checked)
    Else
        brezti_viena(slbetas, Chart4, "k", "β*", "β*", slmk, Check_abgk.Checked)
    End If
End Sub

```

```

Private Sub Btn_gr_gz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_gz.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_segm(seggamas, Chart4, "t", " $\gamma^*$ ", " $\gamma^*$ ")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.avgamas, Chart4, "k", " $\gamma^*$ ", " $\gamma^*$ ", mk, Check_abgk.Checked)
    Else
        brezti_viena(slgamas, Chart4, "k", " $\gamma^*$ ", " $\gamma^*$ ", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_laz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_laz.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_segm(seglalfas, Chart4, "t", " $\log(\alpha^*+1)$ ", " $\log(\alpha^*+1)$ ")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.lavalfas, Chart4, "k", " $\log(\alpha^*+1)$ ", " $\log(\alpha^*+1)$ ", mk,
Check_abgk.Checked)
    Else
        brezti_viena(sllalfas, Chart4, "k", " $\log(\alpha^*+1)$ ", " $\log(\alpha^*+1)$ ", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_lbz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_lbz.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_segm(seglbetas, Chart4, "t", " $\log(\beta^*+1)$ ", " $\log(\beta^*+1)$ ")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.lavbetas, Chart4, "k", " $\log(\beta^*+1)$ ", " $\log(\beta^*+1)$ ", mk,
Check_abgk.Checked)
    Else
        brezti_viena(sllbetas, Chart4, "k", " $\log(\beta^*+1)$ ", " $\log(\beta^*+1)$ ", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_lgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_lgz.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_segm(seglgamas, Chart4, "t", " $\log(\gamma^*+1)$ ", " $\log(\gamma^*+1)$ ")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.lavgamas, Chart4, "k", " $\log(\gamma^*+1)$ ", " $\log(\gamma^*+1)$ ", mk,
Check_abgk.Checked)
    Else
        brezti_viena(sllgamas, Chart4, "k", " $\log(\gamma^*+1)$ ", " $\log(\gamma^*+1)$ ", slmk, Check_abgk.Checked)
    End If
End Sub

Private Sub Btn_gr_abg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_abg.Click
    If Check_segmentinis.Checked = True Then
        brezti_tris_segm(segalf, segbet, seggam, Chart3, "t", " $\alpha$ ,  $\beta$ ,  $\gamma$ ", " $\alpha$ ", " $\beta$ ", " $\gamma$ ")
    ElseIf arslenkantys = False Then
        brezti_tris(eilutes.avalf, eilutes.avbet, eilutes.avgam, Chart3, "k", " $\alpha$ ,  $\beta$ ,  $\gamma$ ", " $\alpha$ ",
" $\beta$ ", " $\gamma$ ", mk, Check_abg.Checked)
    Else
        brezti_tris(slalf, slbet, slgam, Chart3, "k", " $\alpha$ ,  $\beta$ ,  $\gamma$ ", " $\alpha$ ", " $\beta$ ", " $\gamma$ ", slmk,
Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_labg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_gr_labg.Click
    If Check_segmentinis.Checked = True Then
        brezti_tris_segm(segalf, seglbet, seglgam, Chart3, "t", " $\alpha$ ,  $\beta$ ,  $\gamma$ ", " $\log(\alpha+1)$ ",
" $\log(\beta+1)$ ", " $\log(\gamma+1)$ ")
    ElseIf arslenkantys = False Then

```



