



**KAUNO TECHNOLOGIJOS UNIVERSITETAS  
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS  
TAIKOMOSIOS MATEMATIKOS KATEDRA**

**Dovilė Ašeriškytė**

**FRAKTALINIŲ DIMENSIJŲ SKAIČIAVIMAS  
KAI KURIOMS ŽMOGAUS ORGANIZMO  
FIZIOLOGINIŲ PROCESŲ REALIZACIJOMS**

Magistro darbas

**Vadovai**  
**prof. dr. Z. Navickas**  
**prof. habil. dr. A. Vainoras**

**KAUNAS, 2005**



**KAUNO TECHNOLOGIJOS UNIVERSITETAS**  
**FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS**  
**TAIKOMOSIOS MATEMATIKOS KATEDRA**

**TVIRTINU**  
**Katedros vedėjas**  
**prof. dr. J.Rimas**  
**2005**

**FRAKTALINIŲ DIMENSIJŲ SKAIČIAVIMAS KAI**  
**KURIOMS ŽMOGAUS ORGANIZMO FIZIOLOGINIŲ**  
**PROCESŲ REALIZACIJOMS**

Taikomosios matematikos magistro baigiamasis darbas

**Kalbos konsultantas**  
**dr. J. Džežulskienė**  
**2005**

**Recenzentas**  
**prof.habil.dr. G.Dzemyda(MII)**  
**2005**

**Recenzentas**  
**prof.habil.dr. J.Poderys(LKKA)**  
**2005**

**Vadovai**  
**prof. dr. Z. Navickas**  
**2005**  
**prof.habil.dr.A.Vainoras(KMU)**  
**2005**

**Atliko**  
**FMMM-3 gr. stud.**  
**D. Ašeriškytė**  
**2005**

**KAUNAS, 2005**

## KVALIFIKACINĖ KOMISIJA

**Pirmininkas:** Leonas Saulis, profesorius (VGTU)

**Sekretorius:** Eimutis Valakevičius, docentas (KTU)

**Nariai:** Algimantas Jonas Aksomaitis, profesorius (KTU)

Vytautas Janilionis, docentas (KTU)

Vidmantas Povilas Pekarskas, profesorius (KTU)

Rimantas Rudzkis, banko NORD/LB vyriausiasis analitikas

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

Ašeriškytė D. Counting of Fractal Dimensions for Some Physiological Process Realizations of Human Organism: Master's Work in Applied Mathematic/ supervisors prof. dr. Z. Navickas; Department of Applied Mathematic, Faculty of Fundamental Sciences, Kaunas University of Technology; prof. habil. dr. A. Vainoras, Kaunas University of Medicine. – Kaunas, 2005. – 85 psl.

## SUMMARY

For correct specification human's physiological state, it is very important to evaluate the changes of main human organism systems. Fractal dimensions of the parameters of the human organism, according to proposed model which includes three functional elements – periphery, regulation and supplying systems were analyzed. The parameters that characterize the function of those systems, that is heart rate, JT interval, systolic and diastolic blood pressure have been studied. Interpolation of discrete data from the physical load obtained by provocative incremental bicycle ergometry stress test was made by cubic spline. For those approximated parameters fractal dimensions (capacity, information, correlation) were counted.

The differences for various groups of persons (sportsmen, healthy persons, patients with ischemic heart disease) were investigated. Fractal dimensions integrates all features of reaction to load and recovery.

The study revealed that distributions of fractal dimensions significantly differs between investigated groups for the most analyzed parameters. It is important that distribution of most studied parameters not differs between groups of different age and gender of the same physical activity.

Existing difference between groups with different physical activity and absence of it in groups of different age and gender but with the same physical fitness could point out the conclusion that studied fractal dimensions could be used as measure of human functional state and healthiness.

## TURINYS

ĮVADAS.....	8
1. TEORINĖ DALIS .....	10
1.1. Funkcijų interpoliavimo sąvoka .....	10
1.2. Lagranžo interpoliacinė išraiška.....	10
1.3. Interpoliavimas tiesine funkcija .....	11
1.4. Interpoliavimas kubiniu splainu .....	12
1.5. Baigtiniai skirtumai .....	14
1.6. Funkcijų suglodinimas .....	15
1.7. Mažiausių kvadratų metodas.....	16
1.8. Įvadas apie fraktalus.....	17
1.9. Metrinės erdvės sąvoka. Metrikos sąvoka.....	18
1.10. Hausdorfo atstumas. Fraktalinės erdvės apibrėžimas .....	20
1.11. Fraktalinė dimensija .....	22
1.12. Pagrindiniai parametrai .....	24
1.13. Amžiaus įtaka širdies ir kraujagyslių sistemos reguliavimui .....	26
1.14. Širdies ir kraujagyslių sistemos funkcinių rodmenų kaita, atliekant fizinio krūvio mėginus .....	27
1.15. Fizinio krūvio įtaka širdies ir kraujagyslių sistemos funkcijai.....	30
1.16. Raumenų kraujotaka.....	30
1.17. Išeminė širdies liga.....	32
1.18. Programavimo kalba .....	34
2. TIRIAMOJI DALIS .....	35
2.1. Duomenys.....	35
2.2. Tyrimas.....	36
2.3. Programos aprašymas.....	43
IŠVADOS.....	45
LITERATŪRA.....	46
PRIEDAI .....	48
1 priedas. Tiriamojo kontingento duomenys .....	48
2 priedas. Duomenų papildymas interpoliuojant Lagranžo polinomu .....	49
3 priedas. Parametrų dimensijų vidurkiai, kai $\epsilon=0,01$ .....	51
4 priedas. Parametrų dimensijų vidurkiai, kai $\epsilon=0,1$ .....	52
5 priedas. Parametrų informacinių dimensijų vidurkių dinamika, kai $\epsilon=0,05$ .....	53
6 priedas. Parametrų užimtumo dimensijų vidurkių dinamika, kai $\epsilon=0,05$ .....	54
7 priedas. Parametrų koreliacinių dimensijų vidurkių dinamika, kai $\epsilon=0,05$ .....	55
8 priedas. Parametrų dimensijų vidurkių dinamika, kai $\epsilon=0,01$ .....	56
9 priedas. Parametrų dimensijų vidurkių dinamika, kai $\epsilon=0,1$ .....	59
10 priedas. Parametrų dimensijų skirstinių palyginimas .....	62
11 priedas. Parametrų dimensijų skirstinių palyginimas pagal amžių .....	65
12 priedas. Parametrų fraktalinių dimensijų vidurkių dinamikos .....	67
13 priedas. Programos tekstų fragmentai .....	68

## LENTELIŲ SĄRAŠAS

1.1 lentelė. Baigtinių skirtumų užrašymas lentelė.....	14
1.2 lentelė. Baigtinių skirtumų skaičiavimo pavyzdys.....	15
2.1 lentelė. Pradinių duomenų struktūros pavyzdys.....	36
2.2 lentelė. Parametrų normos .....	38
2.3 lentelė. Tiriamų grupių parametrų užimtumo dimensijų vidurkiai .....	38
2.4 lentelė. Tiriamų grupių parametrų informacinių dimensijų vidurkiai .....	39
2.5 lentelė. Tiriamų grupių parametrų koreliacinių dimensijų vidurkiai .....	40
2.6 lentelė. Palyginimas parametrų užimtumo dimensijų skirstinių tirtose vyrų grupėse .....	42

## PAVEIKSLŲ SĄRAŠAS

1.1 pav. Taškų jungimas laužtėmis.....	11
1.2 pav. JT intervalo grafinis vaizdas.....	25
1.3 pav. Kraujo tėkmės( ml/ min ) perskirstymas įvairiuose organuose ramybės ir fizinio krūvio metu.....	26
1.4 pav. Žmogaus organizmo funkcinės būklės įvertinimo modelis krūvio metu.....	28
1.5 pav. Integralinio vertinimo modelis ir organizmo sistemų tarpusavio ryšius rodantys žymenys.....	29
2.1 pav. Parametro funkcinė priklausomybė, gauta interpoliuojant kubiniu splineu .....	36
2.2 pav. Taškų sklaidos diagramos pavyzdys .....	37
2.3 pav. Informacinių dimensijų vidurkių dinamika vyrų grupėse .....	40
2.4 pav. Informacinių dimensijų vidurkių dinamika moterų grupėse .....	41
2.5 pav. Programos langas dimensijų analizei atlikti .....	44

## ĮVADAS

Dar Hipokratas 460 – 400 metais prieš Kristaus gimimą sakė: „Visos atliekančios kurią nors funkciją kūno dalys gerai išlavėja, išlieka sveikos ir lėčiau sensta, jei jos naudojamos saikingai ir atlieka užduotis, prie kurių yra įpratusios. Tačiau jei šios kūno dalys nieko neveikia ir paliekamos be darbo, jos neišlavėja, yra linkusios į ligas ir greitai pasensta”. Pastarąjį dešimtmetį atlikti moksliniai tyrimai bei įvairios studijos parodė, kad nepakankamas fizinis aktyvumas tiek sąlyginai sveikiems žmonėms, tiek sergantiems širdies ir kraujagyslių sistemos ligomis gali sukelti nepageidaujamų pasekmių. Ir atvirkščiai, tinkamai parinktas ir adekvačiai dozuojamas fizinis krūvis gali pagerinti gyvenimo kokybę, t.y. fizinio krūvio toleravimą, sumažinti ligos sukeltų simptomų intensyvumą ar neleisti jiems atsirasti.

Vis dažniau galime surasti darbų, kur skaičiuojamos fraktalinės dimensijos. Jos naudojamos įvertinti įvairius gamtos objektus, nuotraukas, elektrokardiogramos rezultatus, medžiagų skirtumus. Pabandėme pritaikyti kelių fraktalinių dimensijų skaičiavimo algoritmus integralinio žmogaus būklės vertinimo modelio tyrimui.

Šiame darbe nagrinėjome parametrus, apibūdinančius žmogaus funkcinės būklės vertinimo modelį, kurį sudaro trys tarpusavyje sąveikaujančios fiziologinės sistemos: periferija, reguliacinė sistema ir aprūpinimo sistema. Tyrėme parametrų dinamiką krūvio metu bei atsigaunant po jo. Skaičiavome užimtumo, informacinę ir koreliacinę dimensijas.

Tyrimui pasirinkome penkias tiriamųjų grupes – profesionalius sportininkus vyrus ir moteris, mažai sportuojančius ir nesportuojančius, neturinčius klinikinių nusiskundimų vyrus ir moteris, išemine širdies liga sergančius vyrus. Nagrinėjome, ar skiriasi jų parametrų apskaičiuotosios dimensijos.

Sukurta taikomoji programa, kuri saugo duomenis, juos analizuoja, grupuoja, atlieka reikalingus skaičiavimus, rezultatus įvairia forma pateikia vartotojui (grafikai, skaitmeninė informacija) ekrane. Programa parašyta „C++ Builder” aplinkoje. Statistinė gautų rezultatų analizė atlikta SPSS paketu.

Mokslinė veikla:

Pranešimai tarptautinėse konferencijose:

Biomedicininė inžinerija (1, 2, 3)

Medicon and Health telematics (4)

Pranešimai studentų konferencijoje:

Taikomoji matematika (5, 6)



## Pranešimai konferencijoje

Matematika ir matematikos dėstymas

„Informacinės dimensijos tyrimas, naudojant žmogaus organizmo fiziologinių funkcijų modelį“

„Informacinės sistemos e – sveikatai kūrimas, panaudojant Milio ir Muro automatų sąsūką“

## Priimtas straipsnis į žurnalą „Medicina“

„Sistolinio kraujo spaudimo prognozavimo galimybė asmeniui, remiantis elektrokardiogramos parametrais“

Jaunųjų mokslininkų darbų parodoje – konkurse KTU TECHNORAMA' 2004 laimėta III – oji vieta už darbą „Fizinio krūvio efektų vertinimo metodikos“

## 1. TEORINĖ DALIS

### 1.1. Funkcijų interpoliavimo sąvoka

Gamtos moksluose ir technikoje argumentas ir funkcijos reikšmė nustatoma stebėjimais. Stebėtojas tam tikrais laiko momentais išmatuoja ir užrašo argumento  $x_i$  ir funkcijos  $y=f(x_i)$  reikšmes. Duomenys surašomi į lentelę, kuria išreiškiama funkcijos priklausomybė nuo argumento atskiruose taškuose. Gautoji lentelė naudojama teoriniam ar praktiniam reiškinių tyrimui.

**Apibrėžimas.** Funkcijos  $f(x)$  duotos reikšmių lentelė, išreiškimas analizine formule vadinamas interpoliavimu, jei reikšmės, gautos pagal formulę taškuose  $x_i$ , sutampa su funkcijos reikšmėmis lentelėje.

Apskritai interpoliavimo sąvoka yra platesnė.

Tarkime duota funkcijos  $y=f(x)$  reikšmių lentelė  $(x_i, y_i)$ ; čia  $i=0, 1, \dots, n$ ,  $y_i$  reikšmės yra tikslios arba paklaidos tokios mažos, kad praktiškai jų galima nepaisyti. Reikia rasti aproksimuojančią funkciją  $y=F(x)$ , priklausančią funkcijų klasei  $K$  ir tenkinančią sąlygas

$$F(x_i) = y_i \quad i=0, 1, \dots, n \quad (1.1)$$

Šios sąlygos vadinamos Lagranžo interpoliavimo sąlygomis, o pati funkcija  $y=F(x)$  – Lagranžo interpoliacinė funkcija, arba tiesiog interpoliacinė funkcija (7).

### 1.2. Lagranžo interpoliacinė išraiška

Ieškosime  $n$ -tos eilės daugianario  $F(x)$  tenkinančio Lagranžo interpoliavimo sąlygas (1.1) formulė. Išnagrinėkime polinomus:

$$F_i(x) = \frac{(x-x_0)\dots(x-x_{i-1})(x-x_{i+1})\dots(x-x_n)}{(x_i-x_0)\dots(x_i-x_{i-1})(x_i-x_{i+1})\dots(x_i-x_n)}, \quad i = \overline{0, n}.$$

Jie turi tokią savybę:  $F_i(x_j) = \begin{cases} 1, & \text{kai } i = j, \\ 0, & \text{kai } i \neq j; \end{cases} \quad j \in \{0, 1, \dots, n\}.$

Pasinaudodami polinomais  $F_i(x)$ , galime iš karto parašyti interpoliacinio polinomo  $F(x)$  išraišką:

$$F(x) = \sum_{i=0}^n F_i(x) y_i. \quad (1.2)$$

Šis polinomas interpoliacinis, nes jis yra  $n$ -tos eilės polinomas ir tenkina Lagranžo interpoliavimo sąlygas. ( 1.2 ) interpoliacinio polinomo išraiška vadinama Lagranžo interpoliacine išraiška, todėl literatūroje simbolis  $F(x)$  keičiamas  $L_n(x)$ .

### 1.3. Interpoliavimas tiesine funkcija

Sakykime turime taškus  $x_0, x_1, \dots, x_n$  ir funkcijos  $y=f(x)$  reikšmes  $f(x_0), f(x_1), \dots, f(x_n)$ . Iš visų interpoliacinių polinomų paprasčiausias yra pirmojo laipsnio interpoliacinis polinomas. Jį apskaičiuojame, naudodamiesi dviem argumentu ir funkcijos reikšmėmis.

Užrašykime Lagranžo interpoliacinį polinomą:

$$L_n(x) = \sum_{k=0}^n \frac{(x-x_0) \dots (x-x_{k-1})(x-x_{k+1}) \dots (x-x_n)}{(x_k-x_0) \dots (x_k-x_{k-1})(x_k-x_{k+1}) \dots (x_k-x_n)} f(x_k).$$

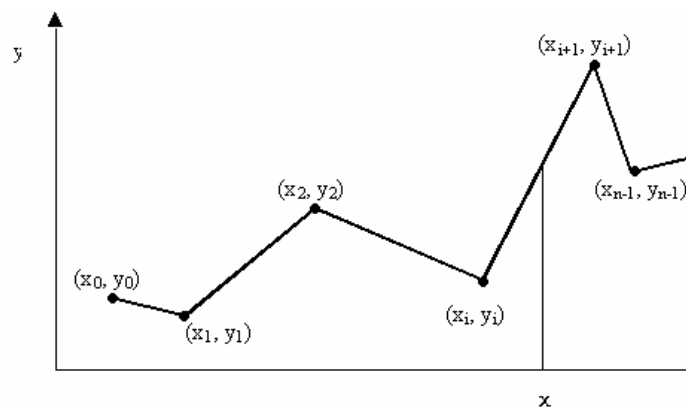
Jei  $f(x_i)$  ir  $f(x_{i+1})$  yra dvi minėtos funkcijos reikšmės, tai paėmę  $n = 1$ , gauname:

$$L_1(x) = \frac{x-x_{i+1}}{x_i-x_{i+1}} f(x_i) + \frac{x-x_i}{x_{i+1}-x_i} f(x_{i+1}) = f(x_i) + \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} (x-x_i).$$

Tokiu būdu, pirmojo laipsnio interpoliacinis polinomas funkcijai, kurios reikšmės žinomos  $n+1$  taškuose  $x_0, x_1, \dots, x_n$ , išreiškiamas formule:

$$L_1(x) = f(x_i) + \frac{f(x_{i+1})-f(x_i)}{x_{i+1}-x_i} (x-x_i), \quad \text{jei } x_i \leq x \leq x_{i+1} \text{ ir } (i = 0, 1, 2, \dots, n-1).$$

Interpoliavimas pirmojo laipsnio polinomais vadinamas tiesiniu interpoliavimu. Tiesinio interpoliavimo formulės (polinomo) grafikas yra laužtė, jungianti taškus  $(x_i, f(x_i))$ , ( $i = 0, 1, \dots, n$ ) (1.1 pav.).



1.1 pav. Taškų jungimas laužtėmis

## 1.4. Interpoliavimas kubiniu splainu

Splainas – tai tolydžioji iki  $p$ -tosios eilės išvestinės imtinai funkcija, sudaryta iš kurios nors funkcijos dalių. Istoriskai splainai pradėti konstruoti iš  $n$ -tosios eilės polinomo dalių, todėl čia, kalbėdami apie splainus, laikysime juos funkcijomis, sudarytomis iš polinomo dalių. Praktikoje dažniausiai naudojami kubiniai splainai.

**Uždavinio formulotė.** Duota funkcijos  $y = f(x)$  reikšmių lentelė  $(x_i, y_i)$ ,  $i = \overline{0, N}$ . Reikia rasti kubinį splainą  $y = g(x)$ , tenkinantį Lagranžo interpoliavimo sąlygą

$$g(x_i) = y_i, i = \overline{0, N}.$$

Kad interpoliacinis kubinis splainas būtų apibrėžtas vienareikšmiškai, pasirenkamos kraštinės sąlygos

$$g''(x_0) = g''(x_N) = 0.$$

Jos dar vadinamos natūraliomis kraštinėmis sąlygomis ir rodo, kad plieninės liniuotės, išraitytos per interpoliavimo taškus, galai paliekami laisvai kaboti.

Kadangi splaino antroji išvestinė yra tolydžioji ir tiesinė kiekviename intervale  $[x_{i-1}, x_i]$ , ( $i = \overline{1, N}$ ), tai galima rašyti:

$$g''(x) = m_{i-1} \frac{x_i - x}{h_i} + m_i \frac{x - x_{i-1}}{h_i}; \quad (1.3)$$

čia  $m_i = g''(x_i)$ ,  $i = \overline{0, N}$ , o  $h_i = x_i - x_{i-1}$ .

(1.3) lygybę suintegravę du kartus, gauname:

$$g(x) = m_{i-1} \frac{(x_i - x)^3}{6h_i} + m_i \frac{(x - x_{i-1})^3}{6h_i} + A_i \frac{x_i - x}{h_i} + B_i \frac{x - x_{i-1}}{h_i}; \quad (1.4)$$

čia  $A_i, B_i$  – integravimo konstantos.

$A_i, B_i$  parinkime taip, kad  $g(x)$  tenkintų interpoliavimo sąlygą, t.y. kad būtų

$$g(x_{i-1}) = y_{i-1} \text{ ir } g(x_i) = y_i.$$

Iš (1.4) lygybės gauname:

$$A_i = y_{i-1} - \frac{m_{i-1}h_i^2}{6}, B_i = y_i - \frac{m_i h_i^2}{6}.$$

Vadinasi, kubinio splaino išraiška intervale  $[x_{i-1}, x_i]$  yra tokia:

$$g(x) = m_{i-1} \frac{(x_i - x)^3}{6h_i} + m_i \frac{(x - x_{i-1})^3}{6h_i} + \left(y_{i-1} - \frac{m_{i-1}h_i^2}{6}\right) \frac{x_i - x}{h_i} + \left(y_i - \frac{m_i h_i^2}{6}\right) \frac{x - x_{i-1}}{h_i}. \quad (1.5)$$

(1.5) formulės nežinomi dydžiai  $m_i$  yra splaino antrųjų išvestinių reikšmės taškuose  $x_i$ . Jas apskaičiuojame remdamiesi sąlyga

$$g'(x_i - 0) = g'(x_i + 0) \quad (i = \overline{1, N-1}) \quad (1.6)$$

ir tardami, kad  $m_0 = m_N = 0$ , t.y. pasinkdami natūralias kraštines sąlygas.

Nesunku įsitikinti, kad

$$g'(x_i - 0) = \frac{h_i}{6} m_{i-1} + \frac{h_i}{3} m_i + \frac{y_i - y_{i-1}}{h_i},$$

$$g'(x_i + 0) = -\frac{h_{i+1}}{3} m_i - \frac{h_{i+1}}{6} m_{i+1} + \frac{y_{i+1} - y_i}{h_{i+1}}.$$

Tada (1.6) lygčių sistemą galima užrašyti matricine išraiška:

$$Am = Hy; \quad (1.7)$$

čia  $A$  – simetrinė kvadratinė  $(N-1)$ -osios eilės trijstrižainė matrica

$$A = \begin{bmatrix} \frac{h_1 + h_2}{3} & \frac{h_2}{6} & 0 & \dots & 0 & 0 \\ \frac{h_2}{6} & \frac{h_2 + h_3}{3} & \frac{h_3}{6} & \dots & 0 & 0 \\ 0 & \frac{h_3}{6} & \frac{h_3 + h_4}{3} & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & \dots & \frac{h_{N-1}}{6} & \frac{h_{N-1} + h_N}{3} \end{bmatrix},$$

$H$  – stačiakampė  $(N-1) \times (N+1)$  matrica

$$H = \begin{bmatrix} \frac{1}{h_1} & -\left(\frac{1}{h_1} + \frac{1}{h_2}\right) & \frac{1}{h_2} & 0 & \dots & 0 & 0 & 0 \\ 0 & \frac{1}{h_2} & -\left(\frac{1}{h_2} + \frac{1}{h_3}\right) & \frac{1}{h_3} & \dots & 0 & 0 & 0 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0 & 0 & 0 & 0 & \dots & \frac{1}{h_{N-1}} & -\left(\frac{1}{h_{N-1}} + \frac{1}{h_N}\right) & \frac{1}{h_N} \end{bmatrix},$$

$m$ ,  $y$  – matricos stulpeliai

$$m = \begin{bmatrix} m_1 \\ m_2 \\ \dots \\ m_{N-1} \end{bmatrix}, \quad y = \begin{bmatrix} y_0 \\ y_1 \\ \dots \\ y_N \end{bmatrix}.$$

(1.7) sistema yra tiesinių lygčių sistema su trijstrižaine matrica  $A$ , kurios pagrindinės įstrižainės elementai didesni už kitų eilutės elementų sumą. Taigi ši sistema yra suderintoji ir turi vienintelį sprendinį (7).

## 1.5. Baigtiniai skirtumai

Jei atstumai tarp interpoliavimo taškų vienodi patogiu naudotis vadinamaisiais baigtiniais skirtumais. Skirtumai

$$\Delta f(x_i) = f(x_{i+1}) - f(x_i) \quad (i = 0, 1, 2, \dots, n-1)$$

vadinami pirmos eilės baigtiniais skirtumais.

Analogiškai,  $k$ -tos eilės baigtiniais skirtumais vadinami dydžiai

$$\Delta^k f(x_i) = \Delta^{k-1} f(x_{i+1}) - \Delta^{k-1} f(x_i) \quad (k=1, 2, 3, \dots). \quad (1.8)$$

Baigtiniai skirtumai surašomi į tokią lentelę:

1.1 lentelė

Baigtinių skirtumų užrašymas lentelė

$x_i$	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$	...
$x_0$	$f(x_0)$				
$x_1$	$f(x_1)$	$\Delta f(x_0)$			
$x_2$	$f(x_2)$	$\Delta f(x_1)$	$\Delta^2 f(x_0)$		
$x_3$	$f(x_3)$	$\Delta f(x_2)$	$\Delta^2 f(x_1)$	$\Delta^3 f(x_0)$	
...	...	...	...	...	...

Dabar ištersime baigtinių skirtumų savybes.

**Lema.**  $k$ -tos eilės baigtinių skirtumų suma yra lygi skirtumui tarp paskutiniojo ir pirmojo ( $k-1$ )-os eilės baigtinių skirtumų:

$$\sum_{i=0}^{n-k} \Delta^k f(x_i) = \Delta^{k-1} f(x_{n-k+1}) - \Delta^{k-1} f(x_0).$$

Irodymas. Jei yra žinoma ( $n+1$ ) funkcijos reikšmių  $f(x_i)$  interpoliavimo taškuose  $x_i = x_0 + ih$  ( $i = 0, 1, 2, \dots, n$ ), tai galima apskaičiuoti ( $n-k+1$ )  $k$ -tos eilės baigtinių skirtumų  $\Delta^k f(x_0)$ ,  $\Delta^k f(x_1)$ , ...,  $\Delta^k f(x_{n-k})$ .

Remdamiesi baigtinių skirtumų apibrėžimu ((1.8) formule), gauname:

$$\begin{aligned} \Delta^k f(x_0) + \Delta^k f(x_1) + \dots + \Delta^k f(x_{n-k-1}) + \Delta^k f(x_{n-k}) &= \Delta^{k-1} f(x_1) - \Delta^{k-1} f(x_0) + \Delta^{k-1} f(x_2) - \Delta^{k-1} f(x_1) + \dots + \\ + \Delta^{k-1} f(x_{n-k}) - \Delta^{k-1} f(x_{n-k-1}) + \Delta^{k-1} f(x_{n-k+1}) - \Delta^{k-1} f(x_{n-k}) &= \Delta^{k-1} f(x_{n-k+1}) - \Delta^{k-1} f(x_0). \end{aligned}$$

Tai ir reikėjo įrodyti.

Naudojantis šia savybe, galima patikrinti, ar tiksliai apskaičiuoti baigtiniai skirtumai. Tuo tikslu 1.1 lentelėje papildomai prijungiamos dvi eilutės:  $(n + 2)$ -je užrašoma stulpelio suma, o  $(n + 3)$ -je-apatinio ir viršutinio stulpelio elementų skirtumas.  $(n + 2)$ -ji eilutė žymima simboliu  $\Sigma$ , o  $(n + 3)$ -ji – simboliu S. Jeigu baigtiniai skirtumai buvo apskaičiuoti teisingai, tai  $\Sigma$ -os eilutės  $j$ -jo stulpelio skaičius bus lygus S-os eilutės  $(j - 1)$ -jo stulpelio skaičiui.