```

        brezti_tris(eilutes.lavalf, eilutes.lavbet, eilutes.lavgam, Chart3, "k", "α, β, γ",
"log(α+1)", "log(β+1)", "log(γ+1)", mk, Check_abg.Checked)
    Else
        brezti_tris(sllalf, sllbet, sllgam, Chart3, "k", "α, β, γ", "log(α+1)", "log(β+1)",
"log(γ+1)", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_a_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_a.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(segalf, Chart3, "t", "α", "α")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.aval, Chart3, "k", "α", "α", mk, Check_abg.Checked)
    Else
        brezti_viena(slalf, Chart3, "k", "α", "α", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_b_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_b.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(segbet, Chart3, "t", "β", "β")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.avbet, Chart3, "k", "β", "β", mk, Check_abg.Checked)
    Else
        brezti_viena(slbet, Chart3, "k", "β", "β", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_g_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_g.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(seggam, Chart3, "t", "γ", "γ")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.avgam, Chart3, "k", "γ", "γ", mk, Check_abg.Checked)
    Else
        brezti_viena(slgam, Chart3, "k", "γ", "γ", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_la_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_la.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(seglalf, Chart3, "t", "log(α+1)", "log(α+1)")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.lavalf, Chart3, "k", "log(α+1)", "log(α+1)", mk, Check_abg.Checked)
    Else
        brezti_viena(sllalf, Chart3, "k", "log(α+1)", "log(α+1)", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_lb_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_lb.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(seglbet, Chart3, "t", "log(β+1)", "log(β+1)")
    ElseIf arslenkantys = False Then
        brezti_viena(eilutes.lavbet, Chart3, "k", "log(β+1)", "log(β+1)", mk, Check_abg.Checked)
    Else
        brezti_viena(sllbet, Chart3, "k", "log(β+1)", "log(β+1)", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_gr_lg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_gr_lg.Click
    If Check_segmentinis.Checked = True Then
        brezti_viena_seg(seglgam, Chart3, "t", "log(γ+1)", "log(γ+1)")
    ElseIf arslenkantys = False Then

```

```

        brezti_viena(eilutes.lavgam, Chart3, "k", "log( $\gamma$ +1)", "log( $\gamma$ +1)", mk, Check_abg.Checked)
    Else
        brezti_viena(sllgam, Chart3, "k", "log( $\gamma$ +1)", "log( $\gamma$ +1)", slmk, Check_abg.Checked)
    End If
End Sub

Private Sub Btn_trint_papild_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_trint_papild.Click
    arpapildoma = False
    Btn_prad_graf_Click(0, EventArgs.Empty)
    Papildomi_taskai.Clear()
    Panel_papild.Visible = False
    Check_abg.Checked = False
    Check_abgk.Checked = False
    Check_prad.Checked = False
    Check_tikrin.Checked = False
End Sub

Private Sub Btn_br_papild_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_br_papild.Click
    npav = Text_npav.Text
    'Nuskaitome duomenis į masyvą
    Dim visas_tekstas As String = Papildomi_taskai.Text
    visas_tekstas = Replace(visas_tekstas, ",", ".")

    If visas_tekstas = Nothing Then
        Return
    End If

    Dim nariai() As String = visas_tekstas.Split(vbCrLf)

    Dim k As Integer = 0
    Dim pmk = UBound(nariai)
    For i = 0 To pmk
        If i > mk + 1 Then
            Exit For
        End If
        If nariai(i) = Nothing Then
            Continue For
        ElseIf IsNumeric(nariai(i)) = False Then
            Continue For
        Else
            newline(k) = CDb1(Val(nariai(i)))
        End If
        k = k + 1
    Next
    nmk = k - 1

    If Check_prad.Checked = True Then
        prad_duom_braiz()
    End If

    If Check_tikrin.Checked = True Then
        tikriniu_braiz()
    End If

    If Check_abgk.Checked = True Then
        Btn_gr_abgz_Click(0, EventArgs.Empty)
    End If

    If Check_abg.Checked = True Then
        Btn_gr_abg_Click(0, EventArgs.Empty)
    End If

    arpapildoma = True
    Panel_papild.Visible = False
End Sub

```

```

Private Sub Btn_papildpan_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_papildpan.Click
    Panel_papild.Visible = True
End Sub

Private Sub Btn_pk_cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_pk_cancel.Click
    Panel_papild.Visible = False
End Sub

Private Sub Btn_prad_graf_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_prad_graf.Click
    Check_prad.Checked = False
    Check_tikrin.Checked = False
    Check_abgk.Checked = False
    Check_abg.Checked = False
    arslenkantys = False
    Check_segmentinis.Checked = False

    Dim pavy As String = Chart1.ChartAreas(0).AxisY.Title
    Dim pavx As String = Chart1.ChartAreas(0).AxisX.Title
    Dim pav1 As String = Chart1.Series(0).Name
    Dim pav2 As String = Chart1.Series(1).Name
    Dim pav3 As String = Chart1.Series(2).Name

    brezti_tris(eilutes.pxx, eilutes.pyy, eilutes.pzz, Chart1, pavx, pavy, pav1, pav2, pav3, mk
+ 2, False)
    brezti_tris(eilutes.l11, eilutes.l12, eilutes.l13, Chart2, "k", "Tikrinės reikšmės", "l1",
"l2", "l3", mk, False)
    brezti_tris(eilutes.avalfas, eilutes.avbetas, eilutes.avgamas, Chart4, "k", "α*, β*, γ*",
"α*", "β*", "γ*", mk, False)
    brezti_tris(eilutes.avalf, eilutes.avbet, eilutes.avgam, Chart3, "k", "α, β, γ", "α", "β",
"γ", mk, False)
End Sub

Private Sub Btn_slenkantys_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_slenkantys.Click
    Panel_slenkantys.Visible = True
End Sub

Private Sub Btn_br_slenk_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_br_slenk.Click
    Dim Value As Integer
    If Integer.TryParse(zirkles_box.Text, Value) Then
        If Value > 0 AndAlso Value < mk Then
            Dim zirkles = CInt(Int(zirkles_box.Text))
            arslenkantys = True
            slmk = mk - zirkles + 1

            'Suvidurkinti pradiniai duomenys
            slxx = Slenk_vidurkis(zirkles, eilutes.pxx, mk + 2)
            slyy = Slenk_vidurkis(zirkles, eilutes.pyy, mk + 2)
            slzz = Slenk_vidurkis(zirkles, eilutes.pzz, mk + 2)
            Dim pavy As String = Chart1.ChartAreas(0).AxisY.Title
            Dim pavx As String = Chart1.ChartAreas(0).AxisX.Title
            Dim pav1 As String = Chart1.Series(0).Name
            Dim pav2 As String = Chart1.Series(1).Name
            Dim pav3 As String = Chart1.Series(2).Name
            brezti_tris(slxx, slyy, slzz, Chart1, pavx, pavy, pav1, pav2, pav3, slmk + 2,
Check_prad.Checked)

            'Suvidurkintos tikrinės reikšmės
            sl11 = Slenk_vidurkis(zirkles, eilutes.l11, mk)
            sl12 = Slenk_vidurkis(zirkles, eilutes.l12, mk)
            sl13 = Slenk_vidurkis(zirkles, eilutes.l13, mk)
            brezti_tris(sl11, sl12, sl13, Chart2, "k", "Tikrinės reikšmės", "l1", "l2", "l3",
slmk, Check_tikrin.Checked)

            'Suvidurkinti parametrai
            slalfas = Slenk_vidurkis(zirkles, eilutes.avalfas, mk)

```

```

slbetas = Slenk_vidurkis(zirkles, eilutes.avbetas, mk)
slgamas = Slenk_vidurkis(zirkles, eilutes.avgamas, mk)
slalf = Slenk_vidurkis(zirkles, eilutes.avalf, mk)
slbet = Slenk_vidurkis(zirkles, eilutes.avbet, mk)
slgam = Slenk_vidurkis(zirkles, eilutes.avgam, mk)
brezti_tris(slalfas, slbetas, slgamas, Chart4, "k", "α*, β*, γ*", "α*", "β*", "γ*",
                                                    slmk, Check_abgk.Checked)

brezti_tris(slalf, slbet, slgam, Chart3, "k", "α, β, γ", "α", "β", "γ", slmk,
                                                    Check_abg.Checked)

'Suvidurkinti logaritmuoti parametrai
sllalfas = Slenk_vidurkis(zirkles, eilutes.lavalfas, mk)
sllbetas = Slenk_vidurkis(zirkles, eilutes.lavbetas, mk)
sllgamas = Slenk_vidurkis(zirkles, eilutes.lavgamas, mk)
sllalf = Slenk_vidurkis(zirkles, eilutes.lavalf, mk)
sllbet = Slenk_vidurkis(zirkles, eilutes.lavbet, mk)
sllgam = Slenk_vidurkis(zirkles, eilutes.lavgam, mk)

'Suvidurkinta dispersija
Check_segmentinis.Checked = False
Panel_slenkantys.Visible = False
Else
    MessageBox.Show("Įveskite tinkamą intervalą")
End If
Else
    MessageBox.Show("Įveskite tinkamą intervalą")
End If
End Sub

Private Sub Btn_slen_cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_slen_cancel.Click
    Panel_slenkantys.Visible = False
End Sub