Kaip tai atliekama, matyti iš 1.2 lentelės.

**1.2 lentelė**

**Baigtinių skirtumų skaičiavimo pavyzdys**

$x_i$	$f(x_i)$	$\Delta f(x_i)$	$\Delta^2 f(x_i)$	$\Delta^3 f(x_i)$
1	2			
2	4	2		
3	15	11	9	
4	36	21	10	1
$\Sigma$	-	34	19	1
S	34	19	1	-

Šioje lentelėje baigtiniai skirtumai apskaičiuoti teisingai, nes atitinkami  $\Sigma$  ir S eilučių elementai lygūs. (8).

## 1.6. Funkcijų suglodinimas

Nagrinėdami interpoliavimo uždavinį, funkciją  $f(x)$ , apibrėžtą reikšmių lentele, stengėmės pakeisti tokia aproksimuojančiąja funkcija  $F(x)$  ( $n$ -tojo laipsnio polinomu ir pan.), kad taškuose  $x_i$  ( $i = 0, \dots, n$ )  $F(x_i)$  reikšmės būtų lygios  $f(x_i)$  reikšmėms.

Labai dažnai  $f(x_i)$  reikšmės yra eksperimento rezultatai ir turi matavimo bei metodo paklaidų. Todėl reikalauti, kad aproksimuojančioji funkcija  $F(x)$  tenkintų sąlygą  $F(x_i) = f(x_i)$  ( $i = 0, \dots, n$ ), būtų neprotinga. Geriau rasti tokią funkciją  $F(x)$ , kuri pagal pasirinktą kriterijų geriausiai aproksimuotų  $f(x)$ . Toks funkcijos  $F(x)$  apskaičiavimo metodas vadinamas suglodinimu. Atsižvelgiant į suglodinimo kriterijų, galima gauti įvairias  $F(x)$  išraiškas.

**Uždavinio formulavimas.** Sakykime, funkcija  $y = f(x)$  nusakyta reikšmių lentele  $(x_i, \tilde{y}_i)$ ; čia  $i = 1, \dots, m$ . Simbolis  $\tilde{y}_i$  rodo, kad  $f(x)$  reikšmės taškuose  $x_i$  yra apytikslės. Taip pat žinoma

aprosimuojančiosios funkcijos  $F(x)$  analizinė išraiška:  $F(x, a_0, \dots, a_n)$ ; čia  $a_i$  ( $i = 0, \dots, n$ ) – nežinomi parametrai ir  $n \ll m$ . Reikia rasti tokias parametrų  $a_i$  reikšmes, su kuriomis  $F(x)$  geriausiai aproksimuotų funkciją  $f(x)$ . Sprendžiant šį uždavinį, taikomi įvairūs suglodinimo metodai. Vienas jų – mažiausių kvadratų metodas (9).

## 1.7. Mažiausių kvadratų metodas

Dažno eksperimento tikslas- nustatyti tam tikro dydžio  $y$  priklausomybę nuo kito dydžio  $x$ , imant baigtinį skaičių atskirų reikšmių  $x_i$ ,  $i = 1, 2, \dots, n$ . Tokio eksperimento rezultatas –  $x_i$  ir jas atitinkančių eksperimentinių  $y_i$  reikšmių lentelė arba kreivė, nubrėžta per taškus  $(x_i, y_i)$ . Išnagrinėsime vieną būdą, kaip gauti tą kreivę atitinkančios analizinės funkcijos išraišką. Tokias funkcijas vadinsime empirinėmis, (gr. *empeiria*- „pažinimas, paremtas patyrimu“) o jų ieškojimo procesą- aproksimavimu (lot. *approximo*- „artėju“). Kadangi kiekvienos funkcijos analizinėje išraiškoje, be jos argumento  $x$ , yra ir tam tikri skaitiniai koeficientai, tai bendroji empirinės funkcijos analizinė išraiška yra tokia:

$$y=f(x, a, b, c, \dots). \quad (1.9)$$

Išsiaiškinkime, kaip reikia parinkti empirinės funkcijos parametrus  $a, b, c, \dots$ , kad gauta empirinė funkcija geriausiai atitiktų eksperimentu nustatytą priklausomybę. Tokio uždavinio sprendimas priklauso nuo to, ką sutarsime laikyti „geriausiu“ atitikimu. Galima, pavyzdžiui, reikalauti, kad maksimalūs atstumai tarp empirinės (1.9) kreivės ir eksperimento taškų arba atstumų tarp empirinės kreivės ir eksperimento taškų moduliai būtų minimalūs. Labiausiai paplitęs tokių uždavinių sprendimo būdas yra mažiausių kvadratų metodas.

Sakykime, kad argumento  $x_i$  reikšmę atitinka eksperimentu gauta  $y_i$  reikšmė. Įrašę  $x_i$  reikšmę į empirinės funkcijos (1.9) išraišką, gausime funkcijos reikšmę  $f(x_i, a, b, c, \dots)$ , paprastai nelygią  $y_i$ , nes  $y_i$  reikšmėms turi įtaką eksperimento paklaidos. Skirtumą  $y_i - f(x_i, a, b, c, \dots)$  vadiname empirinės funkcijos nuokrypiu. Mažiausių kvadratų metodo reikalavimas yra toks: parametrai  $a, b, c, \dots$  turi būti tokie, kad nuokrypių kvadratų suma būtų minimali. Taigi

$$S = \sum_{i=1}^n (y_i - f(x_i, a, b, c, \dots))^2 \rightarrow \min.$$

Tokiu atveju funkcijos  $S$  dalinės išvestinės kintamųjų  $a, b, c, \dots$  atžvilgiu turi būti lygios nuliui:

$$\frac{\partial S}{\partial a} = 0, \frac{\partial S}{\partial b} = 0, \frac{\partial S}{\partial c} = 0, \dots$$

Išdiferencijavę  $S$  parametrų  $a, b, c, \dots$  atžvilgiu ir prilyginę gautas dalines išvestines nuliui, gauname lygčių sistemą, sudarytą iš tiek lygčių, kiek parametrų turi funkcija  $y=f(x, a, b, c, \dots)$ :



$$\begin{cases} \sum_{i=1}^n (y_i - f(x_i, a, b, c, \dots)) \frac{\partial f}{\partial a} = 0, \\ \sum_{i=1}^n (y_i - f(x_i, a, b, c, \dots)) \frac{\partial f}{\partial b} = 0, \\ \dots \end{cases} \quad (1.10)$$

Išnagrinėsime dažniausią atvejį, kai funkcija  $y = f(x, a, b, c, \dots)$  yra tiesinė, t.y.

$$y = ax + b.$$

Kadangi  $\frac{\partial f}{\partial a} = x$ ,  $\frac{\partial f}{\partial b} = 1$ , tai iš (1.10) sistemos gauname

$$\begin{cases} \sum_{i=1}^n (y_i - (ax_i + b))x_i = 0, \\ \sum_{i=1}^n (y_i - (ax_i + b)) = 0, \end{cases}$$

kuri pertvarkoma į tokią:

$$\begin{cases} a \sum_{i=1}^n x_i^2 + b \sum_{i=1}^n x_i = \sum_{i=1}^n x_i y_i, \\ a \sum_{i=1}^n x_i + nb = \sum_{i=1}^n y_i. \end{cases}$$

Pasinaudoję Kramerio formulėmis, gauname:

$$\begin{aligned} a &= \frac{\sum_{i=1}^n x_i \sum_{i=1}^n y_i - n \sum_{i=1}^n x_i y_i}{\left(\sum_{i=1}^n x_i\right)^2 - n \sum_{i=1}^n x_i^2}, \\ b &= \frac{\sum_{i=1}^n x_i y_i \sum_{i=1}^n x_i - \sum_{i=1}^n y_i \sum_{i=1}^n x_i^2}{\left(\sum_{i=1}^n x_i\right)^2 - n \sum_{i=1}^n x_i^2}. \end{aligned} \quad (1.11)$$

## 1.8. Įvadas apie fraktalus

Gamtos mokslas ir matematika visais laikais ieškojo tvarkos Visatoje. Paskutiniai bandymai šioje paieškoje siejami su chaoso teorijos atsiradimu. Kodėl ši teorija vadinama chaoso teorija? Todėl, kad ji leidžia apčiuopti tam tikrą tvarką reiškiniuose, kuriuos iki šiol žmonės vadino ir laikė visiškai chaotiškais. Nors, antra vertus, tie reiškiniai vis tiek išlieka sunkiai valdomais ir prognozuojamais.

Iki šių dienų linija buvo pagrindinis statybinis blokas, leidžiantis suprasti ir pavaizduoti Visatą. Chaoso teorija naudoja kitą geometriją – fraktalinę. Kas yra fraktalas? Sunku kol kas duoti formalų

apibrėžimą. Galima būtų teigti, kad tikrasis fraktalas – tai be galo daug kartų savyje atsikartojantis objektas. Pats fraktalų pradininkas amerikiečių matematikas B. Mandelbrotas (Benoit B. Mandelbrot) yra įsitikinęs, kad gamtoje iš vis nėra glodžių objektų – tik fraktaliniai. Vien dėl to, kad mūsų matematinis aparatas yra silpnas, mes turime realius (fraktalinius) objektus pakeisti idealiais (bet tikrovėje neegzistuojančiais) objektais – glodžiosiomis linijomis ir paviršiais, elipsėmis ir parabolėmis, plokštumomis, elipsoidais ir panašiai. Apie 1964 metus jisai pradėjo suvokti, kad įvairius reiškinius, painias figūras, jų konstravimo procedūras galima apibendrinti, sistematizuoti. To dar niekas nebuvo padaręs. 1975 m. B. Mandelbrotas sugalvojo ir įvedė terminą *fraktalas*, norėdamas duoti savo pirmajai publikacijai šiais klausimais pavadinimą. Tačiau paaiškinti šio termino nesugebėjo ir neskubėjo, sakydamas, jog „gerą vyną irgi reikia išlaikyti prieš pilstant į butelius...“. Ir vis dėlto, kai kurie bruožai, charakteringi įvestiems fraktalams, jau tada buvo akivaizdūs:

- ✓ visos figūros, kurias B. Mandelbrotas tyrė ir vadino fraktalais, turėjo savybę – nereguliarios, bet atsikartojančios savyje, t.y. žiūrint pro didinamąjį stiklą į atskirą figūros fragmentą, pastarasis primindavo visą figūrą;

- ✓ fraktalus (kaip aibes) sudaro be galo daug taškų, kurių tarpusavio išsidėstymas toks sudėtingas, kad neįmanoma nei perprasti, nei aprašyti jų geometrijos.

Klasikinė geometrija – tai tik pirmoji realių objektų struktūros aproksimacija. Ji tinka, kai projektuose norime pateikti technologinį produktą, ir yra labai netiksli, kai siekiame aprašyti realius gamtos kūrinius.

Fraktalinė geometrija – tai klasikinės geometrijos plėtinys. Jos pagalba galima kurti tikslus matematinis fizinių struktūrų modelius (nuo paparčio lapo iki galaktikų). Fraktalų teorija labai patraukli ir aktuali. Ji keičia tradicinį požiūrį į įprastus reiškinius. Fraktalų teorija – tai kompiuterinės eros darinys.

Fraktalai apibrėžiami, aprašant sąryšius tarp atskirų (fraktalinės) aibės fragmentų, panašiai kaip, tarkime, Saulės sistemoje, kur fiksuojamos ne atskirų dangaus objektų koordinatės, o pateikiami gravitacijos dėsniai bei nurodomos pradinės sąlygos.

Pats terminas fraktalas paimtas iš lotynų kalbos: *fractus* reiškia nereguliarią fragmentą, *frangere* – suskaidymą į nereguliarus fragmentus (10, 11, 12).

## 1.9. Metrinės erdvės sąvoka. Metrikos sąvoka

Fraktalinė geometrija nagrinėja gana paprastų geometrinių erdvių poaibių struktūrą. Pati geometrinė erdvė žymima  $X$ .

**Apibrėžimas.** Erdvė vadinama aibe  $X$ , aibės elementai vadinami erdvės taškais.

Erdvės sąvoka reiškia, jog yra tam tikras požymis, jungiantis visus elementus į aibę.

Keletas erdvės pavyzdžių.

1.  $X = \mathbf{R}$ . Kiekvienas taškas  $x \in X$  yra realusis skaičius (požymis, jungiantis taškus į aibę), arba taškas skaičių tiesėje.
2.  $X = \mathbf{R}^2$ . Tai euklidinė plokštuma. Kiekviena realiųjų skaičių pora  $x_1, x_2 \in \mathbf{R}$  nusako atskirą tašką erdvėje  $\mathbf{R}^2$ . Kiekvieną tašką  $x \in X$  galima užrašyti  $x = (x_1, x_2)$ .

Tai tiesinės erdvės – jose apibrėžta, kaip sudėti du erdvės taškus, norint gauti naują (trečiąją) tašką toje pačioje erdvėje, būtent:

Jei  $x, y \in X = \mathbf{R}$ , tai  $x + y \in \mathbf{R}$ .

Jei  $x, y \in \mathbf{R}^2$ , tai  $x + y = (x_1, x_2) + (y_1, y_2) = (x_1 + x_2, y_1 + y_2) \in \mathbf{R}^2$ .

Be to, šių erdvių narius galima dauginti iš skaliaro, t.y. iš  $\alpha \in \mathbf{R}$ .

3.  $X = I = \{x \mid x = (x_1, x_2) \in \mathbf{R}^2, 0 \leq x_1 \leq 1, 0 \leq x_2 \leq 1\}$ . Tai erdvė, sutampanti su užpildytu kvadratu.

**Apibrėžimas.** Metrinerdve  $(X, d)$  vadinama erdvė  $X$  su joje apibrėžta realiaja funkcija (metrika)  $d: X \times X \rightarrow \mathbf{R}$ , nusakančia atstumą tarp bet kurių dviejų erdvės  $X$  taškų. Reikalaujama, kad funkcija  $d$  tenkintų tokias aksiomas:

1.  $d(x, y) = d(y, x), \quad \forall x, y \in X;$
2.  $0 < d(x, y) < \infty, \quad \forall x, y \in X, x \neq y;$
3.  $d(x, x) = 0, \quad \forall x \in X;$
4.  $d(x, y) \leq d(x, z) + d(z, y), \quad \forall x, y, z \in X.$

Trumpiausio kelio tarp erdvės taškų sąvoka siejama su metrika toje erdvėje ir priklauso nuo jos. Vienas iš metrikos pavyzdžių  $X = \mathbf{R}, d(x, y) = |x - y|, \forall x, y \in X$ . Tai euklidinė metrika.

**Apibrėžimas.** Dvi metrikos,  $d_1$  ir  $d_2$ , erdvėje  $X$  vadinamos ekvivalenčiomis, jeigu yra skaičiai  $c_1$  ir  $c_2$  ( $0 < c_1 < c_2 < \infty$ ) tokie, kad

$$c_1 d_1(x, y) \leq d_2(x, y) \leq c_2 d_1(x, y), \quad \forall (x, y) \in X \times X.$$

Jei metrikos  $d_1$  ir  $d_2$  yra ekvivalenčios erdvėje  $X$ , tai jos vienodai interpretuoja, kurie  $X$  taškai yra artimi ir kurie nutolę vienas nuo kito. Perėjimą nuo vienos metrikos  $d_1$  erdvėje  $X$  prie kitos jai ekvivalenčios metrikos  $d_2$  galima įsivaizduoti kaip pačios erdvės baigtinį (aprežtą) deformavimą. Kitaip sakant ekvivalentumo reikalavimas – tai reikalavimas, kad nebūtų begalinio erdvės „ištempimo“ arba visiško jos „suspaudimo“. Imkime metrinerdve  $(X, d)$ .

**Apibrėžimas.** Sakoma, jog erdvėje  $X$  apibrėžta transformacija (atvaizdis, arba funkcija)  $f$ , igyjanti reikšmes erdvėje  $X$ , jeigu kiekvieną tašką  $x \in X$  atitinka vienas ir tik tai vienas taškas  $f(x) \in X$ . Jeigu  $S \subset X$ , tai  $f(S) = \{f(x) \mid x \in S\}$ . Kai  $f(X) = X$ , sakome, jog  $f$  yra siurjekcinė transformacija.

Transformacija  $f$  yra abipusiškai vienareikšmė (arba injekcinė), jeigu  $f(x) = f(y)$  ( $x, y \in X$ ) implikuoja  $x = y$ .

Transformacija  $f$  yra apgręžiama (arba bijekcinė), jeigu ji yra abipusiškai vienareikšmė ir  $f(X)=X$ . Šiuo atveju galima apibrėžti transformaciją  $f^{-1}: X \rightarrow X$  (vadinamą atvirkštine duotajai transformacijai  $f$ ) tokiu būdu:  $f^{-1}(y) = x$ ; čia  $x \in X$  yra vienintelis taškas toks, kad  $y = f(x)$ .

Imkime dvi metrinės erdves  $(X_1, d_1)$  ir  $(X_2, d_2)$ .

**Apibrėžimas.** Metrinės erdves  $(X_1, d_1)$  ir  $(X_2, d_2)$  vadinamos ekvivalenčiomis, jeigu yra apgręžiamoji transformacija  $h: X_1 \rightarrow X_2$  tokia, kad metrika  $\tilde{d}_1$  erdvėje  $X_1$ , nusakoma lygybe

$$\tilde{d}_1(x, y) = d_2(h(x), h(y)), \quad \forall (x, y) \in X_1,$$

yra ekvivalenti metrikai  $d_1$ .

**Apibrėžimas.** Metrinės erdves  $(X, d)$  taškų seka  $\{x_n\}_{n=1}^{\infty}$  vadinama Koši seka, jeigu

$$(\forall \varepsilon > 0)(\exists N)(\forall n, m > N) : d(x_n, x_m) < \varepsilon.$$

Taigi, kuo toliau „einam išilgai sekos“, tuo artimesni (metrikos  $d$  prasme) tampa sekos taškai (nariai). Bet, tai nereiškia, kad sekos nariai artėja prie kokio nors fiksuoto taško. Galimas atvejis, jog sekos nariai artėja prie taško, kuris nepriklauso erdvei  $(X, d)$ .

**Apibrėžimas.** Metrinė erdvė  $(X, d)$  vadinama pilnaja, jeigu kiekviena erdves  $X$  Koši seka  $\{x_n\}_{n=1}^{\infty}$  turi ribą  $x \in X$ .

Vadinasi, jeigu  $\{x_n\}_{n=1}^{\infty}$  yra erdves  $X$  taškų Koši seka ir  $(X, d)$  yra pilnoji metrinė erdvė, tai yra taškas  $x \in X$  toks, kad su bet koku  $\varepsilon > 0$  rutulys  $B(x, \varepsilon)$  talpina be galo daug sekos taškų  $x_n$ .

**Apibrėžimas.** Tarkime, kad  $S \subset X$  yra metrinės erdves  $(X, d)$  poaibis. Poaibis  $S$  yra kompaktinis, jeigu iš kiekvienos begalinės poaibio  $S$  sekos  $\{x_n\}_{n=1}^{\infty}$  galima išskirti posekį, turintį ribą poaibyje  $S$ .

## 1.10. Hausdorfo atstumas. Fraktalinės erdves apibrėžimas

Nagrinėsime erdvę  $X \subset \mathbf{R}^2$  su joje apibrėžta euklidine metrika  $d$ .

**Apibrėžimas.** Tarkime, kad  $(X, d)$  yra pilnoji metrinė erdvė. Visų netuščiųjų kompaktinių  $X$  poaibių aibę vadinsime Hausdorfo erdve ir žymėsime  $H(X)$ .

Jeigu  $A, B \in H(X)$ , tai  $A \cup B \in H(X)$ , bet  $A \cap B$  nebūtinai priklauso erdvei  $H(X)$ .

**Apibrėžimas.** Tarkime, kad  $x \in X$  ir  $B \in H(X)$ . Skaičius

$$d(x, B) = \min_{y \in B} \{d(x, y)\}$$

vadinamas taško  $x \in X$  atstumu iki aibės  $B$ .

Gali kilti klausimas – kaip galima teigti, jog realiųjų skaičių aibė  $\{d(x, y) \mid y \in B\}$  turi mažiausią reikšmę? Galima, kadangi  $B$  yra netuščiasis ir kompaktinis erdves  $X$  poaibis.

**Apibrėžimas.** Tarkime, kad  $(X, d)$  yra pilnoji metrinė erdvė, ir  $A, B \in H(X)$ . Skaičius

$$d(A, B) = \max_{x \in A} \{d(x, B)\}$$

vadinamas atstumu nuo aibės  $A$  iki aibės  $B$ .

Išvardinsime svarbesnes su atstumo sąvoka susijusias savybes.

1. Jeigu  $B, C \in H(X)$  ir  $x \in X$ , tai  $(B \subset C) \Rightarrow (d(x, C) \leq d(x, B))$ .

Iš tikrųjų, su bet koku tašku  $x \in X$  turime:

$$d(x, C) = \min_{y \in C} \{d(x, y)\} = \min(\min_{y \in B} \{d(x, y)\}, \min_{y \in C \setminus B} \{d(x, y)\}) \leq \min_{y \in B} \{d(x, y)\} = d(x, B).$$

2. Jeigu  $A, B \in H(X)$ , tai (bendru atveju)  $d(A, B) \neq d(B, A)$ .

Iš tikrųjų, pakanka paimti du poaibius  $A$  ir  $B$  tokius, kad  $B \subset A$ , ir tarti, jog yra taškas  $x \in A$  toks, kad  $d(x, B) = \varepsilon$ ,  $\varepsilon > 1$ . Tada  $d(A, B) \geq \varepsilon$ . Tuo pat metu, įvertinant tai, kad  $B \subset A$ , galima teigti, jog minimalus atstumas nuo bet kurio taško  $y \in B$  iki  $A$  lygus 0, t.y.  $d(B, A) = 0$ . Vadinasi  $d(A, B) \neq d(B, A)$ .

3. Tarkime, kad  $A, B \in H(X)$  ir  $A \neq B$ . Tada  $d(A, B) \neq 0$  arba  $d(B, A) \neq 0$ .

4. Jeigu  $A, B, C \in H(X)$ , tai  $(B \subset C) \Rightarrow (d(A, C) \leq d(A, B))$ .

Ši savybė įrodoma prieštaros būdu, remdamiesi pirma savybe. Tarkime, kad  $d(A, B) < d(A, C)$ . Tada yra taškas  $x \in A$  toks, kad  $d(x, C) > d(x, B)$ , su visais  $y \in A$ . Parinkę  $y = x$ , gauname prieštaravimą.

5.  $A, B, C \in H(X)$ , tai  $d(A \cup B, C) = \max\{d(A, C), d(B, C)\}$ . Pakanka pastebėti, jog

$$d(A \cup B, C) = \max_{x \in A \cup B} \{d(x, C)\} = \max\{\max_{x \in A} \{d(x, C)\}, \max_{x \in B} \{d(x, C)\}\} = \max\{d(A, C), d(B, C)\}$$

6. Jeigu  $A, B, C \in H(X)$ , tai  $d(A, B) \leq d(A, C) + d(C, B)$ .

Pakeiskime abi nelygybės puses atitinkamomis išraiškomis, būtent:  $d(A, B) = \max_{x \in A} \{\min_{y \in B} \{d(x, y)\}\}$  ir

$$d(A, C) + d(C, B) = \max_{x \in A} \{\min_{z \in C} \{d(x, z)\}\} + \max_{w \in C} \{\min_{y \in B} \{d(w, y)\}\}.$$

Kadangi maksimumas pagal  $w \in C$  yra nemažesnis už reikšmę, gaunamą pakeitus  $w$  bet kuria kita reikšme (tarkime  $z$ ), tai galime užrašyti

$$d(A, C) + d(C, B) \geq \max_{x \in A} \{\min_{z \in C} \{d(x, z)\}\} + \min_{y \in B} \{d(z, y)\} = \max_{x \in A} \{\min_{z \in C} \{d(x, z) + \min_{y \in B} \{d(z, y)\}\}\}.$$

**Apibrėžimas.** Tarkime, kad  $(X, d)$  yra pilnoji metrinė erdvė. Hausdorfo atstumas  $h$  tarp taškų  $A, B \in H(X)$  yra nusakomas lygybe

$$h(A, B) = \max\{d(A, B), d(B, A)\}.$$

Įrodysime, jog ką tik įvestas Hausdorfo atstumas  $h(A, B)$ ,  $A, B \in H(X)$ , tenkina visus metrikai keliamus reikalavimus, t.y. jį galima interpretuoti kaip metriką erdvėje  $H(X)$ . Imkime  $A, B, C \in H(X)$ .

Aišku, kad  $h(A, A) = \max\{d(A, A), d(A, A)\} = d(A, A) = \max_{x \in A} \{d(x, A)\} = 0$ ;

įvertinę  $A, B$  kompaktiškumą, užrašome:

$$h(A, B) = d(a, b), \text{ kur } a \in A, b \in B.$$

Taigi,  $0 \leq h(A, B) < \infty$ , su visais  $A, B \in H(X)$ . Jeigu  $A \neq B$ , tai galime teigti, jog yra taškas  $a \in A$  toks, kad  $a \notin B$ . Tada  $h(A, B) \geq d(A, B) > 0$ . Norėdami įrodyti, jog  $h(A, B) \leq h(A, C) + h(C, B)$ , pasinaudosime įrodyta šešta savybe, t.y.

$$d(A, B) \leq d(A, C) + d(C, B), \quad d(B, A) \leq d(B, C) + d(C, A),$$

su visais  $A, B, C \in H(X)$ . Vadinasi,

$$\begin{aligned} h(A, B) &= \max\{d(A, B), d(B, A)\} \leq \max\{d(A, C) + d(C, B), d(B, C) + d(C, A)\} \leq \\ &\leq \max\{d(A, C) + d(C, A)\} + \max\{d(B, C), d(C, B)\} = h(A, C) + h(C, B). \end{aligned}$$

**Apibrėžimas.** Erdvė  $(H(X), h)$  vadinama fraktalų erdve (arba tiesiog, fraktaline erdve).

Būtent šioje, fraktalinėje, erdvėje „gyvena“ fraktalai. Tad, kas tai yra fraktalas? Nepaisant ryškių pasiekimų matematikoje, kol kas nesiryžtama tiksliai įprasminti fraktalą, pateikti formalų fraktalo apibrėžimą. Fraktalai kol kas nusakomi paveikslukų įvairove bei turiniu, kurį tie paveikslukai talpina.

## 1.11. Fraktalinė dimensija

1977 m. B. Mandelbrotas šokiravo mokslinę visuomenę, įvesdamas trupmeninę dimensiją, tuo pačiu atitrūkdamas nuo topologinės (sveikąjo skaičiaus) dimensijos sampratos. Jo dimensija – tai parametras, charakterizuojantis, kaip tankiai fraktalas užima erdvę, naudojamas fraktalų sudėtingumo palyginimui.

Šiuo metu žinoma daug skaičių, charakterizuojančių fraktalus, ir skirtų pastarųjų palyginimui. Visi jie, dažniausiai, vadinami fraktalinėmis dimensijomis. Tai svarbios kiekybinės fraktalų charakteristikos dėl dviejų priežasčių: pirma, jas galima apibrėžti realaus pasaulio objektams, ir, antra, apytiksles jų reikšmes (įverčius) galima gauti eksperimentiškai.