Private Sub Btn_drr_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_drr.Click
    Dim Value As Integer
    If Integer.TryParse(varint_box.Text, Value) Then
        If Value > 0 AndAlso Value < mk + 2 Then
            Papildomi_taskai.Clear()
            Dim varzirk = CInt(Int(varint_box.Text))
            arpapildoma = True
            nmk = mk - varzirk + 3
            newline = dispersija(eilutes.avalfas, mk + 2, varzirk)
            For i = 0 To nmk
                Papildomi_taskai.Text = Papildomi_taskai.Text & newline(i) & vbCrLf
            Next
            Text_npav.Text = "σ(RR)"
        Else
            MessageBox.Show("Įveskite tinkamą intervalą")
        End If
    Else
        MessageBox.Show("Įveskite tinkamą intervalą")
    End If
End Sub

Private Sub Btn_ddqrs_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Btn_ddqrs.Click
    Dim Value As Integer
    If Integer.TryParse(varint_box.Text, Value) Then
        If Value > 0 AndAlso Value < mk + 2 Then
            Papildomi_taskai.Clear()
            Dim varzirk = CInt(Int(varint_box.Text))
            arpapildoma = True
            nmk = mk - varzirk + 3
            newline = dispersija(eilutes.pyy, mk + 2, varzirk)
            For i = 0 To nmk
                Papildomi_taskai.Text = Papildomi_taskai.Text & newline(i) & vbCrLf
            Next
        End If
    End If
End Sub

```

```

        Text_npav.Text = "σ(DQRS)"
    Else
        MessageBox.Show("Įveskite tinkamą intervalą")
    End If
Else
    MessageBox.Show("Įveskite tinkamą intervalą")
End If
End Sub

Private Sub Btn_djt_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_djt.Click
    Dim Value As Integer
    If Integer.TryParse(varint_box.Text, Value) Then
        If Value > 0 AndAlso Value < mk + 2 Then
            Papildomi_taskai.Clear()
            Dim varzirk = CInt(Int(varint_box.Text))
            arpapildoma = True
            nmk = mk - varzirk + 3
            newline = dispersija(eilutes.pzz, mk + 2, varzirk)
            For i = 0 To nmk
                Papildomi_taskai.Text = Papildomi_taskai.Text & newline(i) & vbCrLf
            Next
            Text_npav.Text = "σ(JT)"
        Else
            MessageBox.Show("Įveskite tinkamą intervalą")
        End If
    Else
        MessageBox.Show("Įveskite tinkamą intervalą")
    End If
End Sub

Private Sub Btn_laikai_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_laikai.Click
    Panel_laikai.Visible = True
End Sub

Private Sub Btn_laik_cancel_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_laik_cancel.Click
    Check_segmentinis.Checked = False
    Panel_laikai.Visible = False
End Sub

Private Sub Btn_br_laikas_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_br_laikas.Click
    laikoskale = True

    'Nuskaitome duomenis į masyvą
    Dim visas_tekstas As String = laikas_box.Text
    visas_tekstas = Replace(visas_tekstas, ",", ".")

    If visas_tekstas = Nothing Then
        Return
    End If

    Dim nariai() As String = visas_tekstas.Split(vbCrLf)

    Dim k As Integer = 0
    Dim pmk = UBound(nariai)
    For i = 0 To pmk
        If nariai(i) = Nothing Then
            Continue For
        ElseIf IsNumeric(nariai(i)) = False Then
            Continue For
        Else
            minutes(k) = CDb1(Val(nariai(i)))
        End If
        k = k + 1
    Next

```

```

Dim minkiek = k - 1

If minkiek <> mk + 2 Then
    MessageBox.Show("Įveskite tinkamą kiekį laiko momentų")
    Return
End If

lmom = CInt(Val(minutes(mk + 2)))

If Check_segmentinis.Checked = True Then
    'Suvidurkinti pradiniai duomenys
    segxx = segmentinis(eilutes.pxx, mk + 2, lmom)
    segyy = segmentinis(eilutes.pyy, mk + 2, lmom)
    segzz = segmentinis(eilutes.pzz, mk + 2, lmom)
    Dim pavy As String = Chart1.ChartAreas(0).AxisY.Title
    Dim pavx As String = "t"
    Dim pav1 As String = Chart1.Series(0).Name
    Dim pav2 As String = Chart1.Series(1).Name
    Dim pav3 As String = Chart1.Series(2).Name
    brezti_tris_segm(segxx, segyy, segzz, Chart1, pavx, pavy, pav1, pav2, pav3)

    'Suvidurkintos tikrinės reikšmės
    segl1 = segmentinis(eilutes.l11, mk, lmom)
    segl2 = segmentinis(eilutes.l12, mk, lmom)
    segl3 = segmentinis(eilutes.l13, mk, lmom)
    brezti_tris_segm(segl1, segl2, segl3, Chart2, "t", "Tikrinės reikšmės", "l1", "l2", "l3")

    'Suvidurkinti parametrai
    segalfas = segmentinis(eilutes.avalfas, mk, lmom)
    segbetas = segmentinis(eilutes.avbetas, mk, lmom)
    seggamas = segmentinis(eilutes.avgamas, mk, lmom)
    segalf = segmentinis(eilutes.avalf, mk, lmom)
    segbet = segmentinis(eilutes.avbet, mk, lmom)
    seggam = segmentinis(eilutes.avgam, mk, lmom)
    brezti_tris_segm(segalfas, segbetas, seggamas, Chart4, "t", "α*, β*, γ*", "α*", "β*", "γ*")
    brezti_tris_segm(segalf, segbet, seggam, Chart3, "t", "α, β, γ", "α", "β", "γ")

    'Suvidurkinti logaritmuoti parametrai
    seglalfas = segmentinis(eilutes.lavalfas, mk, lmom)
    seglbetas = segmentinis(eilutes.lavbetas, mk, lmom)
    seglgamas = segmentinis(eilutes.lavgamas, mk, lmom)
    seglalf = segmentinis(eilutes.lavalf, mk, lmom)
    seglbet = segmentinis(eilutes.lavbet, mk, lmom)
    seglgam = segmentinis(eilutes.lavgam, mk, lmom)
Else
    Dim pavy As String = Chart1.ChartAreas(0).AxisY.Title
    Dim pavx As String = Chart1.ChartAreas(0).AxisX.Title
    Dim pav1 As String = Chart1.Series(0).Name
    Dim pav2 As String = Chart1.Series(1).Name
    Dim pav3 As String = Chart1.Series(2).Name
    brezti_tris(eilutes.pxx, eilutes.pyy, eilutes.pzz, Chart1, pavx, pavy, pav1, pav2, pav3,
mk + 2, False)
    brezti_tris(eilutes.l11, eilutes.l12, eilutes.l13, Chart2, "k", "Tikrinės reikšmės", "l1",
"l2", "l3", mk, False)
    brezti_tris(eilutes.avalfas, eilutes.avbetas, eilutes.avgamas, Chart4, "k", "α*, β*,
γ*", "α*", "β*", "γ*", mk, False)
    brezti_tris(eilutes.avalf, eilutes.avbet, eilutes.avgam, Chart3, "k", "α, β, γ", "α",
"β", "γ", mk, False)
End If
Panel_laikai.Visible = False
End Sub

Private Sub grafikai_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
 MyBase.Load
    brezti_tris(eilutes.l11, eilutes.l12, eilutes.l13, Chart2, "k", "Tikrinės reikšmės", "l1",
"l2", "l3", mk, False)
    brezti_tris(eilutes.avalfas, eilutes.avbetas, eilutes.avgamas, Chart4, "k", "α*, β*, γ*",
"α*", "β*", "γ*", mk, False)

```

```

        brezti_tris(eilutes.avalf, eilutes.avbet, eilutes.avgam, Chart3, "k", "α, β, γ", "α", "β",
"γ", mk, False)
    End Sub
End Class

```

```
Public Class dispersija
```

```
    Inherits System.Windows.Forms.Form
```

```
    Public mycaller As eilutes
```

```
    Dim mins = eilutes.mink
```

```
    Dim nsk = eilutes.dnr - 1
```

```
    Dim kelint As Integer
```

```
    Dim bendra As Boolean = False
```

```
    Dim avvar(mins, 11) As Double
```

```

    Private Sub brezti_tris_var(ByVal kelintas_asmuo, ByVal pirm_koef, ByVal antr_koef, ByVal
trec_koef, ByVal chart, ByVal xtit, ByVal ytit, ByVal ptit, ByVal atit, ByVal ttit)

```