Imkime pilną metrą erdvę  $(X, d)$ . Tarkime, kad  $A$  yra netuščiasis kompaktinis erdvės  $X$  poaibis, t.y.  $A \in H(X)$ . Tarkime, kad  $\varepsilon > 0$ , o  $N(A, \varepsilon)$  žymi mažiausią uždarytųjų rutulių  $B(x, \varepsilon) = \{y \in X \mid d(x, y) \leq \varepsilon\}$ , sudarančių baigtinį aibės  $A$  denginį, skaičių, t.y.  $N(A, \varepsilon)$  yra mažiausias sveikasis skaičius  $M$  toks, kad  $A \subset \bigcup_{n=1}^M B(x_n, \varepsilon)$  (čia  $\{x_1, x_2, \dots, x_M\} \subset X$ ). Tokio skaičiaus  $M$  egzistavimas išplaukia iš to, jog  $A$  yra kompaktinis poaibis: iš tikrųjų, kiekviena taškui  $x \in A$  parinkę atvirąjį (spindulio  $\varepsilon > 0$ ) rutulį, turėsime atvirąjį  $A$  denginį, iš kurio galėsime išrinkti baigtinį  $A$  podenginį, susidedantį, tarkime, iš  $\tilde{M}$  atvirųjų rutulių. Kiekvieną atvirąjį rutulį keisdami uždaruju, gausime  $A$  denginį, sudarytą iš  $\tilde{M}$  uždarytųjų rutulių.

Dabar, aibę denginių, susidedančių iš ne daugiau kaip  $\tilde{M}$  uždaryjū (spindulio  $\varepsilon > 0$ ) rutulių, pažymėkime  $\mathcal{D}$ . Aišku, aibė  $\mathcal{D}$  turės bent vieną elementą. Apibrėžkime funkciją  $f: \mathbf{D} \rightarrow \{1, 2, \dots, \tilde{M}\}$  taip:  $f(d) = \{\text{rutulių, sudarančių denginį } d \in \mathbf{D}, \text{ skaičius}\}$ . Tada, teigiamų skaičių aibė  $\{f(d) \mid d \in \mathbf{D}\}$  bus baigtinė. Vadinasi, į ją įeis ir mažiausias skaičius  $N(A, \varepsilon)$ .

Intuityvi idėja, kuria remiantis įvedama fraktalinės dimensijos sąvoka, yra tokia: aibė  $A$  turi fraktalinę dimensiją  $D$ , jeigu

$$N(A, \varepsilon) \approx C\varepsilon^{-D};$$

čia  $C$  yra tam tikra konstanta; simbolio „ $\approx$ “ panaudojimą suprantame taip: jeigu  $f(\varepsilon)$  ir  $g(\varepsilon)$  yra dvi realiosios funkcijos, tai „lygybė“  $f(\varepsilon) \approx g(\varepsilon)$  reiškia, jog

$$\lim_{\varepsilon \rightarrow 0} \frac{\ln f(\varepsilon)}{\ln g(\varepsilon)} = 1.$$

Išsprendę ankstesniąją „lygybę“  $D$  atžvilgiu, gauname:

$$D \approx \frac{\ln N(A, \varepsilon) - \ln C}{\ln(1/\varepsilon)}.$$

Pastebėsime, kad  $\frac{\ln C}{\ln(1/\varepsilon)}$  artėja prie nulio, kai  $\varepsilon \rightarrow 0$ .

**Apibrėžimas.** Sakykime, kad  $A \in H(X)$  ir  $(X, d)$  yra metrinė erdvė. Tarkime, kad  $N(A, \varepsilon)$  žymi mažiausią sveikąjį uždaryjū (spindulio  $\varepsilon$ ) rutulių, sudarančių aibės  $A$  denginį, skaičių. Jeigu riba

$$D = \lim_{\varepsilon \rightarrow 0} \frac{\ln(N(A, \varepsilon))}{\ln(1/\varepsilon)}$$

egzistuoja, tai skaičius  $D$  vadinamas fraktaline aibės  $A$  dimensija; sakysime, jog „ $A$  turi fraktalinę dimensiją  $D$ “, ir žymėsime  $D(A) = D$ .

Panagrinėsime metodus skirtus panašių taškų telkinių struktūrų dydžio nustatymui. Nagrinėkime kvadratinį tinklelį (langelio dydis  $\varepsilon$ ), uždėtą ant stebimos taškinės struktūros. Kiekvienoje tinklelio dalyje suskaičiuojamas į ją papuolusių taškų skaičius  $n_i$ . Jis dalinamas iš  $N$  – bendro taškų skaičiaus:

$$P_i(\varepsilon) = \frac{n_i}{N}.$$

Tuomet bendru atveju fraktalinė dimensija apskaičiuojama pagal formulę:

$$D_q = \frac{1}{q-1} \frac{\log \sum_i (P_i(\varepsilon))^q}{\log \varepsilon}.$$

Kai  $q=0$ , turime užimtumo dimensiją,  $q=2$  – koreliacijos dimensiją. Objektai, kurių užimtumo dimensija skiriasi nuo jų topologinės dimensijos, vadinami fraktalais.

Apibrėžkime informacinę funkciją:

$$I \equiv - \sum_{i=1}^{N_{\varepsilon}} P_i(\varepsilon) \log[P_i(\varepsilon)],$$

kur  $N_{\varepsilon}$  – skaičius užimtų langelių.

Tuomet informacinė dimensija apibrėžiama taip

$$d_{\text{inf}} \equiv - \lim_{\varepsilon \rightarrow 0} \frac{I}{\log(\varepsilon)} = \lim_{\varepsilon \rightarrow 0} \frac{\sum_{i=1}^{N_{\varepsilon}} P_i(\varepsilon) \log[P_i(\varepsilon)]}{\log(\varepsilon)}.$$

Pakeitus  $I = \log N_{\varepsilon}$  – gausime užimtumo dimensiją, o  $I = \log \sum_i (P_i(\varepsilon))^2$  – koreliacijos dimensija.

## 1.12. Pagrindiniai parametrai

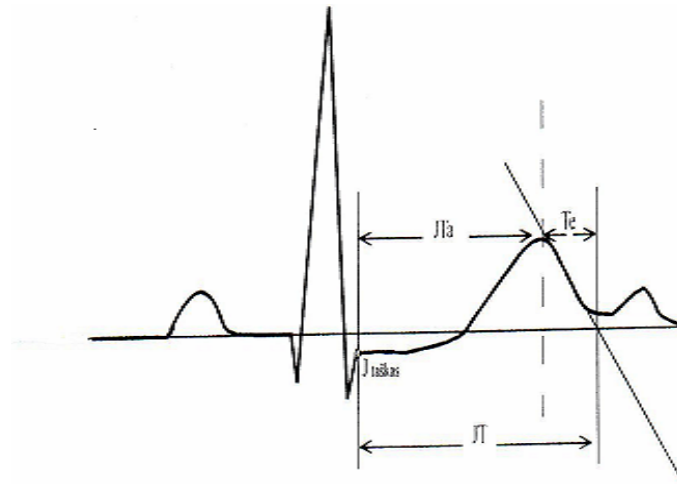
**ŠSD** – širdies susitraukimų dažnis. Parametro kitimo ribos 20 – 250 kartų per minutę. Fizinio krūvio metu ŠSD didėja todėl, kad intensyviu širdies darbu, nes būtina patenkinti dirbančių raumenų poreikius. Po fizinio krūvio ŠSD ne iš karto grįžta į pradinį lygį. Atlikus mažo intensyvumo fizinį krūvį, ŠSD greitai mažėja iki lygio, buvusio ramybės būsenoje. Atliekant didelio intensyvumo fizinius pratimus, ŠSD pokyčius galima suskirstyti į dvi fazes: greito eksponentiško mažėjimo ir lėto mažėjimo iki lygio, buvusio ramybės būsenoje. Manoma, kad ilgalaikių treniruočių metu ŠSD mažėja dėl kelių veiksnių. Tai priklauso nuo pratimų intensyvumo, jų apimties, trukmės, pertraukų tarp pratimų bei pratime dalyvaujančių raumenų kiekio. Kol kas mažai išnagrinėti ŠSD pokyčių fiziologiniai mechanizmai, taikant ilgalaikes fizines treniruotes.

**S** – sistolinis arterinio kraujo spaudimas. Parametro kitimo ribos 80 – 250 mmHg. Fizinio krūvio metu padidėjus į širdį pritekancio veninio kraujo tūriui, padidėja galinis diastolinis spaudimas skilvelyje, o taip pat sistolės metu išstumiamo kraujo kiekis. Miokardo susitraukimo galimybės ir spaudimas aortoje apsprendžia kraujo išstūmimo jėgą ir periferijos pasipriešinimą, kurį kraujas, tekėdamas arterijomis, turi nugalėti. Šie veiksniai turi įtakos sistolinio tūrio pokyčiams fizinio krūvio metu. Netreniruotų žmonių sistolinis kraujo tūris krūvio metu siekia 100 – 120 ml, sportuojančių žmonių padidėja iki 160 – 200 ml. Didėjant fizinio krūvio intensyvumui minutinis širdies tūris didėja ir dėl ŠSD. Treniruotos širdies raumuo pasižymi didelėmis susitraukimo galimybėmis, todėl sistolinis tūris didėja iki maksimalių žmogaus vystomų galingumų. Manoma, jog šis fiziologinis mechanizmas yra vienas iš svarbiausių sveikatą stiprinančių pratimų efektų. Atlikta daug tyrimų, patvirtinančių ilgalaikių fizinių treniruočių įtaką sistolinio tūrio didėjimui tiek ramybės būsenoje, tiek fizinio krūvio metu.



**D** – diastolinis arterinio kraujo spaudimas. Parametro kitimo ribos – 20 – 130 mmHg. Fizinio krūvio metu diastolinis arterinio kraujo spaudimas mažėja (tai įtakoja periferijos pasipriešinimas).

**JT** – intervalas elektrokardiogramoje nuo jungties taško J iki T bangos pabaigos (JT intervalas). Parametro kitimo ribos 0,14 – 0,36 s. Fizinio krūvio metu šis intervalas trumpėja. JT intervalas apibūdina skilvelių repoliarizacijos trukmę. JT intervalas lemia skirtingus elektrofiziologinius reiškinius ir yra skirstomas į JTa intervalą (nuo taško J iki T viršūnės) bei Te intervalą (nuo T bangos viršūnės iki T pabaigos) ( 1.1 pav. ). JT intervalo pokyčiams įtakos turi reguliacinė nervų sistema. Yra žinoma, kad organizmo metaboliniai pokyčiai yra susiję su repoliarizacijos pokyčiais. Derivacijos, kuriose JT intervalas trumpesnis, rodo, kad tose miokardo zonose vyksta ankstyvesnė repoliarizacija, greitesni metaboliniai pokyčiai, o ten, kur JT intervalas ilgesnis, rodo lėtesnę repoliarizaciją bei lėtesnes metaboliškas reakcijas. Nustatyta, kad JT intervalas sveikiems žmonėms fizinio krūvio metu trumpėja iki 140 ms, tuo tarpu išemine širdies liga sergantiems JT intervalas krūvio metu sutrumpėja žymiai mažiau.



1.2 pav. JT intervalo grafinis vaizdas

**N** – galingumas, matuojamas W.

**N<sub>max</sub>** – tyrimo metu pasiektas maksimalus galingumas, W.

**RR** – laiko intervalas tarp dviejų širdies susitraukimų. Intervalas apskaičiuojamas tokiu būdu (indeksas *i* nurodo, kad parametrai skaičiuojami esant tam pačiam galingumui N ):

$$RR_i = \frac{60}{\check{SSD}_i} \quad (1.12)$$

**JT/RR** – parametras, siejantis reguliacinę ir aprūpinančiąją sistemas ir yra santykis (indeksas *i* reiškia, kad parametrai gaunami esant tam pačiam galingumui N ):

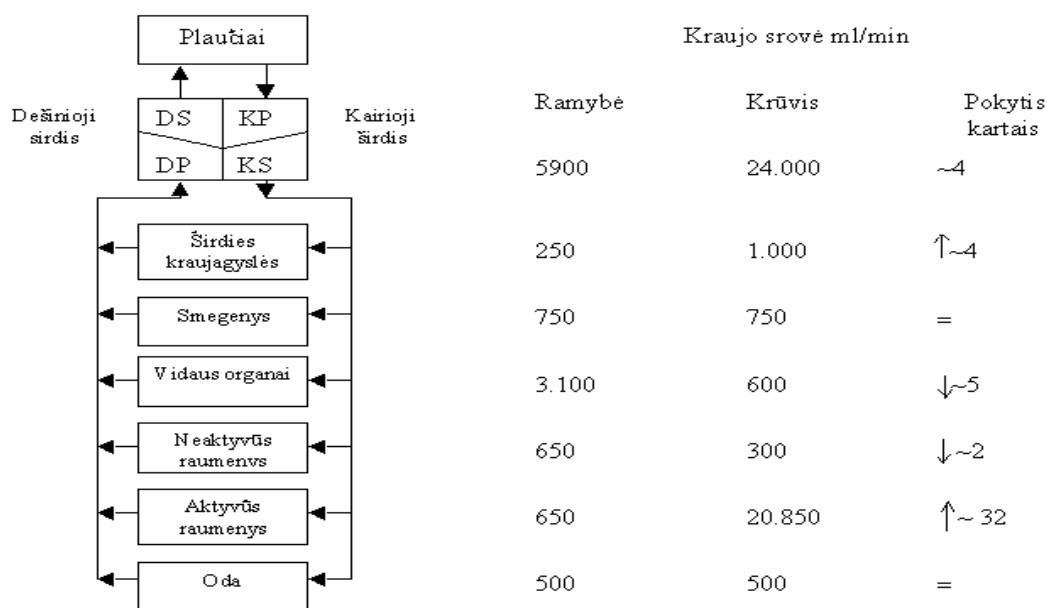
$$JT / RR_i = \frac{JT_i}{RR_i} \quad (1.13)$$

Ilgalaikių fizinių treniruočių metu sumažėja ŠSD bei sutrumpėja JT intervalas, t.y. širdžiai tenka mažesnis krūvis. Taigi santykis JT/RR mažėja. Vyrų širdys apytikriai trečdaliu yra didesnės už moterų, jų šis santykis mažesnis negu moterų. Kokios fiziologinės ypatybės lemia šias savybes, nėra žinoma.

### 1.13. Amžiaus įtaka širdies ir kraujagyslių sistemos reguliavimui

Augant organizmui, pirmuosius 10 – 15 metų, kol dar neišsivystė kraujo srovės organizme persikirstymo mechanizmas, pagrindinė svarba, didinant širdies darbingumą fizinio aktyvumo metu, tenka širdies susitraukimų dažniui. Tuo galime įsitikinti nagrinėdami fiziologinio maksimalaus širdies dažnio priklausomybę nuo amžiaus: kuo žmogus jaunesnis, tuo maksimalus širdies susitraukimų dažnis didesnis. Visiškai susiformavus organizmui, padidėjus širdies susitraukimo jėgai, didėja išstumiamo į aortą kraujo kiekis – sistolinis tūris. Todėl subrendus organizmui, pagrindinį širdies ir kraujagyslių sistemos darbingumo didėjimą apibūdina suminio vertinimo kitimas. Vyresniems žmonėms, atsiradus pokyčiams kraujagyslėse, mažėja diastolinė funkcija (kaip kompensacija didėja arterinio kraujo spaudimas). Jau vien šie pokyčiai lemia, jog fizinis aktyvumas skirtingais amžiaus etapais aktyvina skirtingus fiziologinius adaptacinius mechanizmus, t.y. skirtingus jų parametrus. Taigi amžius, lytis, įvairios patologijos lemia individualaus fizinio aktyvumo būtinumą.

Panagrinėkime bendruosius dėsningumus, subrendusiam organizmui reaguojant į fizinį aktyvumą (1.3 pav).



1.3 pav. Kraujo tėkmės (ml/min) persiskirstymas įvairiuose organuose ramybės ir fizinio krūvio metu

Pastebime, kad krūvio metu vyksta pokyčiai visame organizme, ne tik aktyviuose raumenyse. Visas organizmas prisitaiko prie kintančio krūvio. Skirtingose organų sistemose šie pokyčiai priešingų kryptių. Labiausia stengiamasi padidinti kraujotaką per aktyvią raumenų grupę bei tą darbą lemiančių organų sistemą (širdį, plaučius). Per organus, kurie labai svarbūs skatinant atlikti krūvį, stengiamasi išlaikyti stabilią kraujotaką (pvz., smegenis, kurios lemia visos sistemos reguliaciją, ir per odą, kuri reguliuoja šilumos apykaitą tarp organizmo ir aplinkos). Kitose organų sistemose tekančio kraujo kiekis labai sumažėja. Visi šie pokyčiai ir sudaro hemodinamikos persikirstymo organizmą, kurio pagrindinė paskirtis atlikti organizmo adaptaciją. Kuo didesnius pokyčius (aišku, neperžengdami tam tikrų fiziologinių pokyčių ribų) mes galime sukelti organizme, tuo prie didesnio krūvio organizmas gali prisitaikyti, t.y. organizmas bus veiklesnis, pajėgesnis.

#### **1.14. Širdies ir kraujagyslių sistemos funkcinį rodmenų kaita, atliekant fizinio krūvio mėginis**

Organizmo adaptavimasis ilgalaikėms fizinėms treniruotėms bei adaptavimosi proceso kontrolė yra aktualus fiziologijos ir sporto medicinos uždavinys. Šio uždavinio sprendimas sietinas su organizmo funkcinės būklės įvertinimu.

Organizmo širdies ir kraujagyslių sistemos funkcinės būklės įvertinimui atliekami fizinio krūvio mėginiai. Fiziologijos, klinikinės medicinos bei sporto mokslo žurnaluose galima rasti daug dabar taikomų žmogaus organizmo funkcinio pajėgumo bei adaptavimosi pokyčių vertinimo protokolų.

Iš gausybės vertinimo kriterijų galima išskirti tris grupes rodiklių, nusakančių organizmo darbingumą ar kitų parametru pokyčius bei nurodančių teigiamą ilgalaikių fizinių treniruočių poveikį sportuojančio žmogaus organizmui. Pirmajai – maksimalių galimybių rodiklių grupei priskiriami fizinio darbingumo ir funkcinio organizmo sistemų maksimalaus aktyvumo laipsnio įvertinimo parametrai (maksimalus išvystytas galingumas, maksimali fizinio krūvio trukmė, maksimalus organizmo funkcijų aktyvumo laipsnis). Kokybiniais priskiriami rodikliai, nurodantys organizmo funkcijų aktyvumo laipsnio pokyčius, atliekant tam tikro intensyvumo fizinius krūvius; kiekybiniais rodikliams – santykiniai rodikliai, parodantys biologinę atliekamo fizinio krūvio vertę. Dabar klinikinėje praktikoje naudojami organizmo funkcinę būklę įvertinantys parametrai ir indeksai ne visai įvertina sisteminių atsaką į fizinį krūvį. Vieni jų atspindi bei rodo kvėpavimo sistemos funkciją, kiti – širdies ir kraujagyslių sistemos atsaką, tretieji apibendrina organizmo pajėgumą bei reakciją į fizinį krūvį, neįvertindami atskirų sistemų veiklos.

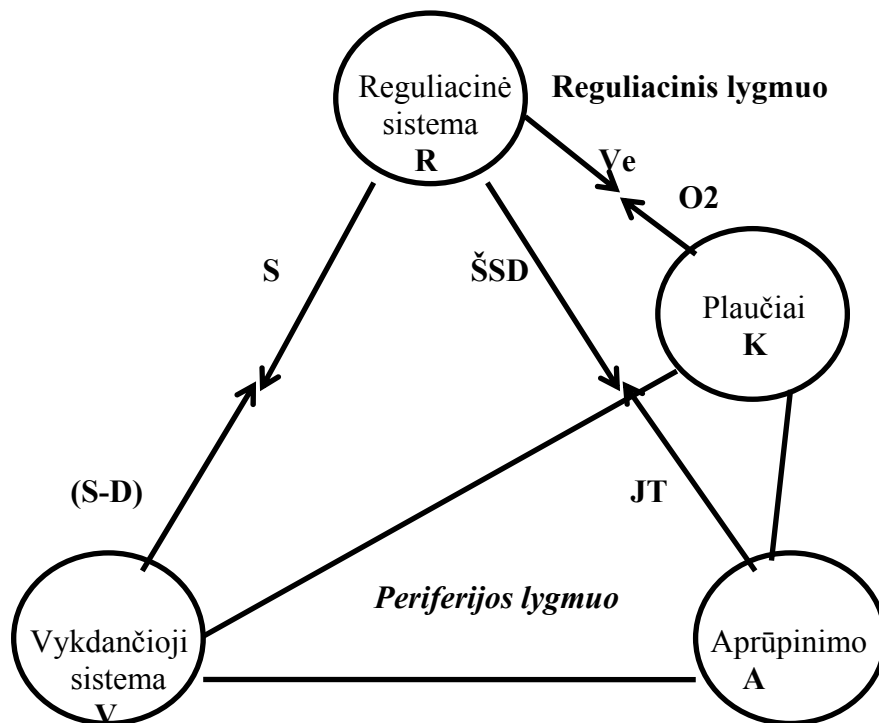
Atliekant pratimą ilgiau nei keletą sekundžių, labai svarbu, kad į skeleto raumenis būtų tiekiamas adekvatus krūviui deguonies ir energetinių medžiagų kiekis. Judėjimo ir atramos sistema

fizinio krūvio metu adekvačiai aprūpinima tada, kai suaktyvėja aprūpinimo sistemų veikla, o tai apsprendžia reguliuojančios sistemos. Atliekant šį tyrimą organizmo reakcijai į fizinį krūvį įvertinti išskirtos anksčiau minėtos sistemos: vykdančioji sistema (dirbantis raumenynas), reguliacinė, apimanti centrinę nervų sistemą, autonominių bei humoralinį valdymą ir aprūpinančioji – širdies ir kraujagyslių sistema. Lietuvoje sukurtas integralios organizmo reakcijos į fizinį krūvį modelis teikia naujas analizės galimybes. Naudojantis šiuo modeliu galima įvertinti minėtų sistemų veiklą tiriamajam atliekant fizinio krūvio mėginį. Fizinio krūvio metu vyksta kraujotakos perskirstymas, kas leidžia organizmui adekvačiau atlikti fizinį krūvį, t.y. padėti pagrindinėms organizmo sistemoms adaptuotis fiziniam krūviui. Juo didesni pokyčiai, neviršijant tam tikrų fiziologinių ribų, atsiranda organizme minėtų trijų sistemų atžvilgiu, tuo didesni fizinį krūvį organizmas gali atlikti. To pasekoje organizmo pagrindinės sistemos geriau adaptuojasi bei išvysto didesnę pajėgumą.

Organizmui išvysčius tam tikrą pajėgumą, jame vyksta kompleksinė reakcija į fizinį krūvį, kurią apsprendžia minėti elementai. Remiantis žmogaus fiziologija galima išskirti šiuos pagrindinius elementus, apsprendžiančius pagrindinius pokyčius organizme krūvio metu:

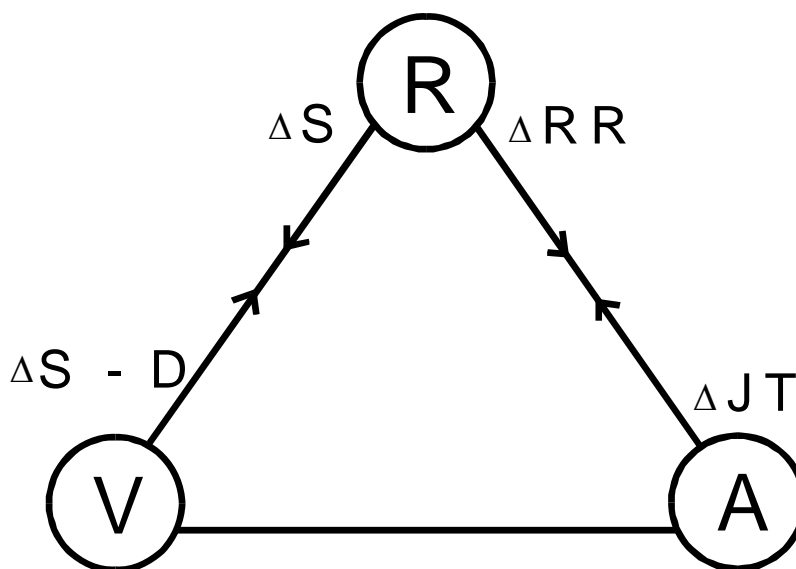
- 1) reguliacinė sistema (R) - smegenys visuose reguliaciniuose lygmenyse;
- 2) (V) vykdančioji sistema - fizinio krūvio metu dirbantis raumenys;
- 3) širdies ir kraujagyslių – aprūpinančioji sistema (A), užtikrinanti adekvačią kraujotaką;
- 4) kvėpavimo sistema (K), kuri, priklausomai nuo pratekančio per plaučius kraujo kiekio, absorbuoja deguonį bei išskiria anglies dvideginį.

Šis, keturias sistemas apibendrinantis modelis schematiškai pavaizduotas 1.4 paveiksle.



1.4 pav. Žmogaus organizmo funkcinės būklės įvertinimo modelis krūvio metu

Santykius tarp šių elementų ir jų funkcijų, galima nusakyti daugeliu parametru. Modelyje nurodyti ir nežalojančiu būdu nustatomi, elementų funkciją atspindintys dydžiai. Šį modelį iš dalies galima supaprastinti jungiant kvėpavimo ir širdies bei kraujagyslių sistemas į vieną aprūpinančią (A) sistemą. Per dešiniąją širdį, plaučius ir kairiąją širdį kraujas teka nuosekliai, nėra nei nuotėkio (praradimo), nei pritekėjimo (papildymo), todėl hemodinamiką šioje organizmo dalyje lemia širdies funkcija ir kvėpavimo funkcija (t. y. absorbuoto deguonies kiekis) nuo jos visiškai priklauso. Taigi galima modelį supaprastinti 1.5 pav.



**1.5 pav. Integralinio vertinimo modelis ir organizmo sistemų tarpusavio ryšius rodantys žymenys (A. Vainoras, 1996), V – vykdančioji sistema, R – reguliacinė sistema, A – aprūpinančioji sistema**

Išskiriami trys funkciniai modelio elementai: 1) vykdančioji sistema (V), fizinio aktyvumo metu veikianti raumenų grupė; 2) reguliacinė sistema (R), apimanti CNS, autonominio bei humoralinio valdymo elementus; 3) širdies ir kraujagyslių - aprūpinančioji sistema (A), atsakinga už centrinę hemodinamiką. Santykių tarp šių elementų galima nusakyti daugeliu parametru, tačiau šiam tyrimui buvo pasirinkti paprasčiausi ir lengviausiai nustatomi. Veikiančią raumenų sistemą vertiname pasiektu galingumu  $N$ , reguliacinę sistemą - intervalu  $RR$  (laikas tarp dviejų širdies susitraukimų), o širdies veiklos suintensyvėjimą –  $JT$  intervalu (intervalas elektrokardiogramoje nuo jungties  $J$  iki  $T$  bangos pabaigos). Organizmo adaptacija fiziniam krūviui vertinta remiantis pasirinktų parametru pokyčiais. Normalizuotų pokyčių suma vertinta kaip nuotolis Euklido erdvėje ir sudaro suminį vertinimą ( $Sv$ ). Jo dydis susijęs su tiriamojo funkcinė būkle: sportininkams jis sudaro 70-100 proc., sveikiems tiramiesiems 60-70 proc., o turintiems patologiją, šis dydis mažėja. Atliekant šį vertinimą, reikšmingas ne tik pats dydis, bet ir jo kitimas bei jį sudarančių parametru tarpusavio santykis. Jo pokyčiai rodo organizmo reguliacinių procesų pokyčius, kurie dar nesukelia patologinių reiškinių, bet

rodo, jog yra jų tikimybė. Šis modelis pagal parametrų pokyčius bei jų santykį padeda įvertinti organizmo atsaką į įvairius poveikius. (13, 14)

### **1.15. Fizinio krūvio įtaka širdies ir kraujagyslių sistemos funkcijai**

Širdies ir kraujagyslių sistema atlieka vieną iš svarbiausių vaidmenų organizmo adaptacijos į įvairius fizinius pratimus mechanizmų grandinėje. Mokslinėje literatūroje atkreipiamas dėmesys į centrinių ir periferinių reakcijų sąsają. Fiziniai krūviai, priklausomai nuo metabolinės dirbančių raumenų apykaitos, sukelia ryškius širdies ir kraujagyslių sistemos veiklos pokyčius.