```
    Dim pirma(mins) As Double
```

```
    Dim antra(mins) As Double
```

```
    Dim trecia(mins) As Double
```

```
    Dim ka = kelintas_asmuo
```

```
    If bendra = False Then
```

```
        For i = 0 To mins
```

```
            pirma(i) = eilutes.disp(ka, i, pirm_koef)
```

```
            antra(i) = eilutes.disp(ka, i, antr_koef)
```

```
            trecia(i) = eilutes.disp(ka, i, trec_koef)
```

```
        Next
```

```
    Else
```

```
        For i = 0 To mins
```

```
            pirma(i) = avvar(i, pirm_koef)
```

```
            antra(i) = avvar(i, antr_koef)
```

```
            trecia(i) = avvar(i, trec_koef)
```

```
        Next
```

```
    End If
```

```
    With chart
```

```
        .Series.Clear()
```

```
        .ChartAreas(0).AxisX.Title = xtit
```

```
        .ChartAreas(0).AxisY.Title = ytit
```

```
        .Series.Add(ptit)
```

```
        .Series.Add(atit)
```

```
        .Series.Add(ttit)
```

```
        .series(0).color = Color.White
```

```
        .series(1).color = Color.White
```

```
        .series(2).color = Color.White
```

```
        .ChartAreas(0).AxisX.Minimum = 0
```

```
        .ChartAreas(0).AxisX.Maximum = mins + 1
```

```
    For i = 0 To mins
```

```
        .Series(0).Points.AddXY(i + 0.5, pirma(i))
```

```
        .Series(1).Points.AddXY(i + 0.5, antra(i))
```

```
        .Series(2).Points.AddXY(i + 0.5, trecia(i))
```

```
        If i = 0 Or i > 5 Then
```

```
            .Series(0).Points(i).Color = Color.Lime
```

```
            .Series(1).Points(i).Color = Color.LawnGreen
```

```
            .Series(2).Points(i).Color = Color.LimeGreen
```

```
        Else
```

```
            .Series(0).Points(i).Color = Color.Firebrick
```

```
            .Series(1).Points(i).Color = Color.Crimson
```

```
            .Series(2).Points(i).Color = Color.DarkRed
```

```
        End If
```

```
    Next
```

```
    For i = 0 To 2
```

```
        .Series(i).ChartType = DataVisualization.Charting.SeriesChartType.Column
```

```
    Next
```

```
    .ChartAreas(0).AxisX.CustomLabels.Clear()
```

```

        For i = 0 To mins + 1
            .ChartAreas(0).AxisX.CustomLabels.Add(i - 0.5, i + 0.5, i.ToString)
        Next
        .ChartAreas(0).RecalculateAxesScale()
    End With
End Sub

Private Sub brezti_viena_var(ByVal kelintas_asmuo, ByVal pirm_koef, ByVal chart, ByVal xtit,
ByVal ytit, ByVal ptit)
    Dim pirma(mins) As Double
    Dim ka = kelintas_asmuo

    If bendra = False Then
        For i = 0 To mins
            pirma(i) = eilutes.disp(ka, i, pirm_koef)
        Next
    Else
        For i = 0 To mins
            pirma(i) = avvar(i, pirm_koef)
        Next
    End If

    With chart
        .Series.Clear()
        .ChartAreas(0).AxisX.Title = xtit
        .ChartAreas(0).AxisY.Title = ytit
        .Series.Add(ptit)
        .series(0).color = Color.White

        For i = 0 To mins
            .Series(0).Points.AddXY(i + 0.5, pirma(i))
            If i = 0 Or i > 5 Then
                .Series(0).Points(i).Color = Color.Lime
            Else
                .Series(0).Points(i).Color = Color.Crimson
            End If
        Next

        .Series(0).ChartType = DataVisualization.Charting.SeriesChartType.Column

        .ChartAreas(0).AxisX.CustomLabels.Clear()
        For i = 0 To mins + 1
            .ChartAreas(0).AxisX.CustomLabels.Add(i - 0.5, i + 0.5, i.ToString)
        Next
        .ChartAreas(0).RecalculateAxesScale()
    End With
End Sub