Širdžiai dirbant ilgą laiką ir daug kartų miokarde siunčiamas tam tikras signalas, kuris veikia ląstelių genetinį aparatą, skatina nukleino rūgščių ir baltymų sintezę. Tai lemia širdies raumens struktūrinius pokyčius, kurie sudaro ilgalaikės adaptacijos fiziniams krūviams pagrindą. Fizinio krūvio įtakoje dėl širdies raumens hipertrofijos bei sienelių sustorėjimo didėja širdies svoris. Krūvio metu hipertrofuotam miokardui reikia didesnio deguonies kiekio. Ilgai buvo manoma, kad kairiojo skilvelio hipertrofija yra vienintelė adaptacinė reakcija fiziniams krūviams. Širdis didėja didėjant jos ertmių tūriui, storėjant skilvelių sienelėms ir tarpkilvelinei pertvarai. Širdies raumuo hipertrofuojasi dėl pristatyto į jį didesnio antinksčių hormono noradrenalino kiekio, taip pat skatinama baltymų sintezė. Įvertinta reguliarių jėgą ugdančių fizinių pratimų įtaka moterų ir vyrų širdies kairiojo skilvelio masei. Širdies kairiojo skilvelio hipertrofija pasireiškė ir moterims, bet, lyginant su vyrais, jų rodikliai buvo gerokai mažesni. Kapiliarų tinklo augimas yra proporcingas miokardo skaidulų didėjimui.

Širdis ir kraujagyslės sudaro kompleksinę organų sistemą, kuri organizme apsprendžia autonominių jų reguliavimą. Priklausomai nuo viso organizmo bei atskirų organų poreikių tai kontroliuoja nervų sistema. Informacija ateina iš spaudimui, tūriui bei cheminiams veiksniams jautrių receptorių, esančių širdies kraujagyslių sistemoje. Ši informacija integruojama smegenų kamieno centruose, koordinuojančiuose kraujo spaudimo, kraujagyslių pasipriešinimo bei širdies sistolinio tūrio ir jos prisipildymo lygmenį per eferentinius nervų sistemos bei humoralinius ryšius. Refleksinis širdies ir kraujagyslių sistemos reguliavimo lygis apsprendžia kraujo perskirstymą kintant kraujotakai ir palaiko pastovų perfuzinį spaudimą.

### **1.16. Raumenų kraujotaka**

Fizinio krūvio metu dirbančių raumenų kraujotaka yra svarbus veiksnys, lemiantis jų darbingumą. Nuo raumenų kraujotakos pokyčių priklauso deguonies vartojimo intensyvumas. Tai stebima tiek fizinio krūvio pradžioje, tiek fizinio krūvio, tiek ir proceso normalizavimosi metu. Raumenų kraujotaka reguliuojama pagal dirbančių raumenų poreikius ir tam sutelkti visi kraujotakos

reguliavimo mechanizmai. Didžiausią kraujotakos intensyvumą pavyksta užregistruoti atliekant fizinį pratimą tam tikrai raumenų grupei, mažesnį, kai į darbą įsitraukia kelios raumenų grupės. Taip yra dėl to, kad širdies darbinio tūrio limitas neleidžia periferiniams kraujotakos organams dirbti maksimaliu intensyvumu.

Pagrindiniu ir tiesioginiu širdies darbo efektyvumo rodikliu yra laikomas minutinis širdies tūris. Ramybės sąlygomis vienos sistolės metu išmetant apie 70 ml kraujo ir esant 60-72 širdies susitraukimams per minutę, širdis perpumpuoja apie 5 l kraujo. Fizinio krūvio metu, dažnėjant pulsui ir didėjant sistoliniam tūriui, širdis savo darbą gali padidinti penkis, šešis kartus, t.y. per minutę perpumpuoti 25 – 30 litrus kraujo. Širdies perpumpuojamo kraujo nebeužtenka, jeigu darbe dalyvauja daug raumenų grupių.

Kraujotakos persiskirstymo mechanizmai reguliuoja kraujo srovės pasiskirstymą tokiu principu, kad daugiausia kraujo tekėtų į dirbančius raumenis. Adaptacija fiziniam krūviui jungia tris labai svarbias grandis. Pirma, tekantis kraujas persiskirsto dirbančio raumens viduje. Antra, padidėja kapiliarų kiekis. Trečia, padidėja mitochondrijų tankis raumens ląstelėje. Autorius teigimu, būtent šie trys veiksniai lemia padidėjusį raumens sugebėjimą suvartoti didesnę deguonies kiekį ir padidinti toleranciją fiziniams krūviams .

Raumenų pompos mechanizmo dėka veninis kraujas lengviau grįžta į širdį. Šie raumenų susitraukimo jėgos sukelti efektai didina griaučių raumenų perfuziją. Raumeniui susitraukiant, kraujas išstumiamas iš kraujagyslių, tai palengvina veninio kraujo grįžimą į širdį ir didina skeleto raumenų perfuziją. Raumenų pompos mechanizmas veikia tik ritmiškų susitraukimų metu – venos užpildomos krauju raumens atsipalaidavimo metu. Raumenų atsipalaidavimo metu spaudimas venulėse ir giliose mažosiose venose yra neigiamas. Kadangi mažųjų venų sienelės ribojasi su aplinkiniais raumenų audiniais, neigiamas spaudimas atveria venų vožtuvus raumenų atsipalaidavimo metu. Sumažėjęs slėgis šiose venose ir mažėjimo laikas priklauso nuo to, kaip greitai teka kraujas iš arterijų per kapiliarus į tuščius venų segmentus.

Raumenų pompa lengvina kraujotaką mažiausiai dviem būdais. Pirma, sumažėja venų spaudimas, antra, padidėja bendroji sistemos kinetinė energija, t.y. padidėja slėgis distaliau esančiose raumenų venose lyginant su proksimaliųjų venų slėgiu, padedančiu varyti kraują per raumenų kraujagysles. Raumenų pompos efektyvumui turi įtakos griaučių raumenų tipas, skaidulų išsidėstymas raumenyse ar raumenų grupėse bei kraujagyslių tinklo struktūra raumenyse, veninių vožtuvų funkcija, gravitacijos efektai ir venas užpildantis slėgis. Žmogaus raumenų pompos efektyvumo įvertinimai rodo, kad nuo 30 iki 60 proc. kraujo perpumpavimo tenka raumenų pompos mechanizmui. Nustatyta, kad lokomotoriniai judesiai (bėgimas, važiavimas dviračiu) bei šių pratimų intensyvumas irgi įtakoja raumenų pompos efektyvumą.

## 1.17. Išeminė širdies liga

Širdies ir kraujagyslių ligos yra viena svarbiausių ne tik medicinos, bet ir socialinių problemų. Širdies ir kraujagyslių ligos sudaro apie pusę visų mirčių Lietuvoje, 1/3 invalidumo bei nulemia 15 – 20 proc. apsilankymų sveikatos priežiūros įstaigose. 1999 metais Lietuvoje nuo širdies ir kraujagyslių ligų mirė 21 903 žmonės, arba 592,0/100 000 gyventojų.

Išeminė širdies liga yra labiausiai paplitusi širdies ir kraujagyslių sistemos liga. Ji yra viena dažniausia civilizuoatų šalių gyventojų mirties priežastis. Užkirsti kelią išeminei širdies ligai sunku, nes jos pradžia beveik nepastebima. Pirmieji ligos požymiai pasireiškia gana vėlai, kada liga yra įsigalėjusi, širdis išsekusi. Todėl gydytojui dažnokai tenka pradėti gydyti toli užleistą ligą. Pastarųjų dešimtmečių mokslininkų duomenimis laiku pradėjus gydyti sergantįjį, išeminę širdies ligą, galima sustabdyti. Labai svarbu, kad ligonis žinotų, kaip dirba jo širdis, kas trukdo jai plakti, kaip galima apsisaugoti, kaip reikia gyventi, kad ilgai galėtų būti aktyvus ir darbingas.

Širdis yra žmogaus kumščio dydžio ir sveria apie 300 – 400 gramų. Sergančiojo žmogaus širdis gali būti ir didesnė. Širdis – tai raumeninis siurblys. Vieną kartą susitraukdama širdis perpumpuoja apie 70 – 80 mililitrų kraujo, o per parą, susitraukdama iki 100 000 kartų širdis perpumpuoja apie 10 tonų kraujo.

Kas tai yra išeminė širdies liga? Išemija – tai dėl kokių nors priežasčių pablogėjusi vainikinių širdies kraujagyslių kraujotaka ir dėl to sutrikęs širdies raumens aprūpinimas krauju ir maisto medžiagomis. Dažniausia jos priežastis – širdies vainikinių kraujagyslių aterosklerozė, kai susidariusios plokštelės užkemša kraujagyslių spindį. Tačiau širdies raumens išemiją gali sukelti ir širdies vainikinių kraujagyslių trombai, spazmai, plaučių, kraujo ir kitos ligos. Skiriamos trys pagrindinės išeminės širdies ligos formos:

- krūtinės angina,
- širdies infarktas,
- staigi mirtis.

Nepakankamai krauju aprūpinimas širdies raumuo pakinta, todėl dažnai pradeda skaudėti širdies plote. Tai vadinama krūtinės angina. Kartais sutrinka širdies ritmas, pasireiškia širdies nepakankamumo požymiai: trūksta oro, dūstama, tinsta kojos. Iš pradžių šie reiškiniai pasireiškia fizinio krūvio metu, kai širdžiai pasidaro sunkiau dirbti, o vėliau, kai dar labiau susiaurėja kraujagyslės, ir ramybės būsenoje. Negaudamos kraujo, širdies raumens ląstelės žūva, prasideda infarktas. Sveikstant, per du mėnesius, infarkto vietoje susidaro randas iš jungiamojo audinio. Dėl rando širdies raumenyje gali susilpnėti širdies susitraukimai, sutrikti jos ritmas.



Staigi mirtis ištinka tada, kai širdies išemija prasideda ūmiai, ir širdis nespėja prisitaikyti prie pakitusių sąlygų, tada pradeda virpėti širdies skilveliai arba širdis sustoja.

Kokios išeminės širdies ligos priežastys? Nustatyta, kad išeminę širdies ligą sukelia ne vienas, o daugelis rizikos veiksnių. Svarbiausi išeminės širdies ligos rizikos faktoriai yra:

- netinkama mityba,
- rūkymas,
- nepakankamas fizinis aktyvumas,
- per didelė kūno masė,
- padidėjęs arterinis kraujospūdis,
- žmogaus psichinė būseną ir patiriami stresai,
- žalingi aplinkos faktoriai, tokie kaip oro užterštumas, įvairūs chemikalai ir t.t.,
- genetinis faktorius.

Ypač nepalankiai veikia kelių rizikos faktorių derinys. Jei vienu metu veikia 2 ar 3 rizikos faktoriai, tikimybė susirgti išemine širdies liga yra dešimtis kartų didesnė. Nepašalinus rizikos faktorių ir gydant tik vaistais, neužkirsime kelio išeminei širdies ligai.

Kokie tyrimai atliekami? Vienas iš pagrindinių tyrimų yra elektrokardiograma (EKG). Esant krūtinės anginai EKG gali būti požymiai, rodantys, kad širdis per mažai aprūpinama krauju ir tuo pačiu deguonimi - vadinamoji išemija. Tačiau visiškoje ramybėje EKG gali būti ir normali. Todėl dažnai tikslinga atlikti vadinamąjį krūvio mėginį – tai arba važiavimas medicininio dviračiu (veloergometrija), arba ėjimas specialiu judančiu takeliu (tredmilas).

Būtent remiantis krūvio mėginio rezultatais sprendžiama dėl diagnozės ir kaip toliau tyrinėti. Jei pacientas pakelia didelį krūvį, neatsiranda EKG išeminių pakitimų, normaliai kyla kraujospūdis ir pulsas, tada dažniausiai siūloma gydytis vaistais (silpnais) ar ieškoti kitos skausmų krūtinėje priežasties ir, gal būt, po kiek laiko krūvio mėginį reiktų pakartoti. Jei pacientas pakelia tik vidutinį ar mažą fizinį krūvį, bet EKG neatsiranda išemijos požymių, skausmo krūtinėje krūvio metu, o būna tik silpnumas, kojų nuovargis, oro stoka, galvos svaigimas, per greitai pakyla kraujospūdis ar pulsas – rekomenduojama gydytis vaistais ir po kiek laiko (savaičių ar mėnesių) tyrimą pakartoti. Jei pacientas pakelia tik mažą fizinį krūvį ar atsiranda skausmas krūtinėje ir/ar elektrokardiogramos pakitimai, tada svarstoma apie širdies zondavimo tikslingumą ir dažniausiai pacientui siūloma šį labai svarbų tyrimą atlikti (15, 16).

## 1.18. Programavimo kalba

C++ Builder – tai objektiškai orientuotų programų kūrimo sistema, kurios priemonėmis galima paruošti įvairaus lygio ir paskirties programas: tiek mėgėjiškas, tiek profesionalias. Programos C++ Builder terpėje kuriamos, įtraukiant į programą reikalingus objektus (komponentus), suteikiant jiems reikalingas savybes ir programuojant C++ kalba reakcijas į įvykius, kurie gali įvykti programos vykdymo metu. Visų tų dalių sujungimą į vientisą C++ kalbos sintaksę atitinkančią programą automatiškai atlieka C++ Builder sistema.

## 2. TIRIAMOJI DALIS

### 2.1. Duomenys

Nagrinėjome 592 diagnostinius tyrimus, gautus naudojant veloergometrinio pakopomis didėjančio provokacinio fizinio krūvio metodiką. Pagal tiriamojo lytį, gyvenimo būdą bei sveikatos būklę jie buvo suskirstyti į penkias grupes – sportininkų vyrų (  $n = 161$ ,  $S_v$  – grupė ), vyrų neturinčių nusiskundimų bei jokių klinikinių simptomų (  $n = 111$ ,  $V$  – grupė ), išemine širdies liga, sergančių vyrų (  $n = 61$ ,  $I_v$  – grupė ), sportininkų moterų (  $n = 210$ ,  $S_m$  – grupė ) ir moterų, neturinčių klinikinių simptomų (  $n = 49$ ,  $M$  – grupė ). Taip pat, tiriant amžiaus įtaką gautiems rezultatams, tiriamieji neturintys klinikinių simptomų pagal amžių ir lytį buvo suskirstyti į šešias grupes: 20-30 metų moterys ( $M_{20-30}$ ) ir vyrai ( $V_{20-30}$ ), 30-40 metų moterys ( $M_{30-40}$ ) ir vyrai ( $V_{30-40}$ ), 40-50 metų moterys ( $M_{40-50}$ ) ir vyrai ( $V_{40-50}$ ). Tiriamojo kontingento duomenys pateikti 1 priede.

Iš kompiuterizuotos 12 derivacijų elektrokardiogramos analizės sistemos „Kaunas – Krūvis“ analizuojamų veloergometrinių mėginių atrinkti reikalingi tyrimui pradiniai parametrai (  $\check{S}SD$ ,  $S$ ,  $D$ ,  $JT$ ) bei duomenys apie pacientą (vardas, pavardė, lytis, amžius, ūgis, svoris, požymis grupės, kuriai jis priskiriamas ir pan.) saugomi tekstinėje byloje „Duomenys.txt“. Vienam tyrimui skirta viena eilutė. Išvestinius parametrus palikome apskaičiuoti programai, siekdami sumažinti įvedamų duomenų klaidų tikimybę. Tai nagrinėjami parametrai santykiečiai  $SDS = \frac{S-D}{S}$ ,  $JTRR = \frac{JT}{RR}$  bei skirtumas  $SD = S - D$ .

Parametras  $RR$  apskaičiuojamas pagal formulę:

$$RR = \frac{60}{\check{S}SD}.$$

Maksimalus, praktiškai pasiekiamas galingumas vyrams buvo 300 W, o moterims 175 W. Galingumas vyrams keistas laipteliais nuo 0 iki 300, kas 50 W, o moterims 0, 50, 75, 100, 125, 150, 175 W. Krūvis didinamas kas minutę ir tęsiamas iki submaksimalaus širdies susitraukimo dažnio arba kol atsiranda klinikiniai požymiai, ribojantys krūvį. Taip pat nagrinėti parametrai kitimai per 3 – 5 atsigavimo minutes, priklausomai nuo to, kaip greit jie atsistatydavo. Norint išlaikyti vienodą duomenų struktūrą, lauką, kuriame duomens nebuvo žymėjome brūkšneliu, o programiškai įrašydavome kodą 999, kurio skaičiavimuose nenaudodavome. Analogiškai, po parametro  $\check{S}SD$ , išvardinami kiti parametrai, t.y.  $S$ ,  $D$ ,  $JT$ , su reikšmėmis gautomis prie atitinkamo galingumo.

Pradinių duomenų pavyzdys yra 2.1 lentelėje:

## 2.1 lentelė

Pradinių duomenų struktūros pavyzdys

Pavardė	Vardas	Amžius, m.	Svoris, kg	Ūgis, cm	ŠSD_0W	ŠSD_50W	ŠSD_100W	...
Petraitis	Petras	22	70	193	72	91	111	...
Jonaitis	Jonas	29	98	184	91	112	125	...
...	...	...	...	...	...	...	...	...

Darbe nagrinėjome parametrų kitimą laike. Pagrindinė problema, jog kiekvieno parametro, gauto tyrimo metu, taškų skaičius svyravo nuo 7 iki 12, priklausomai nuo to, kokią galingumą tiriamasis išvystė (kiek truko tyrimas).

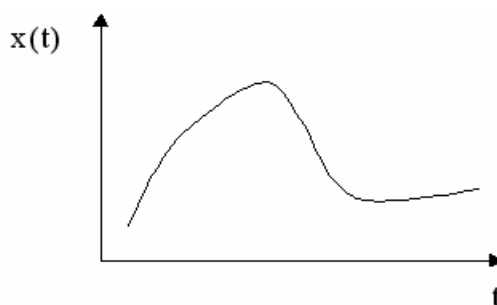
## 2.2. Tyrimas

Pradiniai duomenys, tai kiekvieno tyrimo metu gauti visų nagrinėjamų parametrų diskretūs matavimai priklausantys nuo laiko:

$$S_i, SD_i, SDS_i, JT_i, RR_i, JTRR_i, \quad i = 1..j, \quad j = 7..12.$$

Akivaizdu, jog norint atlikti gilesnę šių dydžių analizę, turime labai mažai matuotų taškų. Tenka imtis tam tikrų gudrybių ir taškų skaičių padidinti. Šiame darbe, tai bandėme padaryti dviem būdais – interpoliuodami pradinius matavimus Lagranžo polinomu bei kubiniu interpoliaciniu splainu.

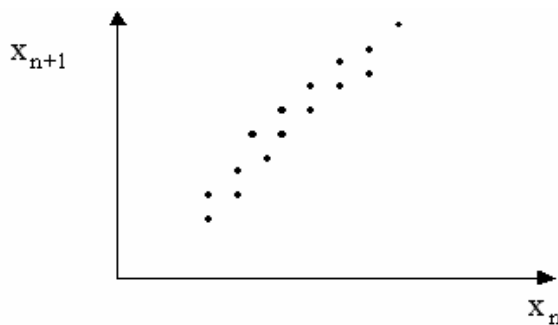
Kaip žinome, interpoliuojant Lagranžo polinomu, funkcijos galuose atsiranda didelės paklaidos. Siekdami sumažinti tas paklaidas, bandėme kelis algoritmus polinomo eilei didinti. Vienas jų, parašytas Matlab aplinkoje, pateikiamas 2 priede. Pagrindinė problema, kad ši tikslinimo procedūra trukdavo pakankamai ilgai – nuo kelių akimirų iki keliolikos minučių ar valandos. Todėl interpoliavimo patikslintu Lagranžo polinomu atsisakėme ir pasirinkome interpoliacinį kubinį splainą, kuriuo šis procesas visada trukdavo tik kelias akimirkas.



2.1 pav. Parametro funkcinė priklausomybė, gauta interpoliuojant kubiniu splainu

Gavę funkcinę išraišką (pavyzdys pateikiamas 2.1 pav.), tam tikru žingsneliu galime apskaičiuoti funkcijos reikšmes tuose taškuose ir taip turėsime didesnę duomenų masę, kurio reikšmės bus tarsi tarpinės tyrimo reikšmės, neprieštaraujančios žmogaus fiziologijai. Reikšmės skaičiuojamos žingsniu  $h = 0,001$ .

Naudojantis apskaičiuotomis funkcijos reikšmėmis sudaromas dvimatis reikšmių sklaidos žemėlapis, t.y. jos atidedamos plokštumoje  $x_n O x_{n+1}$ . Pavyzdys pateikiamas 2.2 pav.



**2.2 pav. Taškų sklaidos diagramos pavyzdys**

Ant parametro gautos taškų sklaidos diagramos uždedamas tinklėlis, kurio vieno langelio aukštis ir plotis yra  $\varepsilon$ .

Skaičiuojamos fraktalinės dimensijos – užimtumo (Hausdorfo, „capacity dimension“), informacinė („information dimension“) ir koreliacinė („correlation dimension“).

Kadangi tyrimo metu nustatyta, kad dimensijos priklauso nuo nagrinėjamų parametrų reikšmių aibės (jos kai kuriems parametrams labai skirtingos: pvz. S reikšmių aibė (80, 240), o JT (1,8; 4,32)) (sumuota per dvyliką derivacijų), tai norėdami gautus skaičius palyginti bei suvienodinti  $\varepsilon$  – tinklėlio dydį, pradinius duomenis normavome. Bendru atveju:

$$NP_i = \frac{P_i - MP}{P_{\max} - P_{\min}}.$$

Čia  $NP_i$  normuota  $P_i$  reikšmė, gauta iš jos atėmus vidurkį ir gautąją reikšmę padalinus iš nagrinėjamo parametro fiksuoto maksimalios ir minimalios ribų skirtumo.

Kad  $\varepsilon$  - tinklėlis būtų pastovus visiems tiriamiesiems koordinatinių ašis apibrėžiamame

$$\begin{aligned} x_{\min} &= y_{\min} = -1, \\ x_{\max} &= y_{\max} = 1. \end{aligned}$$

Parametrų normos pateikiamos 2.2 lentelėje.

## 2.2 lentelė

## Parametrų normos

Parametras	Minimali riba	Maksimali riba
S	80	240
S-D	20	220
(S-D)/S	0.2	1
JT	1.8	4.32
RR	0.2727	1.2
JT/RR	3.6	7.8

Fiksuokime tinklelio langelio dydį  $\varepsilon = 0,05$  ir apskaičiuokime reikšmes, neskaičiuodami ribos, kai  $\varepsilon \rightarrow 0$ :

$$d_{už} = \frac{\log(N_{0,05})}{\log(0,05)}.$$

Užimtumo dimensija leidžia įvertinti, kiek užima nagrinėjamas objektas, tačiau neparodo kaip tankiai užpildomi pasirinkto dydžio  $\varepsilon$  langeliai.