Private Sub dispersija_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
    brezti_tris_var(nsk,0,1,2,Var_Chart1,"t", " $\sigma(\alpha^*)$ ,  $\sigma(\beta^*)$ ,  $\sigma(\gamma^*)$ ", "1. $\sigma(\alpha^*)$ ", "2. $\sigma(\beta^*)$ ", "3. $\sigma(\gamma^*)$ ")
    brezti_tris_var(nsk, 3, 4, 5, Var_Chart2,"t", " $\sigma(\alpha)$ ,  $\sigma(\beta)$ ,  $\sigma(\gamma)$ ", "1. $\sigma(\alpha)$ ", "2. $\sigma(\beta)$ ", "3. $\sigma(\gamma)$ ")

    'Spausdinimas į lentelę
    Dim eil As Integer = 0
    For k = 0 To nsk
        dispersiju_lentele.Rows.Add()
        dispersiju_lentele.Rows(eil).Cells(0).Value = eilutes.dname(k)
        For i = 0 To mins
            dispersiju_lentele.Rows.Add()
            For j = 0 To 11
                dispersiju_lentele.Rows(eil + i).Cells(j + 1).Value = eilutes.disp(k, i, j)
            Next
        Next
        eil = eil + mins + 1
    Next
Next

```



```

For i = 0 To nsk
    tiriamuju_pasirinkimas.Items.Add(eilutes.dname(i))
Next
kelint = nsk
tiriamuju_pasirinkimas.SelectedIndex = kelint
End Sub

Private Sub Btn_var_abgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_abgz.Click
    brezti_tris_var(kelint, 0, 1, 2, Var_Chart1, "t", " $\sigma(\alpha^*)$ ", " $\sigma(\beta^*)$ ", " $\sigma(\gamma^*)$ ", "1.  $\sigma(\alpha^*)$ ", "2.  $\sigma(\beta^*)$ ", "3.  $\sigma(\gamma^*)$ ")
End Sub

Private Sub Btn_var_labgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_labgz.Click
    brezti_tris_var(kelint, 6, 7, 8, Var_Chart1, "t", " $\sigma(\alpha^*)$ ", " $\sigma(\beta^*)$ ", " $\sigma(\gamma^*)$ ", "1.  $\sigma(\log(\alpha^*+1))$ ", "2.  $\sigma(\log(\beta^*+1))$ ", "3.  $\sigma(\log(\gamma^*+1))$ ")
End Sub

Private Sub Btn_var_abg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_abg.Click
    brezti_tris_var(kelint, 3, 4, 5, Var_Chart2, "t", " $\sigma(\alpha)$ ", " $\sigma(\beta)$ ", " $\sigma(\gamma)$ ", "1.  $\sigma(\alpha)$ ", "2.  $\sigma(\beta)$ ", "3.  $\sigma(\gamma)$ ")
End Sub

Private Sub Btn_var_labg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_labg.Click
    brezti_tris_var(kelint, 9, 10, 11, Var_Chart2, "t", " $\sigma(\alpha)$ ", " $\sigma(\beta)$ ", " $\sigma(\gamma)$ ", "1.  $\sigma(\log(\alpha+1))$ ", "2.  $\sigma(\log(\beta+1))$ ", "3.  $\sigma(\log(\gamma+1))$ ")
End Sub

Private Sub Btn_var_az_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_var_az.Click
    brezti_viena_var(kelint, 0, Var_Chart1, "t", " $\sigma(\alpha^*)$ ", " $\sigma(\alpha^*)$ ")
End Sub

Private Sub Btn_var_bz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_var_bz.Click
    brezti_viena_var(kelint, 1, Var_Chart1, "t", " $\sigma(\beta^*)$ ", " $\sigma(\beta^*)$ ")
End Sub

Private Sub Btn_var_gz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_var_gz.Click
    brezti_viena_var(kelint, 2, Var_Chart1, "t", " $\sigma(\gamma^*)$ ", " $\sigma(\gamma^*)$ ")
End Sub

Private Sub Btn_var_laz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_laz.Click
    brezti_viena_var(kelint, 6, Var_Chart1, "t", " $\sigma(\log(\alpha^*+1))$ ", " $\sigma(\log(\alpha^*+1))$ ")
End Sub

Private Sub Btn_var_lbz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_lbz.Click
    brezti_viena_var(kelint, 7, Var_Chart1, "t", " $\sigma(\log(\beta^*+1))$ ", " $\sigma(\log(\beta^*+1))$ ")
End Sub