Gauti tirtų parametrų užimtumo dimensijų vidurkiai tiriamųjų grupėse pateikti lentelėje (vidurkis  $\pm$  standartinis nuokrypis):

## 2.3 lentelė

## Tiriamų grupių parametrų užimtumo dimensijų vidurkiai

	Išemine širdies liga sergantieji	Įvairaus amžiaus vyrai	Sportininkai vyrai	Įvairaus amžiaus moterys	Sportininkės moterys
(S-D)/S	0,91 $\pm$ 0,10	1,04 $\pm$ 0,13	1,18 $\pm$ 0,11	0,99 $\pm$ 0,12	1,10 $\pm$ 0,13
JT/RR	0,97 $\pm$ 0,13	1,05 $\pm$ 0,10	1,08 $\pm$ 0,09	1,06 $\pm$ 0,10	1,09 $\pm$ 0,08
RR	1,00 $\pm$ 0,11	1,03 $\pm$ 0,09	1,10 $\pm$ 0,10	1,01 $\pm$ 0,08	1,12 $\pm$ 0,10
JT	0,99 $\pm$ 0,10	1,00 $\pm$ 0,10	1,07 $\pm$ 0,08	1,01 $\pm$ 0,08	1,12 $\pm$ 0,09
S	1,10 $\pm$ 0,10	1,11 $\pm$ 0,08	1,15 $\pm$ 0,07	0,99 $\pm$ 0,10	1,08 $\pm$ 0,09
S-D	0,97 $\pm$ 0,12	1,03 $\pm$ 0,11	1,19 $\pm$ 0,10	0,93 $\pm$ 0,12	1,07 $\pm$ 0,12

Informacinė dimensija – tai užimtumo dimensijos apibendrinimas. Ji įvertina kaip tankiai užpildomas tinklelio langelis. Į ją įeinanti Šenono entropijos formulė leidžia įvertinti vidutinę informaciją vienam langeliui.

$$d_{\text{inf}} = \sum_{i=1}^N \frac{0,05 P_i(0,05) \log[P_i(0,05)]}{\log(0,05)}$$

Gauti nagrinėtų parametru informacinių dimensijų vidurkiai tiriamųjų grupėse pateikti 2.4 lentelėje (vidurkis ± standartinis nuokrypis):

2.4 lentelė

**Tiriamų grupių parametru informacinių dimensijų vidurkiai**

	Išemine širdies liga sergantieji	Įvairaus amžiaus vyrai	Sportininkai vyrai	Įvairaus amžiaus moterys	Sportininkės moterys
(S-D)/S	0,54±0,09	0,67±0,10	0,76±0,10	0,61±0,10	0,70±0,11
JT/RR	0,57±0,10	0,66±0,09	0,67±0,09	0,66±0,09	0,69±0,08
RR	0,60±0,09	0,65±0,08	0,71±0,09	0,62±0,08	0,73±0,09
JT	0,60±0,09	0,65±0,09	0,69±0,08	0,64±0,09	0,73±0,09
S	0,70±0,10	0,74±0,09	0,77±0,06	0,64±0,09	0,71±0,08
S-D	0,59±0,11	0,68±0,10	0,80±0,10	0,58±0,11	0,69±0,12

Koreliacinė dimensija – tai tikimybinė dimensija, parodanti tikimybę rasti porą atsitiktinių taškų langelyje. Skirtingai nuo informacinės dimensijos, kuri nagrinėja tikimybę rasti bent vieną tašką langelyje, ji parodo porų skaičių duoto atstumo langelyje.

$$d_{\text{kor}} = - \frac{\log \left[ \sum_{i=1}^N (P_i(0,05))^2 \right]}{\log(0,05)}$$

Gauti tirtų parametru koreliacinių dimensijų vidurkiai tiriamųjų grupėse pateikti 2.5 lentelėje (vidurkis ± standartinis nuokrypis):

2.5 lentelė

## Tiriamų grupių parametru koreliacinių dimensijų vidurkiai

	Išemine širdies liga sergantieji	Įvairaus amžiaus vyrai	Sportininkai vyrai	Įvairaus amžiaus moterys	Sportininkės moterys
(S-D)/S	0,49±0,09	0,61±0,10	0,69±0,11	0,55±0,10	0,63±0,12
JT/RR	0,51±0,10	0,59±0,09	0,61±0,08	0,60±0,09	0,64±0,09
RR	0,54±0,09	0,60±0,08	0,65±0,09	0,57±0,08	0,68±0,09
JT	0,55±0,10	0,61±0,10	0,64±0,08	0,59±0,10	0,67±0,09
S	0,64±0,10	0,70±0,10	0,72±0,07	0,60±0,09	0,66±0,08
S-D	0,55±0,11	0,63±0,11	0,74±0,10	0,53±0,11	0,64±0,12

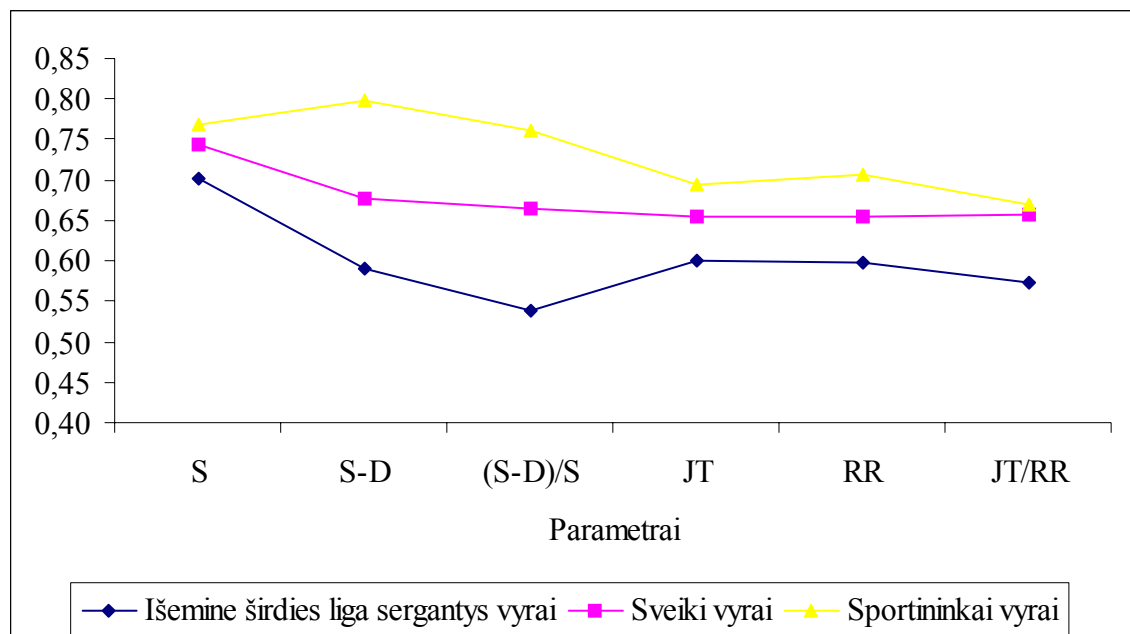
Nagrinėjamų parametru dimensijų (užimtumo, informacinės ir koreliacinės) vidurkiai, fiksuotoms  $\varepsilon$  reikšmėms, kai  $\varepsilon = 0,01$  ir  $\varepsilon = 0,1$  pateikti atitinkamai priede 3 ir priede 4.

Jau pagal vidurkių reikšmes kiekvienam parametru, pastebime, jog patvirtinamas literatūroje pateikiamas faktas:

$$d_{kor} \leq d_{inf} \leq d_{už}.$$

Keičiant  $\varepsilon$  pastebime, kad keičiasi dimensijų reikšmių vidurkiai bei standartinis nuokrypis, t. y. mažėjant  $\varepsilon$  didėja dimensijų vidurkiai, o standartinis nuokrypis mažėja.

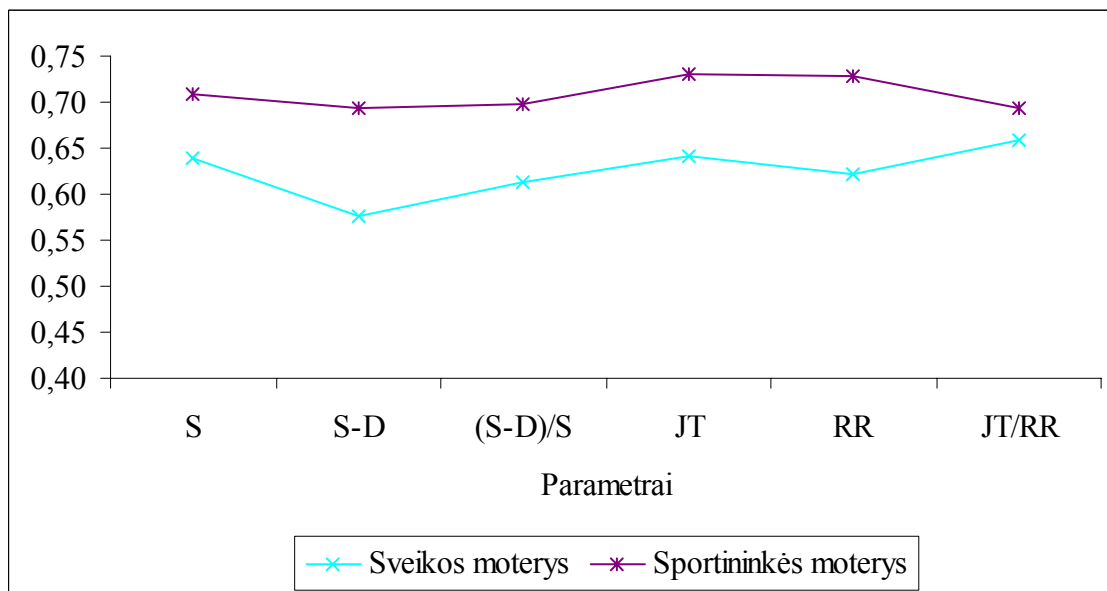
Palyginkime parametru informacinių dimensijų vidurkių dinamiką (2.3 pav. ir 2.4 pav.).



2.3 pav. Informacinių dimensijų vidurkių dinamika vyrų grupėse



Kaip matome didžiausios parametru reikšmės gaunamos sportininkams vyrams, mažiausios reikšmės – išemine širdies liga sergantiems vyrams. O sveikiems vyrams gautos reikšmės išsidėsto per vidurį. Didžiausi skirtumai stebimi parametruose  $S - D$ ,  $\frac{S - D}{S}$ . Todėl darome prielaidą, jog tirtoms vyrų grupėms skyrėsi diastolinio kraujo spaudimo dinamika. Ar kitų parametru skirtumai grupėsi buvo reikšmingi, vėliau patikrinsime statistiniais metodais.



#### 2.4 pav. Informacinių dimensijų vidurkių dinamika moterų grupėse

Tirtose moterų grupėse, stebime, kad sportininkių moterų dimensijų vidurkiai didesni nei sveikų moterų. Jų dinamika yra panaši, tik sveikų moterų parametru skirtumai didesni nei sportininkių. Kaip ir 2.4 pav. Stebime, kad parametro  $\frac{JT}{RR}$  skirtumas yra mažiausias.

5 priede pateikiame bendrą informacinių dimensijų dinamiką moterų ir vyrų grupėms.

6 priede pateikiame paveikslus iliustruojančius užimtumo dimensijų vidurkių dinamiką, o 7 priede – koreliacinių dimensijų. Išskyrus dimensijų vidurkių reikšmes, kitų didelių skirtumų nestebime. Tendencijos tirtose grupėse išlieka panašios visoms dimensijoms.

8 priede ir 9 priede pateikiamos visų nagrinėjamų parametru dimensijų vidurkių dinamika, kai  $\varepsilon=0,01$  ir  $\varepsilon = 0,1$ . Pastebėsime, jog  $\varepsilon = 0,1$  parametru dinamika panašesnė į  $\varepsilon = 0,05$ , tačiau  $\varepsilon = 0,05$  geriau atskiria grupes.  $\varepsilon = 0,01$  dinamika skiriasi nuo kitų, grupių parametrai persipina. Vieni parametru dimensijos sumažėja, o kitų padidėja tiriamoje grupėje.

Mažėjant  $\varepsilon$  tiriamų dimensijų kitimo tendencijos grupėms supanašėja. Kai  $\varepsilon = 0,1$  užimtumo ir koreliacinės dimensijų dinamika skyrėsi nuo informacinės.

Palyginkime, ar gautosios dimensijos skiriasi statistinėmis priemonėmis. Kadangi dauguma tiriamų imčių nėra pasiskirstę pagal  $N(\mu, \sigma^2)$ , tai neparametrinių kriterijų pagalba, tikrinsime hipotezę:

$H_0$  : kintamųjų skirstiniai yra vienodi,

$H_A$  : kintamųjų skirstiniai nėra vienodi.

Naudosime Mano – Vitnio – Vilkoksono rangų sumų kriterijų nepriklausomoms imtims. Skaičiuojama statistika:

$$Z = \frac{U_1 - \mu}{\sigma}; \mu = \frac{n_1 n_2}{2}, \sigma = \sqrt{\frac{n_1 n_2 (n_1 + n_2 + 1)}{12}}, U_1 = n_1 n_2 + \frac{n_1 (n_1 + 1)}{2} - R_1.$$

Čia  $n_1, n_2$  - tirtų imčių dydžiai,  $R_1$  – rangų, priskirtų pirmos imties nariams, suma.

Pasirinktas reikšmingumo lygmuo  $\alpha = 0,05$ .

2.6 lentelė

**Palyginimas parametrų užimtumo dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
Išemine širdies liga sergantieji	Įvairaus amžiaus vyrai	<0.0001	<0.0001	0.078	0.322	0.659	0.002
Išemine širdies liga sergantieji	Sportininkai vyrai	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001	<0.0001
Įvairaus amžiaus vyrai	Sportininkai vyrai	<0.0001	0.097	<0.0001	<0.0001	<0.0001	<0.0001

Kiti nagrinėtų parametrų dimensijų skirstinių palyginimai pateikiami 10 priede. Apibendrinant galima pasakyti, kad daugeliu atveju nagrinėjamų parametrų skirstiniai tirtoms vyrų bei moterų grupėms statistiškai reikšmingai skyrėsi.

Lyginant išemine širdies liga sergančius vyrus su sveikais vyrais, visoms  $\varepsilon$  reikšmėms užimtumo dimensijos JT ir RR parametrų skirstiniai buvo vienodi, tačiau kitoms dimensijoms – skyrėsi.

Lyginant sportininkus vyrus su sveikais, vyrais, kurie sportuoja mažai arba visai nesportuoja, visoms  $\varepsilon$  reikšmėms visų tirtų dimensijų  $\frac{JT}{RR}$  parametro skirstiniai buvo vienodi.

Lyginant neturinčių klinikinių nusiskundimų vyrų grupes pagal amžių (11 priedas), pastebėta, kad  $V_{40-50}$  grupės parametro  $\frac{JT}{RR}$  ir  $V_{20-30}$  grupės parametro  $S$  dimensijų skirstiniai statistiškai reikšmingai skyrėsi nuo kitų vyrų amžiaus grupių. Tuo tarpu moterų grupėse skiriasi -  $M_{40-50}$  grupės

parametrai  $S, S-D$ , o  $M_{20-30}$  grupės parametro  $JT$  dimensijų skirstiniai statistiškai reikšmingai skyrėsi nuo kitų moterų amžiaus grupių.

Dimensijų įvertis yra apskaičiuotų reikšmių  $I(\varepsilon)$  ir atidėtų plokštumoje  $(I(\varepsilon), \log(\varepsilon))$ , juos aproksimuojančios tiesės (mažiausių kvadratų metodu) krypties koeficientas (1.11). 12 priede pateikiame nagrinėtų fraktalinių dimensijų vidurkių dinamiką vyrų, sergančių išemine širdies liga, ir sveikų vyrų (V) grupėse. Patikrinus hipotezę apie parametru dimensijų skirstinių lygybę, gautas rezultatas, kad užimtumo dimensijos skirstiniai yra vienodi – ji stebėtų vyrų grupių neišskiria, pagal nė vieną tirtą modelio parametru ( $p < 0,05$ ). Informacinės dimensijos įverčių skirstiniai skyrėsi visiems parametrams, išskyrus  $JT$  ( $p = 0,08$ ). Koreliacinės dimensijos įverčių skirstiniai skyrėsi parametrams  $RR$  ( $p < 0,0001$ ) ir  $JT$  ( $p = 0,001$ ). Taigi geriausiai parametrus skyrė informacinė dimensija.

### 2.3. Programos aprašymas

Duomenų analizei pagal užsakymą buvo sukurta taikomoji programa vartotojui, žmogaus fiziologijos specialistui. Ji leidžia analizuoti jau turimus duomenis, yra galimybė juos pildyti rankiniu būdu bei iš tam tikros struktūros tekstinių bylų (tai ypač patogu, jei norime duomenų bazę papildyti didesniu tyrimų skaičiumi). Norint perkelti duomenis, galime juos rašyti į tekstinius bylas (eksportuoti duomenis). Pateikiama grafinė informacija, padedanti matyti, įvairias nagrinėjamų parametru kombinacijas, jų sąsajas. Tai įvairios fazinės plokštumos (realizuotos remiantis baigtiniais skirtumais), taškų dinamiką nusakančios sklaidos diagramos, kurioms ir buvo skaičiuotos fraktalinės dimensijos, vaizduojama normuotųjų parametru dinamika laike krūvio bei atsigavimo metu, jungiant taškus tiesėmis bei kubiniu interpoliaciu splineu.

Pradiniai duomenys saugomi duomenų bazėje. Kurią sudaro trys lentelės: „Gydytojai.db”, „Pacientai.db”, „Irasai.db”. Lentelių pavadinimai tiesiogiai atspindi jų saugomos informacijos turinį, t.y. lentelėje „Gydytojai.db” saugoma informacija apie gydytoją, „Pacientai.db” – informacija apie pacientus, o lentelėje „Irasai.db” – tai visi pacientams atlikti tyrimai ir jų duomenys.

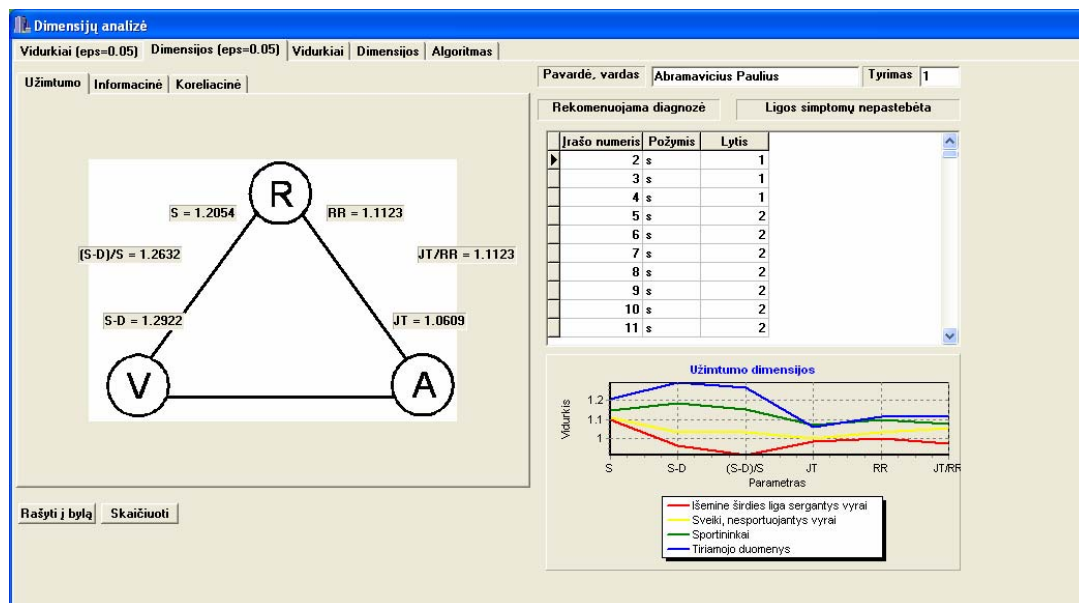
Kol kas programa pritaikyta dirbti vienam gydytojui, tačiau organizuota struktūra lengvai leistų programą modifikuoti ir nematydami vienas kito duomenų galėtų dirbti keli gydytojai.

Reikėtų pastebėti, jog programa buvo kuriama bendradarbiaujant su kolege Kristina Berškiene, kadangi mūsų atliekamų darbų vadovas ir užsakovas yra tas pats asmuo. Jis pageidavo, kad programinė dalis būtų apjungta. Todėl, paleidus programą, pirmame lange, matysime, jog darbą atliko du asmenys. Tačiau tikslu atskirti atliktą darbą bei skirtingą darbų tematiką, šalia pavardžių yra padėti mygtukai, kuriuos paspaudus galima patekti į tos autorės atliktą darbą iliustruojančią programą.

Realizuota galimybė susirasti norimą pacientą pagal pavardę ir vardą.

Mygtukas „Filtrai” atidaro langą, skirtą analizuoti norimos grupės tyrimus. Galima peržiūrėti įvairią grafinę informaciją, ieškant tendencijų būdingų tos grupės tiriamiesiems. Tam tereikia pasirinkti tiriamųjų lytį ( yra galimybė analizuoti abiejų lyčių tiriamuosius ) bei grupę iš atitinkamų sąrašų, pateikiamų ekrane.

Mygtukas „Dimensijos” leidžia analizuoti tiriamų grupių tyrimams paskaičiuotus visų trijų dimensijų – informacinės, užimtumo bei koreliacinės, vidurkius skaitmenine bei grafine forma (2.5 pav.). Galima palyginti, kaip išsidėsto tiriamojo duomenys lyginant su kitomis tirtomis tos lyties grupėmis.



2.5 pav. Programos langas dimensijų analizei atlikti

Remiantis logistine regresija, sudarytas kriterijus, skiriantis sveikus vyrus (ne sportininkus) nuo išemine širdies liga sergančiųjų. Pagal visus parametrus ir visas dimensijas teisingai klasifikuota 92,4% (sveiki teisingai klasifikuoti 94,6%, o ligoniai 88,5%). Palikus apskaičiuotus parametrų  $\frac{S-D}{S}$ ,  $JT$  visų dimensijų reikšmes, teisingai klasifikuota 83,7% tiriamųjų (sveikų – 88,3%, ligonių 75,4%). Tuomet klasifikavimo formulė atrodo taip:

$$f = -14,71 + 32,491 \cdot C\_SDS + 56,086 \cdot C\_JT - 97,754 \cdot I\_SDS - 100,245 \cdot I\_JT + 49,097 \cdot Co\_SDS + 37,225 \cdot Co\_JT.$$

Kai  $f \geq 0$  programa pateikia išvadą, kad galimi ligos simptomai, priešingu atveju – „ligos simptomų nepastebėta”.

Atlikus logistinę regresiją reikšmėms gautoms nefiksavus  $\varepsilon$ , pagal visus parametrus teisingai klasifikuota 80,2% tyrimų (sveikų 88,3%, o ligonių 65,6%).

Nereikalingi tyrimai trinami paspaudus mygtuką „Trinti”, esantį prie įrašų, o pacientas su visais jam atliktais įrašais trinamas paspaudus mygtuką „Trinti”, esantį prie paciento lango dalies.

Norint grįžti į pradinį langą, spaudžiamas mygtukas „Grįžti”, o išeiti iš programos, spaudžiamas „x”, esantis dešiniame viršutiniame lango kampe.

## IŠVADOS

1. Parametrų dimensijos geriausiai išsiskyrė atlikus skaičiavimus, kai  $\varepsilon = 0,05$ . Skaičiuotų dimensijų – užimtumo, informacinės, koreliacinės, parametrų vidurkių tendencijos panašiausios. Geriausiai grupes skyrė informacinė dimensija (sportininkų vyrų ir vyrų neturinčių klinikinių nusiskundimų grupėse statistiškai reikšmingai skyrėsi visi parametrai, išskyrus JT/RR bei S, moterų grupėse skyrėsi visi parametrai).
2. Mažai sportuojančių ar nesportuojančių neturinčių klinikinių nusiskundimų asmenų grupėse tirta amžiaus įtaka visoms dimensijoms. Vyrų grupėse statistiškai reikšmingai išsiskyrė parametrai S ir JT/RR, o moterų grupėse – S, S-D, JT.
3. Parašyta taikomoji programa, skirta vartotojui – žmogaus fiziologijos specialistui.
4. Atliktas darbas – tai tik dar vienas žingsnelis, siekiant giliau nagrinėti žmogaus organizme vykstančius procesus. Gautus rezultatus reikėtų tikrinti su kitais duomenimis. Tai padėtų lengviau juos interpretuoti. Be to, įdomu, jog skyrėsi išemine širdies liga sergančių vyrų visų parametrų dimensijos (informacinės ir koreliacinės) nuo kitų nagrinėtų vyrų grupių. Galbūt, tai padės tiksliau ir anksčiau diagnozuoti išeminę širdies ligą.
5. Modifikuotas fraktalinių dimensijų skaičiavimo algoritmas, t.y. skaičiavimų atlikimas fiksuotoms  $\varepsilon$  reikšmėms, sutaupo laiko ir duoda geresnius rezultatus, skiriant tirtas grupes.

## LITERATŪRA

1. Ašeriškytė D. Maksimalaus išvystomo galingumo prognozavimas fizinio krūvio metu / Ašeriškytė D., Matusevičiūtė K., Vainoras A., Navickas Z. // Biomedicininė inžinerija: tarptautinės konferencijos pranešimų medžiaga [Kauno technologijos universitetas, 2003 m. spalio 23,24 d.] Kaunas, 2003. p. 54 - 57.
2. Matusevičiūtė K. Kelių sąveikaujančių žmogaus fiziologinių sistemų analizė ir modeliavimas / Matusevičiūtė K., Ašeriškytė D., Vainoras A., Navickas Z. // Biomedicininė inžinerija: tarptautinės konferencijos pranešimų medžiaga [Kauno technologijos universitetas, 2003 m. spalio 23,24 d.] Kaunas, 2003. p. 50 - 53.
3. Berškienė K. Sistolinio kraujo spaudimo ir elektrokardiogramos ryšys / Berškienė K., Ašeriškytė D., Vainoras A., Navickas Z., Sedekerskienė V., Vitartaitė A. // Biomedicininė inžinerija: tarptautinės konferencijos pranešimų medžiaga [Kauno technologijos universitetas, 2004 m. spalio 28,29 d.] Kaunas, 2004. p. 180 - 184.
4. Berškienė K. Is it possible to evaluate the systolic arterial blood pressure by electrocardiographic parameters? / Berškienė K., Ašeriškytė D., Vainoras A., Navickas Z., Sedekerskienė V., Vitartaitė A. // IFMBE Proceedings: Medicine and Health Telematics 2004 „Health in the Information Society“ Naples 2004, 308-8 no 522 [4p.].
5. Ašeriškytė D. Funkcinis ryšys tarp dvigubos sandaugos ir JT intervalo / Ašeriškytė D., Matusevičiūtė K., Vainoras A., Navickas Z. // Taikomoji matematika: V studentų konferencijos pranešimų medžiaga [Kauno technologijos universitetas, 2004 m. gegužės 8 d.] Kaunas, 2004. p. 10 - 12.
6. Matusevičiūtė K. Sistolinio kraujo spaudimo prognozavimas pagal EKG parametrus / Matusevičiūtė K., Ašeriškytė D., Vainoras A., Navickas Z. // Taikomoji matematika: V studentų konferencijos pranešimų medžiaga [Kauno technologijos universitetas, 2004 m. gegužės 8 d.] Kaunas, 2004. p. 43 - 45.
7. Plukas, Kostas. Skaitiniai metodai ir algoritmai: vadovėlis aukštųjų mokyklų studentams. Kaunas, 2001. 549 p. ISBN 9955-03-061-5
8. Kvedaras, Bronius; Sapagovas, Mifodijus. Skaičiavimo metodai: vadovėlis respublikos aukštųjų mokyklų matematikos specialybei. Vilnius, 1974. 516 p.
9. Pekarskas, Vidmantas. Diferencialinis ir integralinis skaičiavimas: vadovėlis. Kaunas, 1997-2003 ISBN 9986-13-417-X
10. Valantinas, Jonas. Fraktalinė geometrija: vadovėlis. Kaunas, 1999. 186 p. ISBN 9986-13-705-5
11. Matematiniai metodai. Prieiga per internetą:  
<<http://mathworld.wolfram.com/>>
12. Fraktalinių dimensijų skaičiavimas natūraliems objektams. Prieiga per internetą:

<<http://www.umanitoba.ca/faculties/science/botany/labs/ecology/fractals/fractal.html>>

13. Asmens sveikatos ugdymas. Mokomoji knyga. Kaunas, 2000. 144 psl.

14. Šilanskienė A. Žmogaus organizmo funkcinės būklės kitimo ilgalaikių treniruočių metu vertinimas: daktaro disertacijos santrauka: biomedicinos mokslai: medicina ( 07B )/ Aistė Šilanskienė; Kauno medicinos universitetas. Kaunas, 2003. 32 psl.

15. Išeminės širdies ligos diagnozė. Paruošė V.Grinius. Prieiga per internetą:

<[http://www.kvsc.lt/04\\_sveikata/c\\_gyv\\_sveikata/ischemine.htm](http://www.kvsc.lt/04_sveikata/c_gyv_sveikata/ischemine.htm)>

16. Išeminė širdies liga. Paruošė Vaitiekūnas V. Prieiga per internetą:

<<http://kardiologas-grinius.w3.lt/AKA.htm>>

17. Čekanavičius V, Murauskas G. Statistika ir jos taikymai I, II. Vilnius, 2001.

18. Vidžiūnas A. C++ ir C++ Builder pradmenys. Kaunas, 2002. 315 psl.

19. Blonskis J. Programavimo C++ Builder pavyzdžiai / Jonas Blonskis, Vytautas Bukšnaitis, Vacius Jusas, Romas Marcinkevičius, Jonas Smolinskas. Kaunas, 2002. 291 psl.