Private Sub Btn_var_lgz_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Btn_var_lgz.Click
    brezti_viena_var(kelint, 8, Var_Chart1, "t", " $\sigma(\log(\gamma^*+1))$ ", " $\sigma(\log(\gamma^*+1))$ ")
End Sub

Private Sub Btn_var_a_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_var_a.Click
    brezti_viena_var(kelint, 3, Var_Chart2, "t", " $\sigma(\alpha)$ ", " $\sigma(\alpha)$ ")
End Sub

Private Sub Btn_var_b_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
Btn_var_b.Click

```

```

        brezti_viena_var(kelint, 4, Var_Chart2, "t", " $\sigma(\beta)$ ", " $\sigma(\beta)$ ")
    End Sub

    Private Sub Btn_var_g_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Btn_var_g.Click
        brezti_viena_var(kelint, 5, Var_Chart2, "t", " $\sigma(\gamma)$ ", " $\sigma(\gamma)$ ")
    End Sub

    Private Sub Btn_var_la_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Btn_var_la.Click
        brezti_viena_var(kelint, 9, Var_Chart2, "t", " $\sigma(\log(\alpha+1))$ ", " $\sigma(\log(\alpha+1))$ ")
    End Sub

    Private Sub Btn_var_lb_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Btn_var_lb.Click
        brezti_viena_var(kelint, 10, Var_Chart2, "t", " $\sigma(\log(\beta+1))$ ", " $\sigma(\log(\beta+1))$ ")
    End Sub

    Private Sub Btn_var_lg_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
    Btn_var_lg.Click
        brezti_viena_var(kelint, 11, Var_Chart2, "t", " $\sigma(\log(\gamma+1))$ ", " $\sigma(\log(\gamma+1))$ ")
    End Sub

    Private Sub tiriamuju_pasirinkimas_SelectedIndexChanged(ByVal sender As System.Object, ByVal e
    As System.EventArgs) Handles tiriamuju_pasirinkimas.SelectedIndexChanged
        bendra = False
        kelint = tiriamuju_pasirinkimas.SelectedIndex
        brezti_tris_var(kelint, 0, 1, 2, Var_Chart1, "t", " $\sigma(\alpha^*)$ ,  $\sigma(\beta^*)$ ,  $\sigma(\gamma^*)$ ", " $\sigma(\alpha^*)$ ", " $\sigma(\beta^*)$ ", " $\sigma(\gamma^*)$ ")
        brezti_tris_var(kelint, 3, 4, 5, Var_Chart2, "t", " $\sigma(\alpha)$ ,  $\sigma(\beta)$ ,  $\sigma(\gamma)$ ", " $\sigma(\alpha)$ ", " $\sigma(\beta)$ ", " $\sigma(\gamma)$ ")
    End Sub

    Private Sub Btn_bendr_var_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
    Handles Btn_bendr_var.Click
        bendra = True
        Dim suma As Double
        For i = 0 To mins
            For j = 0 To 11
                suma = 0
                For k = 0 To nsk
                    suma = suma + eilutes.disp(k, i, j)
                Next
                avvar(i, j) = suma / (nsk + 1)
            Next
        Next

        'Spausdinimas į lentelę
        Dim eil = eilutes.dnr * mins + eilutes.dnr
        dispersiju_lentele.Rows.Add()
        dispersiju_lentele.Rows(eil).Cells(0).Value = "Viso"
        For i = 0 To mins
            dispersiju_lentele.Rows.Add()
            For j = 0 To 11
                dispersiju_lentele.Rows(eil + i).Cells(j + 1).Value = avvar(i, j)
            Next
        Next

        brezti_tris_var(kelint, 0, 1, 2, Var_Chart1, "t", " $\sigma(\alpha^*)$ ,  $\sigma(\beta^*)$ ,  $\sigma(\gamma^*)$ ", "1. $\sigma(\alpha^*)$ ", "2. $\sigma(\beta^*)$ ", "3. $\sigma(\gamma^*)$ ")
        brezti_tris_var(kelint, 3, 4, 5, Var_Chart2, "t", " $\sigma(\alpha)$ ,  $\sigma(\beta)$ ,  $\sigma(\gamma)$ ", "1. $\sigma(\alpha)$ ", "2. $\sigma(\beta)$ ", "3. $\sigma(\gamma)$ ")
    End Sub
End Class

```