## TIRIAMOJO KONTINGENTO DUOMENYS

<b>Grupė</b>	<b>Tyrimų skaičius grupėje</b>	<b>Amžius (M ± SEM)</b>
Sportininkų vyrų	161	23,56 ± 0,40
Sportininkų moterų	49	24,88 ± 0,65
Vyrų	111	36,05 ± 0,60
20-30	26	28,37 ± 0,56
30-40	56	35,36 ± 0,35
40-50	29	43,97 ± 0,60
Moterų	210	33,97 ± 0,53
20-30	66	25,17 ± 0,44
30-40	100	35,14 ± 0,28
40-50	44	44,52 ± 0,50
Vyrų, sergančių išemine širdies liga	61	52,44 ± 1,56



### DUOMENŲ PAPILDYMAS INTERPOLIUOJANT LAGRANŽO POLINOMU

Pagal pradinus duomenų masyvus duom1 ir duom2, tam tikru užsiduodamu žingsniu h ieškoma reikšmių viršijančių tikslumą eps. Jos keičiamos atitinkamo taško reikšme ant tiesės, gautos pradiniais duomenimis. Algoritmas stabdomas, kai neberandama taškų, viršijančių eps.

```
function [duomN1, duomN2] = mainHRA1(h, eps, duom1, duom2);
r = 0;
f = 1;
duomN1 = duom1;
duomN2 = duom2;

[lll kkk]=size(duom1);
kk = 1;
while (f == 1)
    f = 0;
    [m1, m2, m3]=maximumA(duomN1, duomN2, h, duom1, duom2, kk, kk+1);
    buvo = 0;
    if (m3 > eps & f==0)
        kk = 1;
        f = 1;
    else
        kk = kk + 1;
    end;

    if (m3 > eps)

        j = 0;
        b1 = 0;
        [l, c] = size(duomN1);
        for(i=1:l)
            if ( (b1==0) && (m1 < duomN1(i)) && m2~=0)
                j=j+1;
                b1 = 1;
                ddd1(j,:) = m1;
                ddd2(j,:) = m2;
            end;
            j=j+1;
            ddd1(j,:) = duomN1(i);
            ddd2(j,:)= duomN2(i);
        end;
        duomN1 = ddd1;
        duomN2 = ddd2;

    end

    if (kk < lll)
        f = 1;
    end;
end;
```

end

Ieškoma maksimali nukrypimo nuo pradinių duomenų reikšmė, fiksuojama vieta, kurioje jis yra.

```
function[v1, v2, v3]=maximumA(dd1,dd2,h,d1,d2,a1,a2)
ma=0;
v=0;
l=0;
for(i=d1(a1):h:d1(a2))
    reiksme=tiese(d1(a1),d2(a1),d1(a2),d2(a2),i);
    k=reiksme-Funct(i,dd1,dd2);
    if(abs(k)>ma)
        ma=abs(k);
        v=i;
        l=reiksme;
    end;
end;
v1=v;
v2=l;
v3=ma;
```

Apskaičiuojama funkcijos reikšmė taške x, eančiame ant tiesės, kurios pradžia taške (x1,y1), o pabaiga (x2,y2).

```
function y=tiese(x1,y1,x2,y2,x);
a=(y2-y1)/(x2-x1);
b=y1-a*x1;
y=a*x+b;
```

Apskaičiuojama Lagranžo polinomo funkcijos reikšmė taške w, kai pradiniai duomenys yra d1 ir d2.

```
function k=Funct(w,d1,d2)
J=0;
[r, c] = size(d1);
for(i=1:r)
    m=1;
    for(t=1:r)
        if(d1(t) ~= d1(i))
            m=m.*(w-d1(t))/(d1(i)-d1(t));
        end;
    end;
    v=m*d2(i);
    J=J+v;
end;
k=J;
```

**PARAMETRŲ DIMENSIJŲ VIDURKIAI, KAI  $\varepsilon = 0,01$** **Tiriamų grupių parametru užimtumo dimensijų vidurkiai**

	<b>Išemine širdies liga sergantieji</b>	<b>Įvairaus amžiaus vyrai</b>	<b>Sportininkai vyrai</b>	<b>Įvairaus amžiaus moterys</b>	<b>Sportininkės moterys</b>
(S-D)/S	0,93±0,07	1,01±0,09	1,11±0,07	0,98±0,08	1,06±0,08
JT/RR	0,97±0,09	1,03±0,07	1,04±0,06	1,03±0,07	1,05±0,06
RR	0,99±0,07	1,01±0,06	1,06±0,07	1,00±0,06	1,07±0,07
JT	0,98±0,07	0,99±0,07	1,04±0,05	1,00±0,06	1,07±0,05
S	1,06±0,07	1,06±0,07	1,09±0,04	0,99±0,06	1,05±0,05
S-D	0,97±0,08	1,01±0,07	1,12±0,07	0,94±0,08	1,04±0,08

**Tiriamų grupių parametru informacinių dimensijų vidurkiai**

(S-D)/S	0,66±0,07	0,75±0,08	0,82±0,07	0,71±0,08	0,77±0,08
JT/RR	0,68±0,08	0,75±0,07	0,75±0,07	0,75±0,07	0,77±0,07
RR	0,70±0,07	0,75±0,06	0,79±0,07	0,73±0,06	0,80±0,07
JT	0,71±0,07	0,74±0,08	0,78±0,06	0,74±0,07	0,80±0,06
S	0,78±0,08	0,81±0,07	0,83±0,05	0,74±0,07	0,78±0,06
S-D	0,70±0,08	0,76±0,08	0,85±0,07	0,69±0,08	0,77±0,09

**Tiriamų grupių parametru koreliacinių dimensijų vidurkiai**

(S-D)/S	0,60±0,07	0,67±0,08	0,73±0,09	0,64±0,08	0,69±0,08
JT/RR	0,60±0,08	0,67±0,07	0,67±0,07	0,67±0,07	0,69±0,07
RR	0,62±0,07	0,69±0,07	0,72±0,07	0,66±0,07	0,73±0,07
JT	0,64±0,08	0,68±0,09	0,71±0,07	0,67±0,09	0,72±0,08
S	0,69±0,09	0,75±0,09	0,75±0,06	0,68±0,08	0,71±0,07
S-D	0,64±0,09	0,70±0,09	0,77±0,08	0,63±0,09	0,70±0,09

Pastaba. Pateikiame dimensijų vidurkius  $\pm$  standartinis nuokrypis

PARAMETRŲ DIMENSIJŲ VIDURKIAI, KAI  $\varepsilon = 0,1$ 

## Tiriamų grupių parametru užimtumo dimensijų vidurkiai

	Išemine širdies liga sergantieji	Įvairaus amžiaus vyrai	Sportininkai vyrai	Įvairaus amžiaus moterys	Sportininkės moterys
(S-D)/S	0,91±0,17	1,07±0,17	1,24±0,14	1,01±0,16	1,15±0,17
JT/RR	0,99±0,18	1,08±0,13	1,12±0,12	1,09±0,14	1,14±0,10
RR	1,03±0,14	1,06±0,12	0,14±0,14	1,04±0,11	1,17±0,13
JT	1,14±0,13	1,16±0,11	1,21±0,09	1,02±0,14	1,12±0,11
S	0,97±0,16	1,05±0,14	1,26±0,13	0,93±0,16	1,11±0,16
S-D	1,00±0,13	1,02±0,13	1,11±0,11	1,05±0,10	1,17±0,10

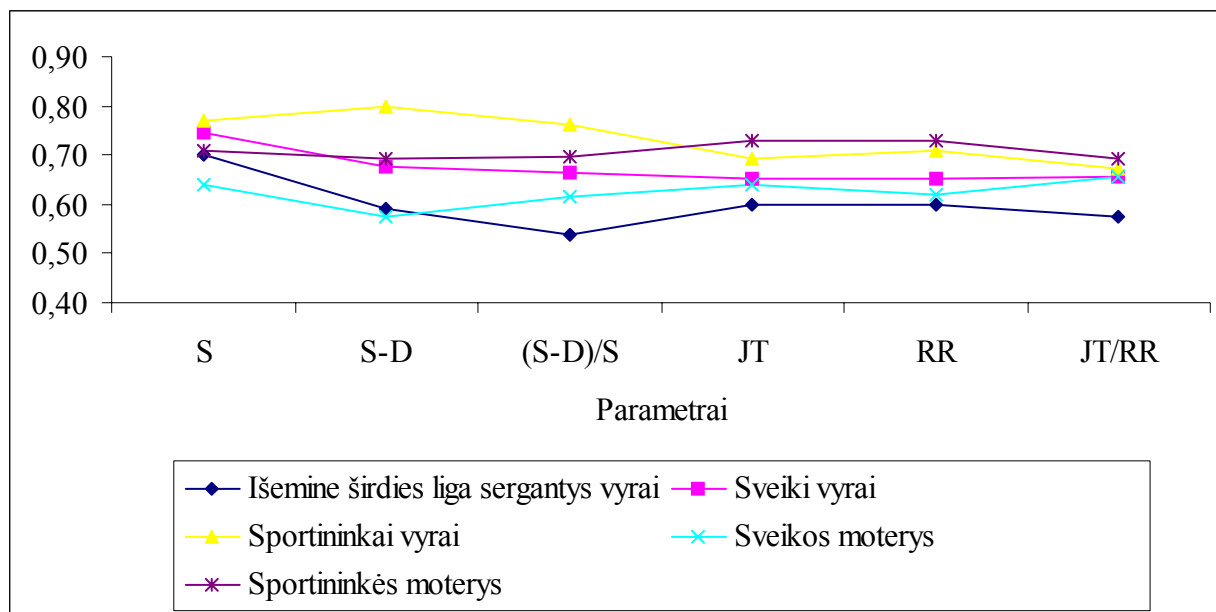
## Tiriamų grupių parametru informacinių dimensijų vidurkiai

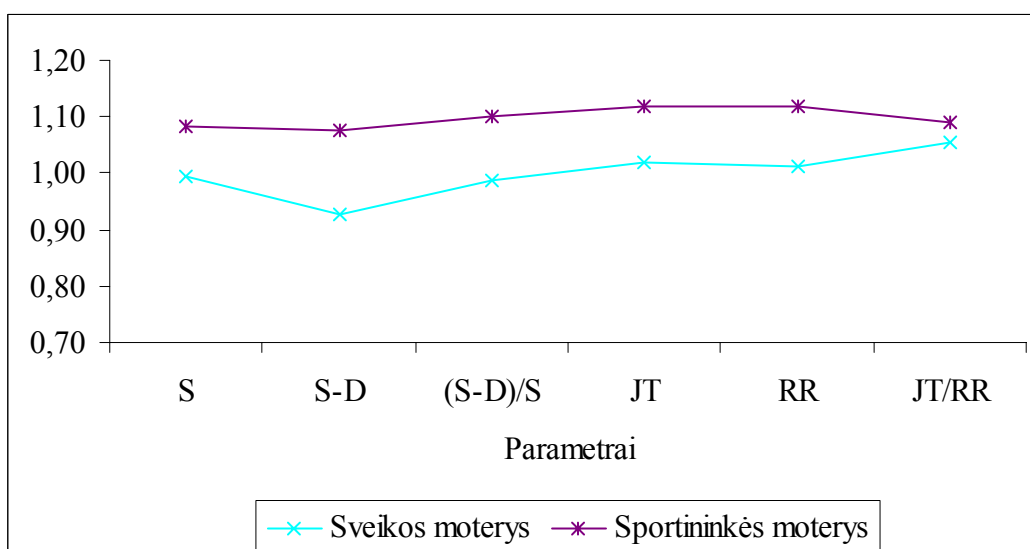
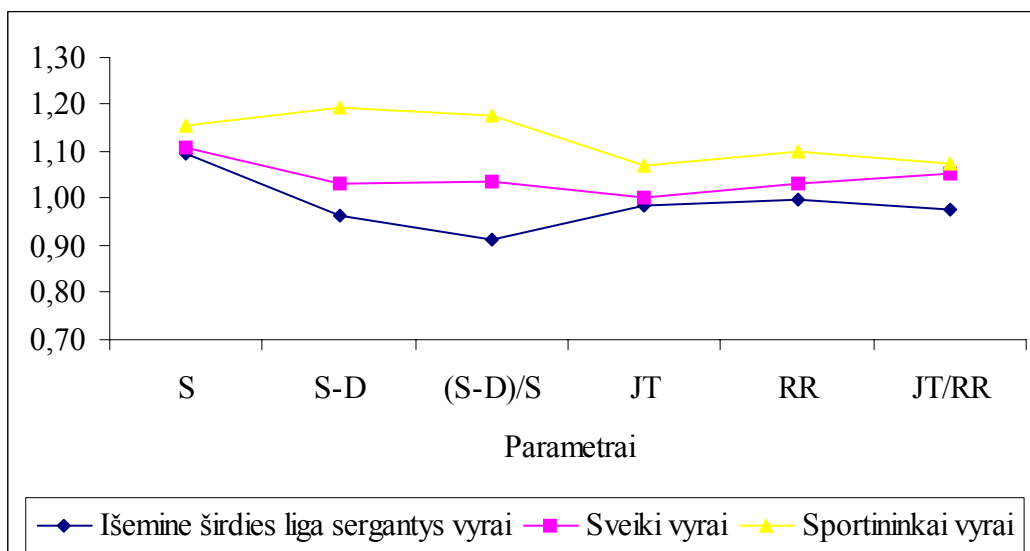
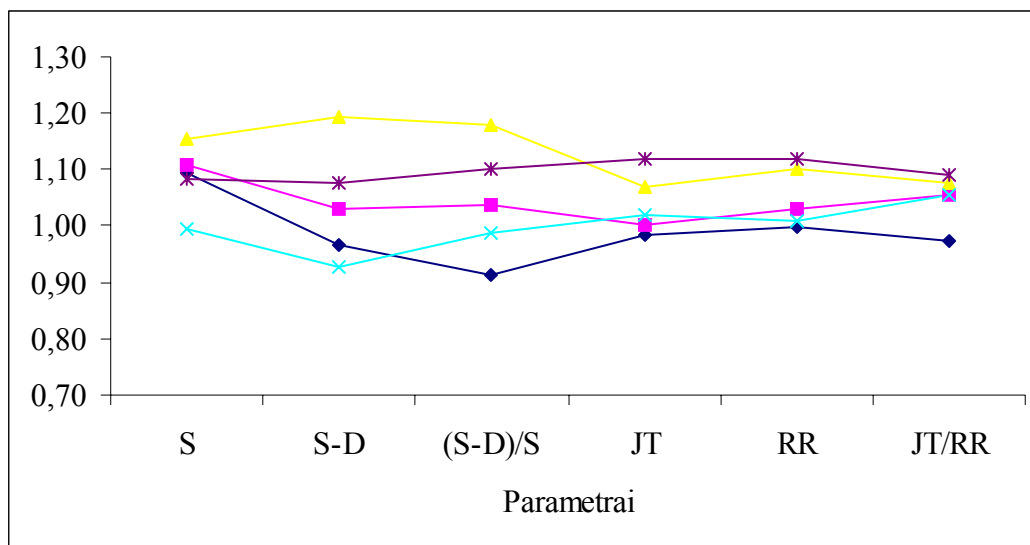
(S-D)/S	0,46±0,11	0,62±0,13	0,74±0,12	0,56±0,13	0,66±0,14
JT/RR	0,52±0,12	0,60±0,10	0,62±0,11	0,61±0,11	0,66±0,10
RR	0,54±0,11	0,60±0,10	0,66±0,11	0,56±0,09	0,69±0,11
JT	0,54±0,11	0,60±0,11	0,65±0,10	0,59±0,10	0,70±0,09
S	0,67±0,12	0,71±0,10	0,74±0,08	0,58±0,11	0,67±0,09
S-D	0,52±0,13	0,63±0,12	0,78±0,12	0,51±0,13	0,65±0,14

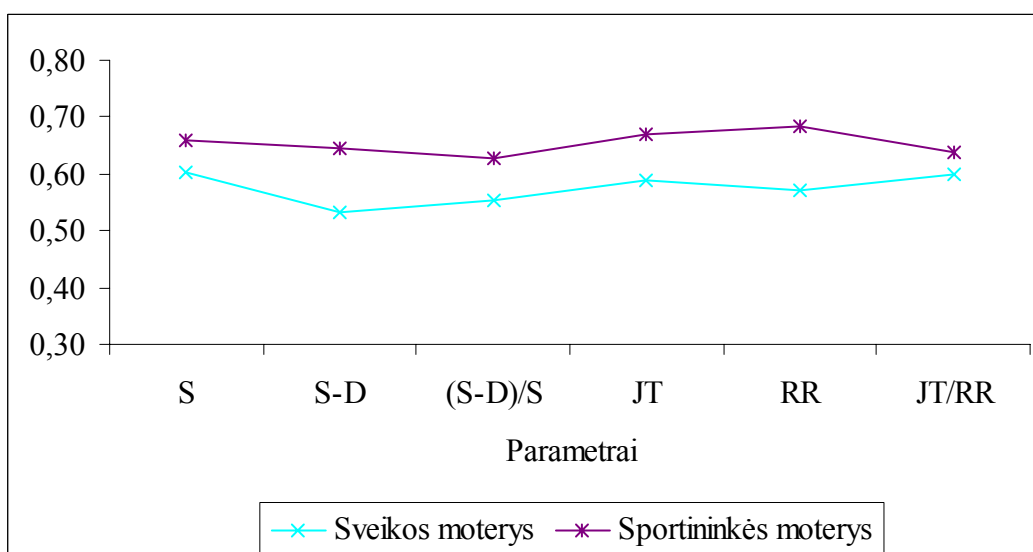
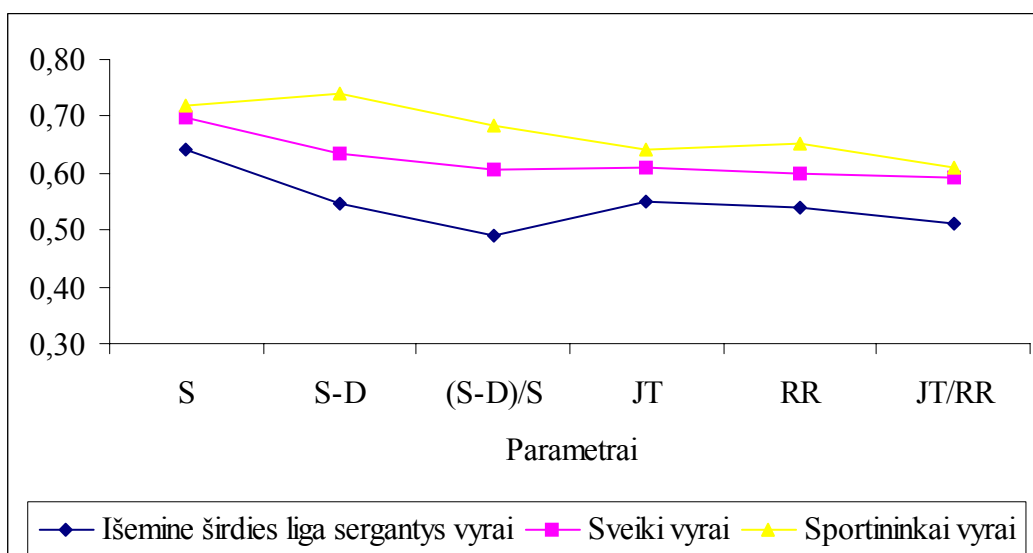
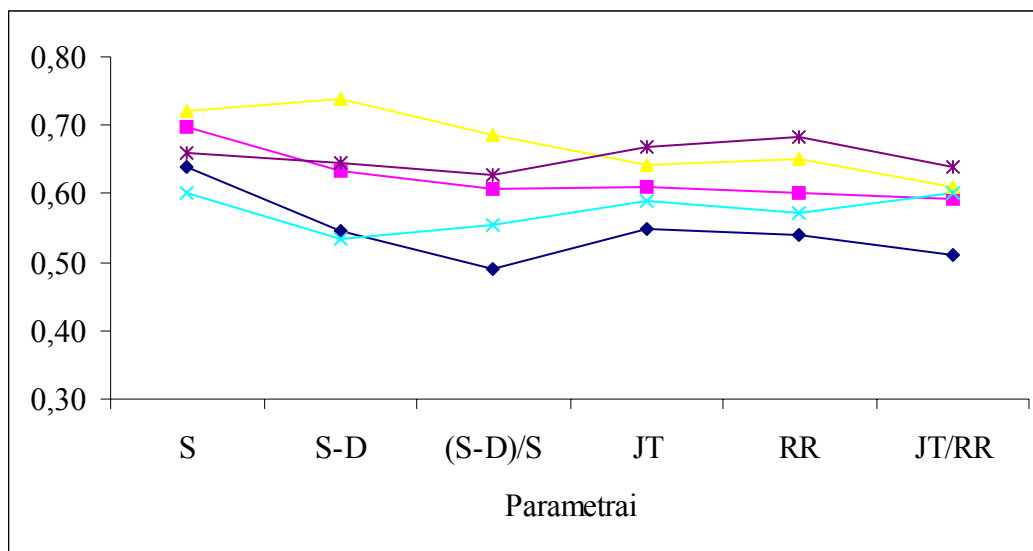
## Tiriamų grupių parametru koreliacinių dimensijų vidurkiai

(S-D)/S	0,41±0,11	0,56±0,13	0,67±0,13	0,50±0,13	0,60±0,15
JT/RR	0,46±0,11	0,55±0,10	0,56±0,11	0,56±0,11	0,61±0,11
RR	0,49±0,11	0,54±0,10	0,60±0,11	0,51±0,09	0,65±0,11
JT	0,61±0,13	0,66±0,11	0,69±0,08	0,54±0,12	0,63±0,10
S	0,48±0,13	0,59±0,13	0,73±0,12	0,46±0,13	0,60±0,15
S-D	0,49±0,12	0,56±0,11	0,60±0,10	0,53±0,11	0,65±0,09

Pastaba. Pateikiame dimensijų vidurkius  $\pm$  standartinis nuokrypis

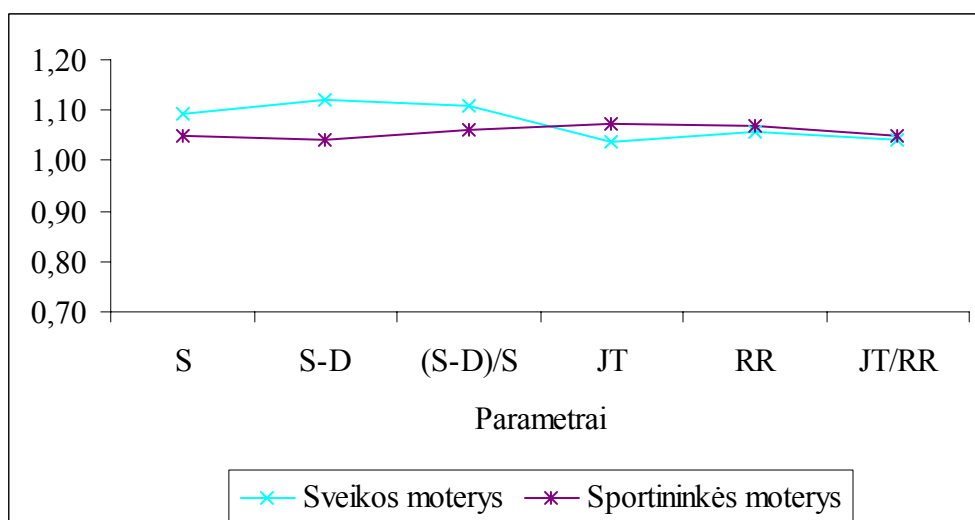
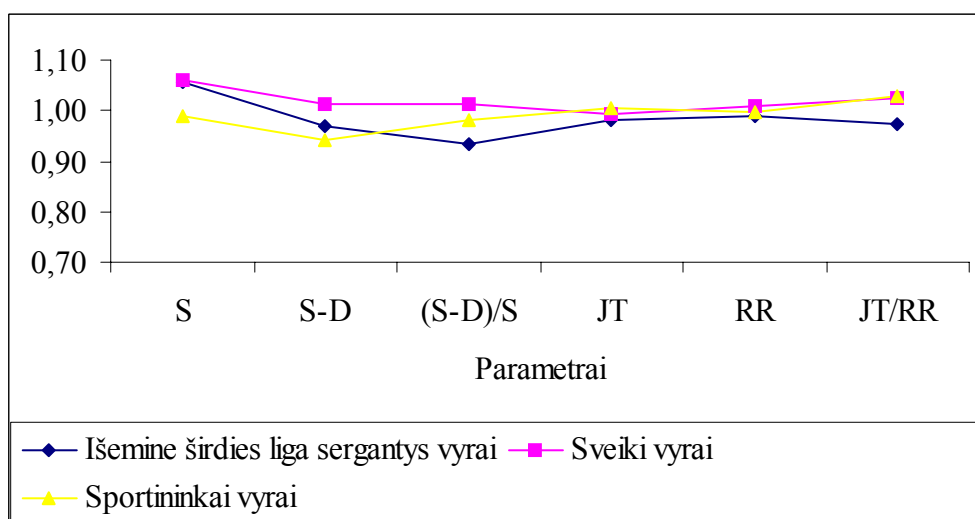
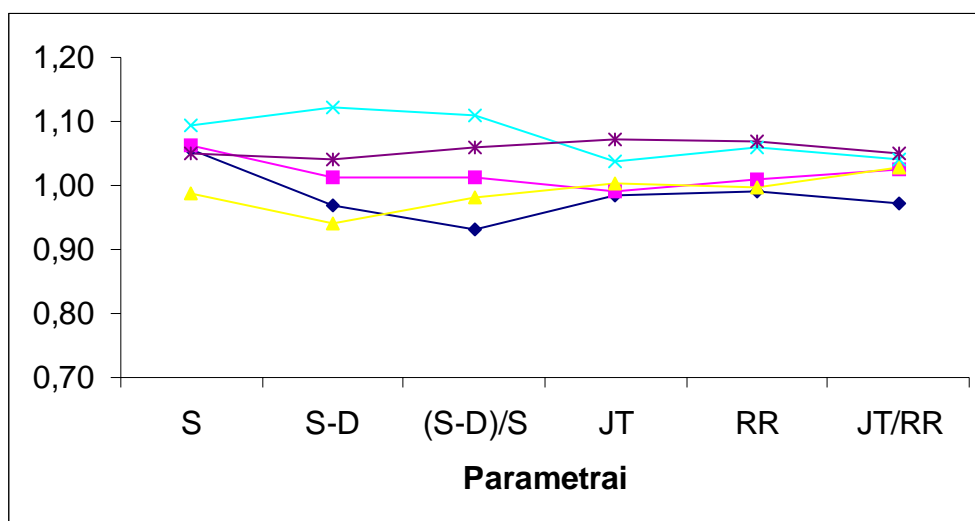
PARAMETRŲ INFORMACINIŲ DIMENSIJŲ VIDURKIŲ DINAMIKA, KAI  $\varepsilon = 0,05$ 

PARAMETRŲ UŽIMTUMO DIMENSIJŲ VIDURKIŲ DINAMIKA, KAI  $\varepsilon = 0,05$ 

PARAMETRŲ KORELIACINIŲ DIMENSIJŲ VIDURKIŲ DINAMIKA, KAI  $\varepsilon = 0,05$ 

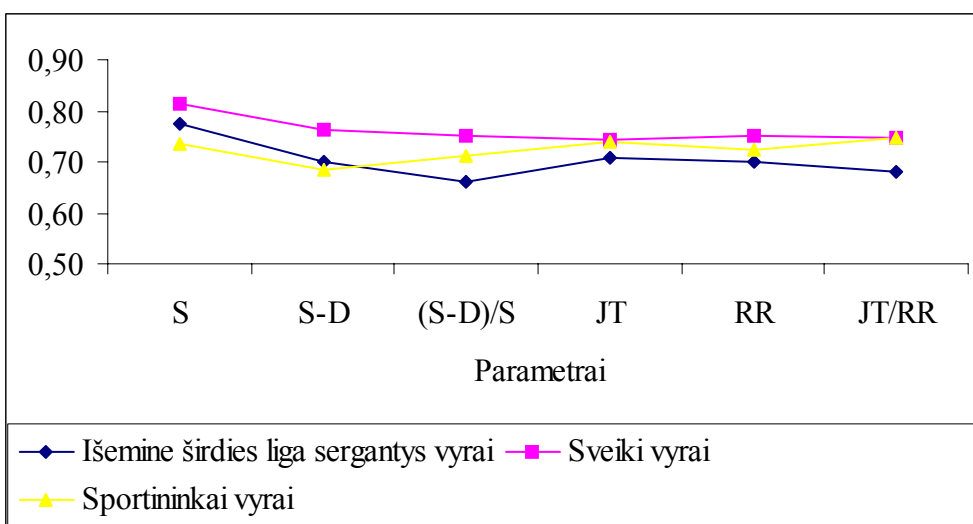
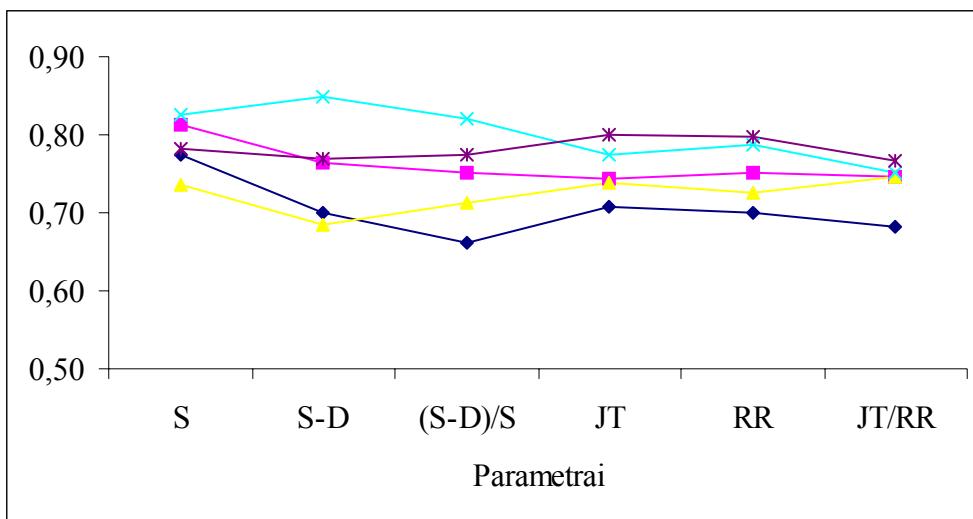
PARAMETRŲ DIMENSIJŲ VIDURKIŲ DINAMIKA, KAI  $\varepsilon = 0,01$ 

## Užimtumo dimensijų vidurkių dinamika

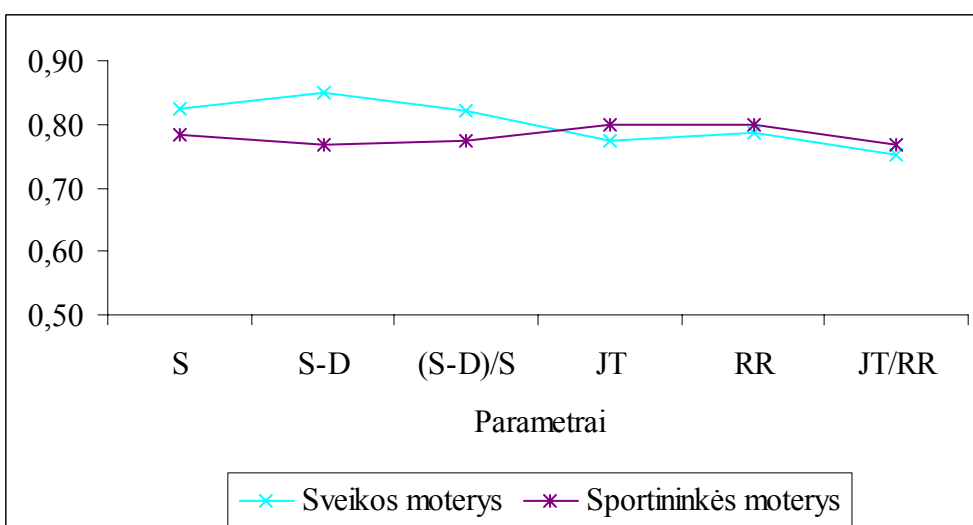




### Informacinių dimensijų vidurkių dinamika

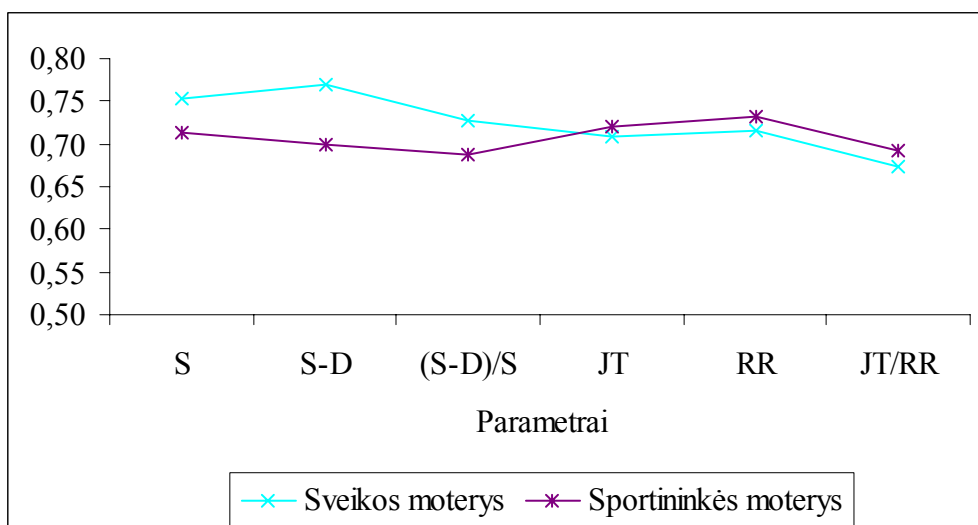
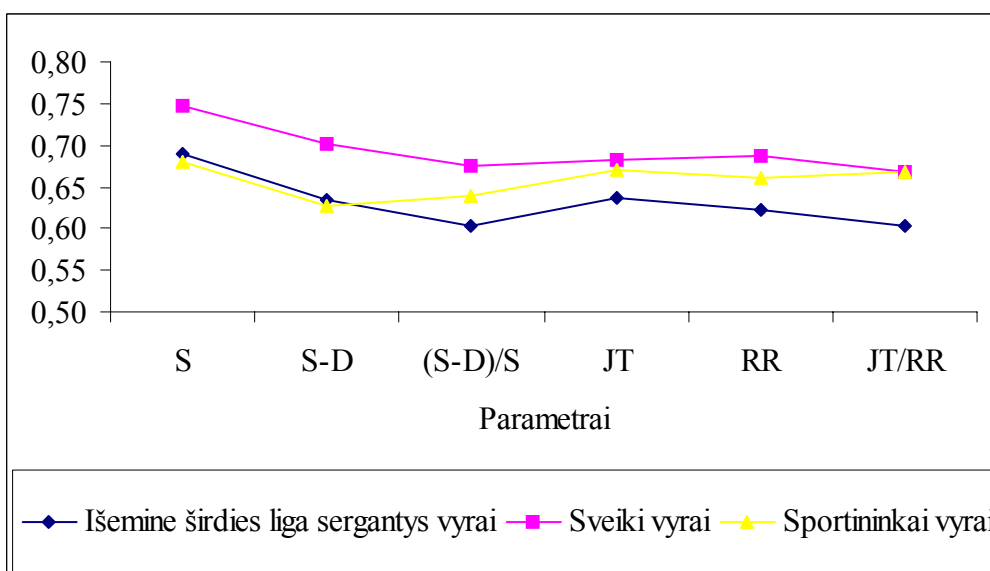
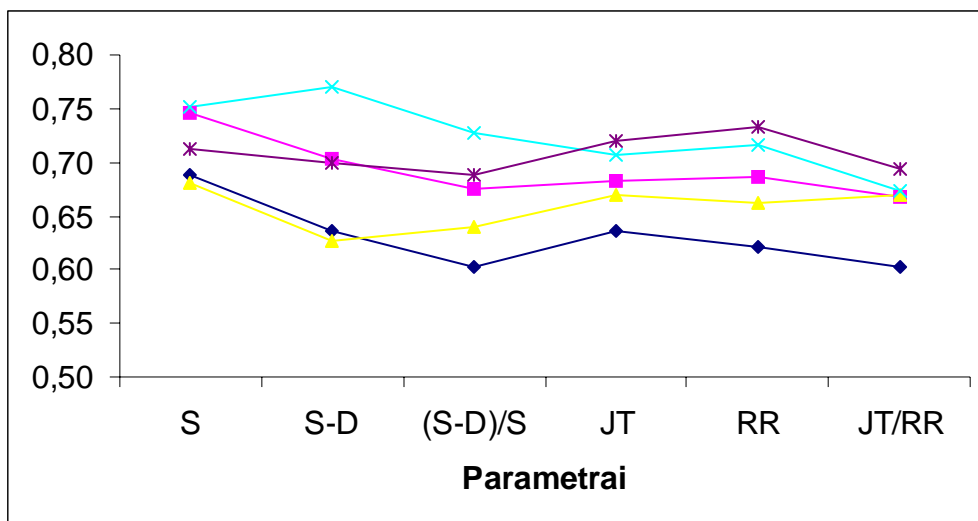


◆ Išeminė širdies liga sergantys vyrai    ■ Sveiki vyrai  
 ▲ Sportininkai vyrai



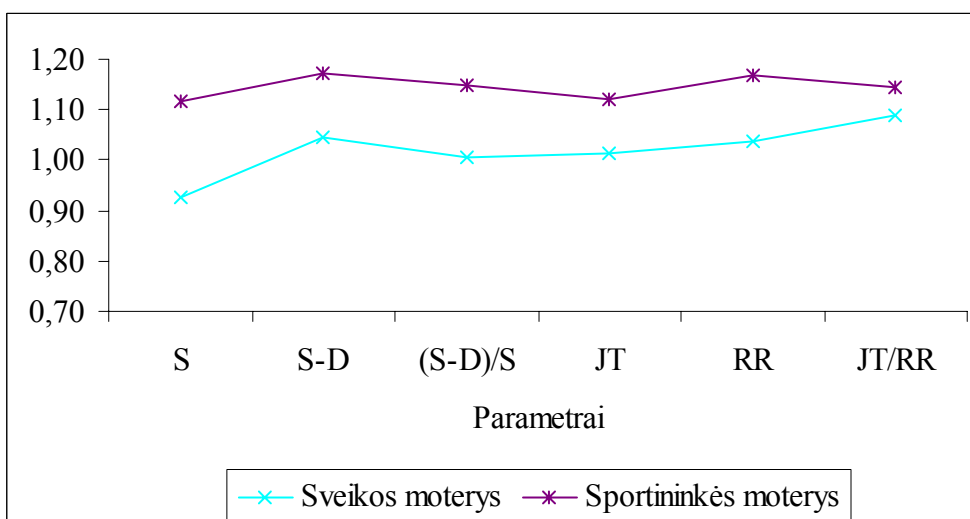
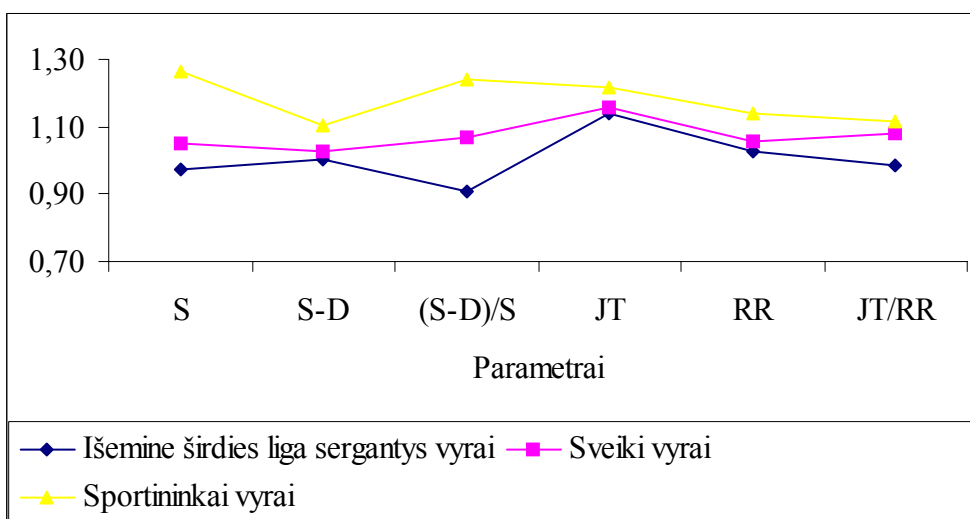
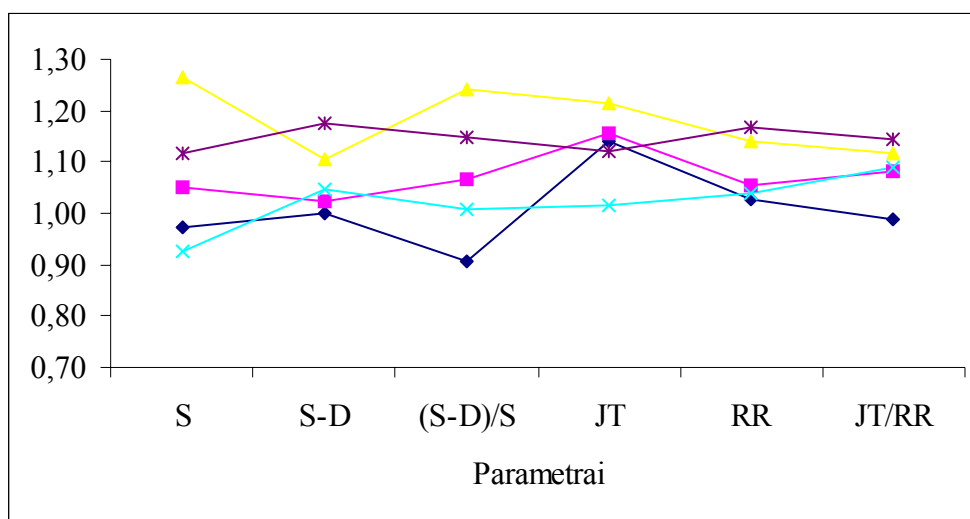
—×— Sveikos moterys    \*— Sportininkės moterys

### Koreliacinių dimensijų vidurkių dinamika

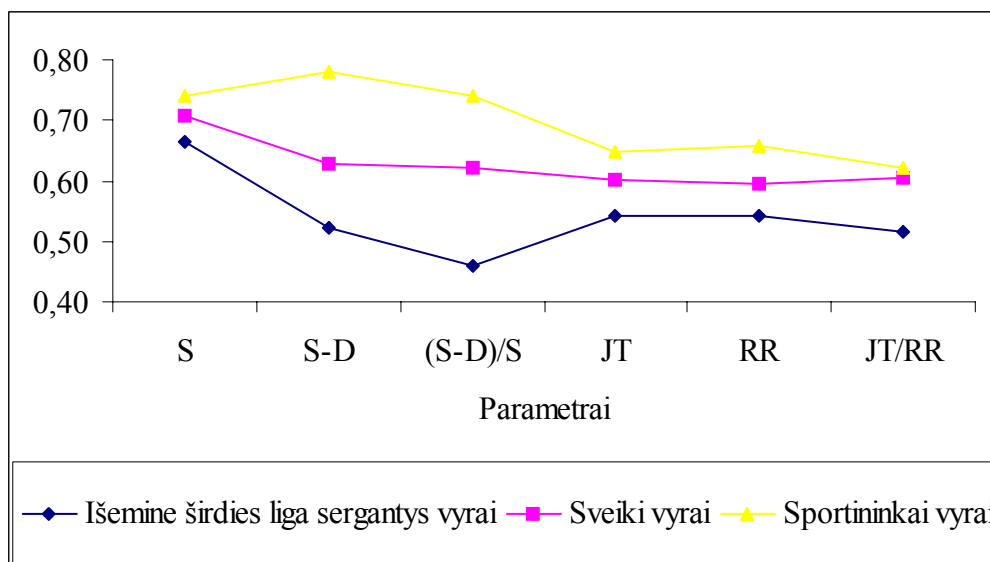
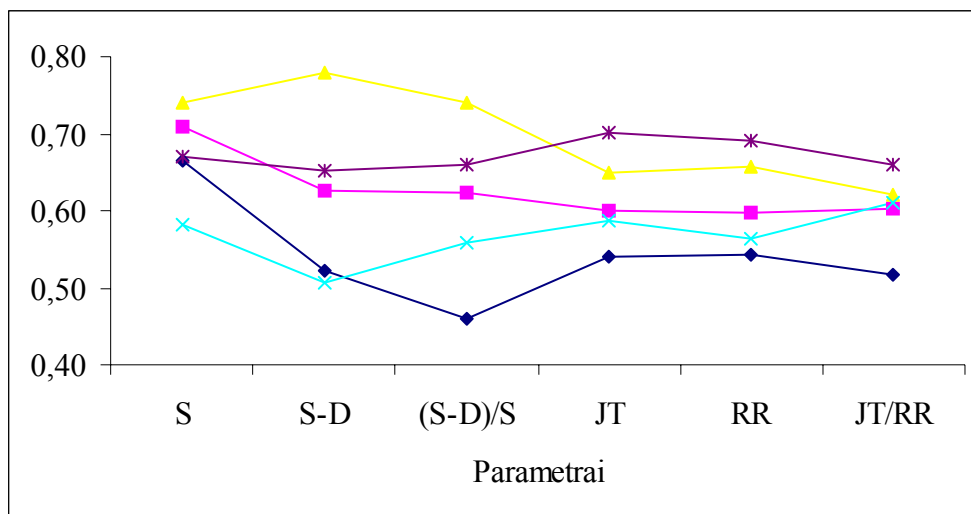


PARAMETRŲ DIMENSIJŲ VIDURKIŲ DINAMIKA, KAI  $\varepsilon = 0,1$ 

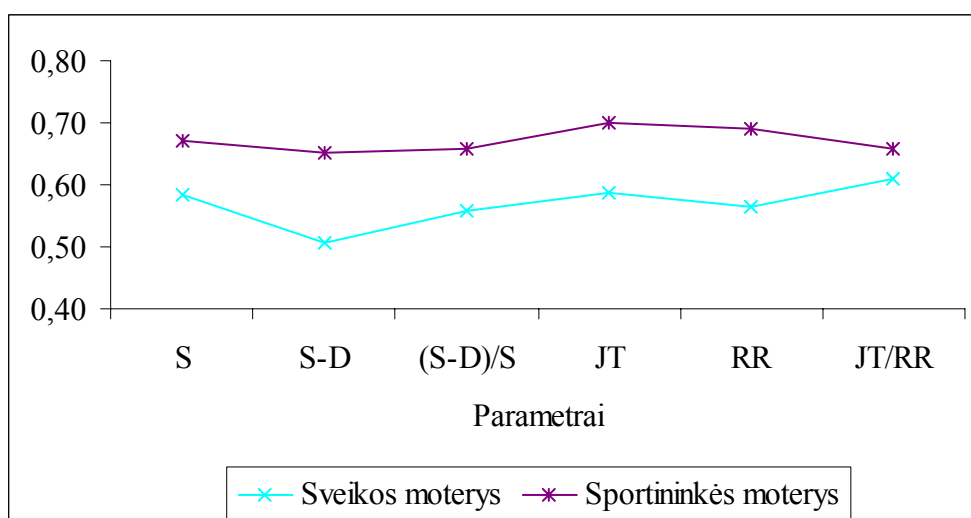
## Užimtumo dimensijų vidurkių dinamika



### Informacinių dimensijų vidurkių dinamika

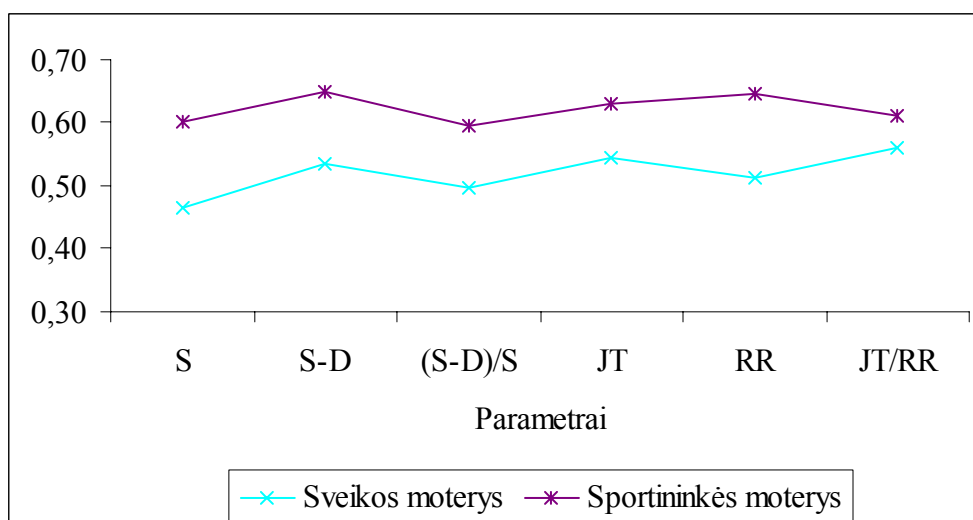
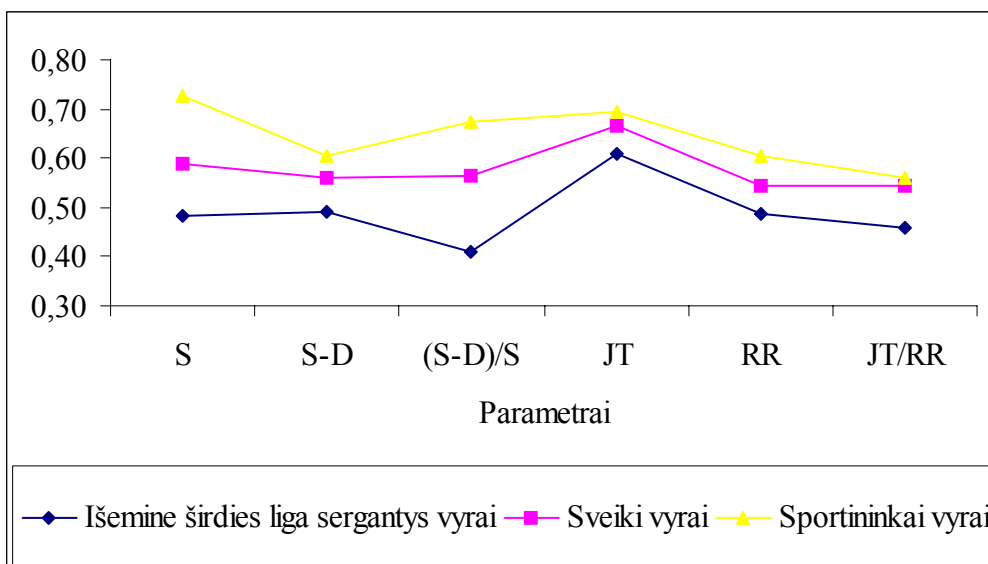
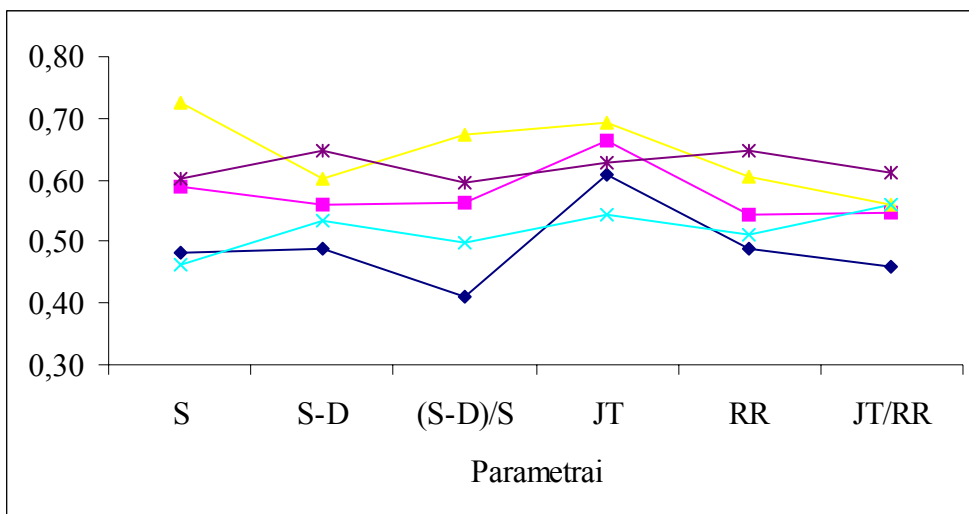


—◆— Išemine širdies liga sergantys vyrai —■— Sveiki vyrai —▲— Sportininkai vyrai



—×— Sveikos moterys —\*— Sportininkės moterys

### Koreliacinių dimensijų vidurkių dinamika



## PARAMETRŲ DIMENSIJŲ SKIRSTINIŲ PALYGINIMAS

Palyginimas parametrų informacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	<0,0001	<0,0001	0,008	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,340	<0,0001	<0,0001	0,037	<0,0001

Palyginimas parametrų koreliacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,190	<0,0001	0,005	0,165	<0,0001

Palyginimas parametrų užimtumo dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,025	<0,0001	<0,0001	<0,0001	<0,0001

Palyginimas parametrų informacinių dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,026	<0,0001	<0,0001	<0,0001	<0,0001

Palyginimas parametrų koreliacinių dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,017	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas užimtumo dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	0,86	0,467	0,755	0,002
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,129	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas informacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	<0,0001	0,002	0,004	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,752	<0,0001	<0,0001	0,195	<0,0001

**Palyginimas koreliacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	<0,0001	0,001	<0,0001	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,475	0,002	0,029	0,890	<0,0001

**Palyginimas užimtumo dimensijų skirstinių tirtose moterų**

**grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,044	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas informacinių dimensijų skirstinių tirtose moterų**

**grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,093	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas koreliacinių dimensijų skirstinių tirtose moterų**

**grupėse (p – reikšmė), kai  $\varepsilon = 0,01$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,059	<0,0001	<0,0001	0,20	<0,0001

**Palyginimas užimtumo dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	0,317	0,721	0,006	0,284
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,023	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas informacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	0,001	<0,0001	0,03	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,363	<0,0001	<0,0001	0,029	<0,0001

**Palyginimas koreliacinių dimensijų skirstinių tirtose vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
I <sub>v</sub>	V	<0,0001	<0,0001	<0,0001	0,004	<0,0001	<0,0001
I <sub>v</sub>	S <sub>v</sub>	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001	<0,0001
V	S <sub>v</sub>	<0,0001	0,354	<0,0001	0,092	<0,0001	<0,0001

**Palyginimas užimtumo dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,007	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas informacinių dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,009	<0,0001	<0,0001	<0,0001	<0,0001

**Palyginimas koreliacinių dimensijų skirstinių tirtose moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,1$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M	S <sub>m</sub>	<0,0001	0,006	<0,0001	<0,0001	<0,0001	<0,0001



## PARAMETRŲ DIMENSIJŲ SKIRSTINIŲ PALYGINIMAS PAGAL AMŽIŲ

Palyginimas užimtumo dimensijų skirstinių vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$ 

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
V <sub>20-30</sub>	V <sub>30-40</sub>	0,504	0,861	0,956	0,901	0,018	0,064
V <sub>20-30</sub>	V <sub>40-50</sub>	0,748	0,021	0,703	0,761	0,019	0,290
V <sub>30-40</sub>	V <sub>40-50</sub>	0,860	0,018	0,727	0,882	0,742	0,714

Palyginimas informacinių dimensijų skirstinių vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$ 

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
V <sub>20-30</sub>	V <sub>30-40</sub>	0,467	0,780	0,936	0,611	0,022	0,154
V <sub>20-30</sub>	V <sub>40-50</sub>	0,698	0,009	0,973	0,919	0,035	0,245
V <sub>20-30</sub>	V <sub>30-40</sub>	0,860	0,011	0,993	0,578	0,897	0,846

Palyginimas koreliacinių dimensijų skirstinių vyrų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$ 

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
V <sub>20-30</sub>	V <sub>30-40</sub>	0,618	0,952	0,905	0,420	0,02	0,188
V <sub>20-30</sub>	V <sub>40-50</sub>	0,448	0,011	0,906	0,736	0,038	0,2
V <sub>30-40</sub>	V <sub>40-50</sub>	0,985	0,020	0,810	0,243	0,746	0,824

**Palyginimas užimtumo dimensijų skirstinių moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$**

Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M <sub>20-30</sub>	M <sub>30-40</sub>	0,047	0,024	0,490	0,026	0,272	0,79
M <sub>20-30</sub>	M <sub>40-50</sub>	0,744	0,004	0,217	0,021	<0,0001	0,048
M <sub>30-40</sub>	M <sub>40-50</sub>	0,096	0,291	0,322	0,454	0,002	0,013

**Palyginimas informacinių dimensijų skirstinių moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$**

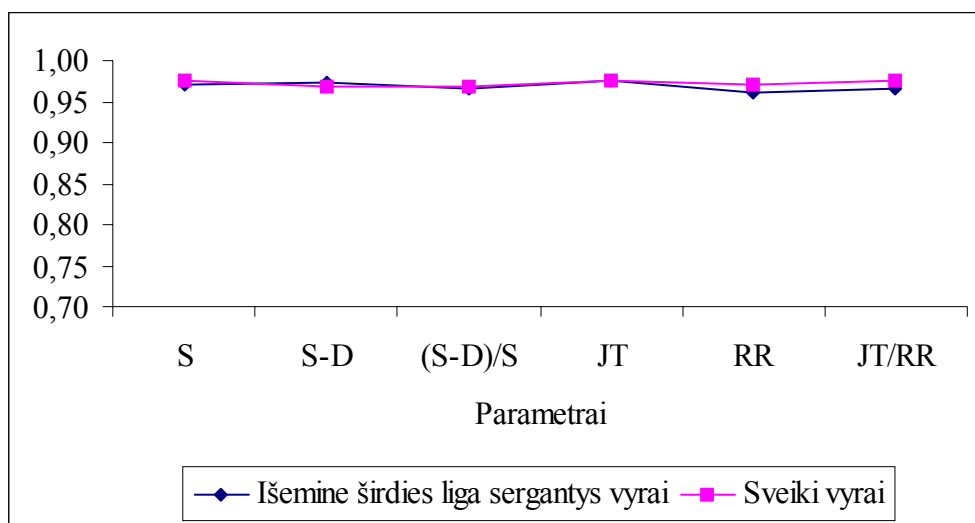
Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M <sub>20-30</sub>	M <sub>30-40</sub>	0,359	0,076	0,185	0,043	0,366	0,966
M <sub>20-30</sub>	M <sub>40-50</sub>	0,951	0,018	0,091	0,024	0,001	0,073
M <sub>30-40</sub>	M <sub>40-50</sub>	0,316	0,280	0,349	0,367	0,004	0,043

**Palyginimas koreliacinių dimensijų skirstinių moterų grupėse (p – reikšmė), kai  $\varepsilon = 0,05$**

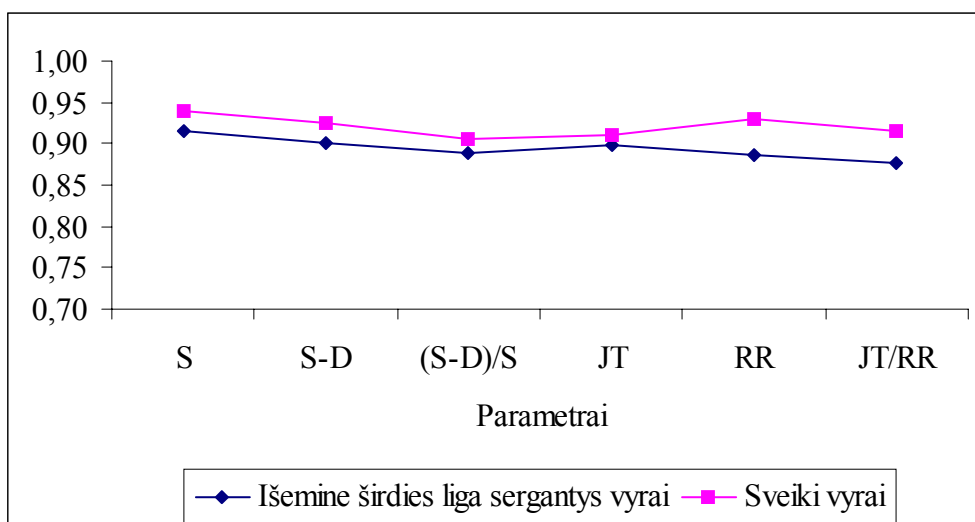
Lyginamos grupės		(S-D)/S	JT/RR	RR	JT	S	S-D
M <sub>20-30</sub>	M <sub>30-40</sub>	0,666	0,259	0,054	0,042	0,391	0,642
M <sub>20-30</sub>	M <sub>40-50</sub>	0,554	0,078	0,026	0,012	0,001	0,028
M <sub>30-40</sub>	M <sub>40-50</sub>	0,256	0,323	0,302	0,231	0,005	0,031

## PARAMETRŲ FRAKTALINIŲ DIMENSIJŲ VIDURKIŲ PALYGINIMAS

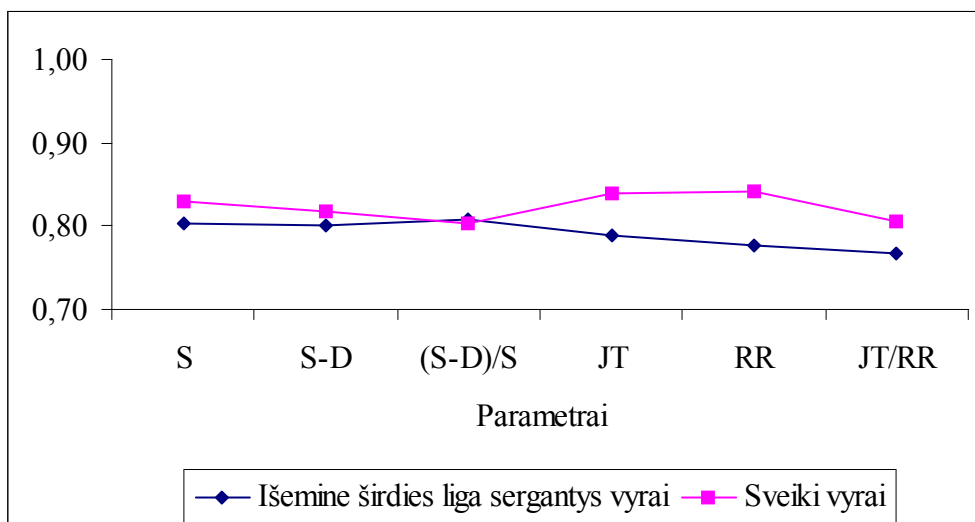
Užimtumo dimensijos vidurkiai



Informacinės dimensijos vidurkiai



Koreliacinės dimensijos vidurkiai



## PROGRAMOS TEKSTŲ FRAGMENTAI

```
//-----
TPagr1 *Pagr1;
const int gg = 12;
//-----
__fastcall TPagr1::TPagr1(TComponent* Owner)
    : TForm(Owner)
{}
//-----
float *TPagr1:: Duomenu_masyvai(short ind )
{
    int i = -1;    float HR[gg];
    if (ind == 0 ){
        if ( DBDM1->IrasaiT->FieldByName("HR_0")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_0")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_50")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_50")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_100")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_100")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_150")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_150")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_200")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_200")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_250")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_250")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_300")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_300")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_1")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_1")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_2")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_2")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_3")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_3")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("HR_4")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_4")->AsInteger;    }
        if ( DBDM1->IrasaiT->FieldByName("HR_5")->AsInteger != 999 ){
            i++;    HR[i] = DBDM1->IrasaiT->FieldByName("HR_5")->AsInteger;    }
        }
        if (ind == 1){
            if ( DBDM1->IrasaiT->FieldByName("S_0")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_0")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_50")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_50")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_100")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_100")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_150")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_150")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_200")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_200")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_250")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_250")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_300")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_300")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_1")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_1")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_2")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_2")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_3")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_3")->AsInteger;    }
                if ( DBDM1->IrasaiT->FieldByName("S_4")->AsInteger != 999 ){
                    i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_4")->AsInteger;    }
            if ( DBDM1->IrasaiT->FieldByName("S_5")->AsInteger != 999 ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_5")->AsInteger;    }    }
        }
        if(ind == 2){
            if ( (DBDM1->IrasaiT->FieldByName("S_0")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_0")->AsInteger != 999) ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_0")->AsInteger - DBDM1->IrasaiT->FieldByName("D_0")->AsInteger;    }
            if ( (DBDM1->IrasaiT->FieldByName("S_50")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_50")->AsInteger != 999) ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_50")->AsInteger - DBDM1->IrasaiT->FieldByName("D_50")->AsInteger;    }
            if ( (DBDM1->IrasaiT->FieldByName("S_100")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_100")->AsInteger != 999) ){
                i++;    HR[i] = DBDM1->IrasaiT->FieldByName("S_100")->AsInteger - DBDM1->IrasaiT->FieldByName("D_100")->AsInteger;    }
        }
    }
}
```





```

    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_250")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_250")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_300")->AsInteger != 999) && (DBDM1->IrasaiT-
>FieldByName("JT_300")->AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_300")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_300")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_1")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("JT_1")-
>AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_1")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_1")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_2")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("JT_2")-
>AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_2")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_2")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_3")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("JT_3")-
>AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_3")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_3")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_4")->AsInteger != 999) && (DBDM1->IrasaiT-
>FieldByName("JT_4")->AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_4")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_4")->AsInteger/60;    }
    if ( (DBDM1->IrasaiT->FieldByName("HR_5")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("JT_5")-
>AsFloat != 999) ){
    i++; HR[i] = (float)DBDM1->IrasaiT->FieldByName("JT_5")->AsFloat*DBDM1->IrasaiT-
>FieldByName("HR_5")->AsInteger/60;    }
    if(ind == 7){
    if ( (DBDM1->IrasaiT->FieldByName("S_0")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_0")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_0")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_50")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_50")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_50")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_100")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_100")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_100")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_150")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_150")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_150")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_200")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_200")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_200")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_250")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_250")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_250")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_300")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_300")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_300")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_1")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_1")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_1")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_2")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_2")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_2")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_3")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_3")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_3")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_4")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_4")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_4")->AsInteger;    }
    if ( (DBDM1->IrasaiT->FieldByName("S_5")->AsInteger != 999) && (DBDM1->IrasaiT->FieldByName("D_5")-
>AsInteger != 999) ){
    i++; HR[i] = DBDM1->IrasaiT->FieldByName("D_5")->AsInteger;    }
    intervalai = i+1;
    return HR;}
//-----
void TPagrl:: Splainas(float *xk, float *yk)
{
int intervalai1 = intervalai -1;
float *h;
h = new float [intervalai];
float *a;
a = new float [intervalai];
float *b;
b = new float [intervalai];
float *c;
c = new float [intervalai];
float *d;
d = new float [intervalai];
h[0]=0; a[0]=0; b[0]=0; c[0]=0; d[0]=0;

```

```

for (int i=1; i<= intervalail; i++)
h[i]=xk[i]-xk[i-1];
for (int i=1; i<= intervalail-1; i++){
    b[i]=(float)-(h[i]+h[i+1])/3;
    if (i==1) a[i]=0;
        else a[i]=(float)h[i]/6;
    if (i==(intervalail-1)) c[i]=0;
        else c[i]=(float)h[i+1]/6;
    d[i]=(float)yk[i-1]/h[i]-yk[i]/h[i]-yk[i]/h[i+1]+yk[i+1]/h[i+1];}
int n1 = intervalail - 1;
float t;
float s[gg], e[gg];
s[1]=0;
e[1]=0;
for (int i=1; i<=n1;i++){
    t = b[i]-a[i]*s[i];
    s[i+1]=(float)c[i]/t;
    e[i+1]=(float) (a[i]*e[i]-d[i])/t;}
m[n1]=e[n1+1];
int j =n1;
for (int i=1; i<=n1;i++) {
m[j]=s[j+1]*m[j+1]+e[j+1];
j--;}
m[0]=0;
m[intervalail]=0;
delete a;
delete b;
delete c;
delete d;
delete h;
}
//-----
float TPagr1:: Reiksme_taske( float *xk, float *yk, float a, int n)
{
int i = 0;
while (xk[i] < a) i = i + 1;
float h = xk[i] - xk[i-1];
float t1 = xk[i] - a;
float t2 = a - xk[i-1];
float p1 = t1*t1*t1/6;
float p2 = t2*t2*t2/6;
float s1 = yk[i-1]-m[i-1]*h*h/6;
float s2 = yk[i]-m[i]*h*h/6;
float ga = (m[i-1]*p1+m[i]*p2+s1*t1+s2*t2)/h;
return ga;
}
//-----
float TPagr1:: ID( float h, int r )
{float dim; float I=0;
for (int i=0; i<r; i++)
    for (int j=0; j<r; j++){
        kiek[i][j]=kiek[i][j]/jj;
        if ( kiek[i][j]!=0 ) I=I+kiek[i][j]*log(kiek[i][j]);    }
if (Dim1->Button5->Tag == 1) {
    I = I*(-1);
    dim = -1*I/log(h);
    return dim;    }
else return I;
}
//-----
float TPagr1:: CD( float h, int r )
{float dim; float I=0;
for (int i=0; i<r; i++)
    for (int j=0; j<r; j++){    if ( kiek[i][j] ) I=I+1;    }
if (Dim1->Button5->Tag == 1) {
    dim = -log(I)/log(h);
    return dim;    }
else return I;
}
//-----
float TPagr1:: CoD( float h, int r )
{float dim; float I=0;
Kiek (h,r);
for (int i=0; i<r; i++)
    for (int j=0; j<r; j++){
        kiek[i][j]=kiek[i][j]/jj;
        if ( kiek[i][j]!=0 ) I=I+kiek[i][j]*kiek[i][j];    }
if (Dim1->Button5->Tag == 1) {
    dim = log(I)/log(h);
    return dim;    }
}

```



```

else return I;
}
//-----
void TPagr1:: Iteracinis_zemelapis(float *xk, float *yk, float zingsnis)
{float pr = xk[0];
float ga = xk[intervalai-1];
int i=0;
jj = (ga-pr)/zingsnis+1;
float *xxx;
xxx = new float[jj];
xx = new float[jj-1];
yy = new float[jj-1];
if (xxx && xx && yy){
while (i<jj/*pr<=ga*/) {
if ((i==0) || (i==jj-1)){
if (i==0) xxx[i] = yk[i];
else xxx[i] = yk[intervalai-1]; }
else xxx[i] = Reiksme_taske( xk, yk, pr, intervalai);
pr = pr + zingsnis;
i++;}
MinMax(xxx, jj);
for (int j=0; j<jj-1; j++){
xx[j]=xxx[j];
yy[j]=xxx[j+1];}
jj=jj-1;
delete xxx;}
}
//-----
void TPagr1:: Kiek( float h, int r )
{float *Ix, *Iy;
Ix = new float[r];
Iy = new float[r];
if (Ix && Iy){
Ix[0]=h1;
for (int i=1; i<r; i++){
Ix[i]=Ix[i-1]+h;
}
Iy[0]=h2;
for (int i=1; i<r; i++) {
Iy[i]=Iy[i-1]-h; }
for (int i=0; i<r; i++)
for (int j=0; j<r; j++)
kiek[i][j]=0;
for (int z=0; z<jj; z++)
for (int i=0; i<r; i++)
for (int j=0; j<r; j++)
if ((xx[z]>=Ix[i]) && (xx[z]<Ix[i+1]) && (yy[z]<=Iy[j]) && (yy[z]>Iy[j+1]))
kiek[i][j]=kiek[i][j]+1;
delete Ix;
delete Iy;}
}
//-----
float TPagr1:: Vidurkis( float *xk, int sk )
{
float vid = 0, suma=0;
for (int i=0; i<sk; i++) suma = suma + xk[i];
vid = suma/sk;
return vid;
}
//-----
float TPagr1:: StandartinisNuokrypis( float *xk, int sk )
{float sumal=0, suma2=0;
for (int i=0; i<sk; i++) {
sumal = sumal + xk[i]*xk[i];
suma2 = suma2 + xk[i];}
float sn = sqrt((sk*sumal - suma2*suma2)/(sk*(sk-1)));
return sn;
}
//-----
void TPagr1:: MinMax( float *xk, int sk )
{min = xk[0];
max = xk[0];
for (int i=1; i<sk; i++) {
if (xk[i] < min) min = xk[i];
if (xk[i] > max) max = xk[i];}
}
//-----
void TPagr1::Valdanti(short ind)
{float xmin, xmax; float *x; float *y; float *yn;
intervalai=0;

```

```

Duomenu_masyvai(0);
y = new float[intervalai];
float *laikinas;
laikinas=Duomenu_masyvai(ind);
for (int i=0; i<intervalai; i++) y[i] = laikinas[i];
if (ind==0){ xmin = 50; xmax = 220; }
else if (ind==1){ xmin = 80; xmax = 240; }
else if (ind==2){ xmin = 20; xmax = 220; }
else if (ind==3){ xmin = 0.2; xmax = 1; }
else if (ind==4){ xmin = 0.15*12; xmax = 0.36*12; }
else if (ind==5){ xmin = 0.2727; xmax = 1.2; }
else if (ind==6){ xmin = 0.3*12; xmax = 0.65*12; }
vid = Vidurkis( y, intervalai );
x = new float[intervalai];
yn = new float[intervalai];
m = new float[intervalai];
for (int i=0; i<intervalai; i++){
    yn[i] =(float) (y[i]-vid)/(xmax-xmin);
    x[i]=i; }
Splainas(x,yn);
Iteracinis_zemelapis(x,yn,0.001 );
float hh = 0.05;//0.1;//0.01;
h1 = floor(min);
h2 = ceil(max);
int r = (float)(h2-h1)/hh+1;
kiek = new float*[r];
for (int j = 0; j < r; j++)
    kiek[j] = new float[r];
Kiek(hh,r);
if (Nustatymail->CheckBox1->Checked) CDn = CD(hh,r);
if (Nustatymail->CheckBox2->Checked) IDn = ID(hh,r );
if (Nustatymail->CheckBox3->Checked) CoDn = CoD(hh,r );
delete yn;
delete x;
delete m;
delete xx;
delete yy;
for (int j = 0; j < r; j++)
    delete[] kiek[j];
delete[] kiek;
delete y;
}
//-----
void TPagrl::ValdantiFrakt(short ind)
{
float xmin, xmax; float *x; float *y; float *yn;
float dim1[4]; float dim2[4]; float dim3[4]; float ep[4];
float dimK1[4]; float dimK2[4]; float dimK3[4];
intervalai=0;
Duomenu_masyvai(0);
y = new float[intervalai];
float *laikinas;
laikinas=Duomenu_masyvai(ind);
for (int i=0; i<intervalai; i++) y[i] = laikinas[i];
if (ind==0){ xmin = 50; xmax = 220; }
else if (ind==1){ xmin = 80; xmax = 240; }
else if (ind==2){ xmin = 20; xmax = 220; }
else if (ind==3){ xmin = 0.2; xmax = 1; }
else if (ind==4){ xmin = 0.15*12; xmax = 0.36*12; }
else if (ind==5){ xmin = 0.2727; xmax = 1.2; }
else if (ind==6){ xmin = 0.3*12; xmax = 0.65*12; }
vid = Vidurkis( y, intervalai );
x = new float[intervalai];
yn = new float[intervalai];
m = new float[intervalai];
for (int i=0; i<intervalai; i++){
    yn[i] =(float) (y[i]-vid)/(xmax-xmin);
    x[i]=i; }
Splainas(x,yn);
for (int i=0; i<4;i++) {
Iteracinis_zemelapis(x,yn,0.001 );
float hh;
if (i==0) hh = 0.005;
else if (i==1) hh = 0.01;
else if (i==2) hh = 0.05;
else if (i==3) hh = 0.1;
ep[i]=log(hh);
h1 = floor(min);
h2 = ceil(max);
int r = (float)(h2-h1)/hh+1;

```



```

        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("HD_RR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("CoD_RR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
        if (Nustatymail->CheckBox9->Checked) {
            Valdanti(6);
            if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("ID_JTRR")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
            if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("HD_JTRR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
            if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("CoD_JTRR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
            DBDM1->IrasaiT->Post();
            DBDM1->IrasaiT->Next(); } }
    else {
        int kiekis = DBDM1->IrasaiT->RecordCount;
        DBDM1->IrasaiT->First();
        for (int j=0; j<kiekis; j++) {
            DBDM1->IrasaiT->Edit();
            if (Nustatymail->CheckBox4->Checked) {
                ValdantiFrakt(1);
                if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_S")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_S")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_S")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
                if (Nustatymail->CheckBox5->Checked) {
                    ValdantiFrakt(2);
                    if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_SD")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_SD")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_SD")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
                if (Nustatymail->CheckBox6->Checked) {
                    ValdantiFrakt(3);
                    if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_SDS")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_SDS")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_SDS")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
                if (Nustatymail->CheckBox7->Checked) {
                    ValdantiFrakt(4);
                    if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_JT")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_JT")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_JT")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
                if (Nustatymail->CheckBox8->Checked) {
                    ValdantiFrakt(5);
                    if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_RR")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_RR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_RR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }

                if (Nustatymail->CheckBox9->Checked) {
                    ValdantiFrakt(6);

                    if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FID_JTRR")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FHD_JTRR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
                    if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_JTRR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
                    DBDM1->IrasaiT->Post();
                    DBDM1->IrasaiT->Next(); } }
    }
}
//-----
void __fastcall TPagr1::Edit1Change(TObject *Sender)
{
    if ( Edit1->Text != ""){
        TLocateOptions Opts;
        Opts.Clear();
        Opts << loCaseInsensitive << loPartialKey;
        if ( DBDM1->PacientaiT->Locate( "PavVard",Edit1->Text, Opts)) { }
    }
}

```

```

    } else DBDM1->PacientaiT->First();
}
//-----
void __fastcall TPagr1::Button10Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount && DBDM1->PacientaiT->RecordCount) {
        Pradzial->Tag = 0;
        Pagr1->Visible = false;
        Filtras1->ShowModal();
        if (Pradzial->Tag != 1) Close();
        Pagr1->Visible = true;
        Pradzial->Tag = 0;    }
}
//-----
void __fastcall TPagr1::Button11Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount) {
        Grafikail->Tag = 1;
        Pradzial->Tag = 0;
        Pagr1->Visible = false;
        Grafikail->ShowModal();
        if (Pradzial->Tag != 1) Close();
        Pagr1->Visible = true;
        Pradzial->Tag = 0;    }
}
//-----
void __fastcall TPagr1::Button12Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount) {
        Grafikail->Tag = 2;
        Pradzial->Tag = 0;
        Pagr1->Visible = false;
        Grafikail->ShowModal();
        if (Pradzial->Tag != 1) Close();
        Pagr1->Visible = true;
        Pradzial->Tag = 0;    }
}
//-----
void __fastcall TPagr1::Button13Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount) {
        Grafikail->Tag = 3;
        Pradzial->Tag = 0;
        Pagr1->Visible = false;
        Grafikail->ShowModal();
        if (Pradzial->Tag != 1) Close();
        Pagr1->Visible = true;
        Pradzial->Tag = 0;    }
}
//-----
void __fastcall TPagr1::Button14Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount) {
        Grafikail->Tag = 4;
        Pradzial->Tag = 0;
        Pagr1->Visible = false;
        Grafikail->ShowModal();
        if (Pradzial->Tag != 1) Close();
        Pagr1->Visible = true;
        Pradzial->Tag = 0;    }
}
//-----
void __fastcall TPagr1::Button1Click(TObject *Sender)
{
    Pacientas = new TPacientas(this);
    Pacientas->Button1->Tag = 1;
    Pacientas->ShowModal();
    delete Pacientas;
    Pacientas = NULL;
}
//-----
void __fastcall TPagr1::Button2Click(TObject *Sender)
{
    if ( DBDM1->PacientaiT->RecordCount > 0){
        Pacientas = new TPacientas(this);
        Pacientas->Button1->Tag = 2;
        Pacientas->ShowModal();
        delete Pacientas;
        Pacientas = NULL;    }
}

```

```
//-----
void __fastcall TPagr1::Button3Click(TObject *Sender)
{
    if (DBDM1->PacientaiT->RecordCount) {
        if (Application->MessageBox ( "Ar tikrai norite ištrinti šį pacientą?"
        ,"Ispėjimas",MB_YESNO+MB_ICONINFORMATION+MB_DEFBUTTON1)==IDYES) {
            trina = true;
            DBDM1->PacientaiT->Delete();
            DBDM1->PacientaiT->Refresh();
            DBDM1->PacientaiT->Close();
            DBDM1->PacientaiT->Open();
            trina= false; } }
    }
//-----
void __fastcall TPagr1::Button6Click(TObject *Sender)
{
    OpenFileDialog->Filter = "Text files (*.txt)|*.TXT";
    trina=true;
    if ( OpenFileDialog->Execute()){
        AnsiString failas = OpenFileDialog->FileName;
        FILE *F; Duom K; AnsiString Pavarde, Vardas, Tdata, Poz; int PacientoNr;

        if (!DBDM1->PacientaiT->RecordCount) PacientoNr=0;
        else {
            PacientoNr = DBDM1->GydytojaiT->FieldByName("Pacientai")->AsInteger; }
        TLocateOptions Opts;
        Opts.Clear();
        Opts << loCaseInsensitive;
        if (FileExists(failas)) {
            F = fopen ( failas.c_str(), "r");
            if ( F ){
                while (!feof(F)) {
                    fscanf ( F, "%d%s\t",&K.Nr,&K.Pavarde);
                    Pavarde = K.Pavarde;
                    fscanf ( F, "%s\t",K.Vardas);
                    Vardas = K.Vardas;
                    fscanf ( F, "%d\t",&K.Tyrimas);
                    fscanf ( F, "%s\t",K.Tdata);
                    Tdata = K.Tdata;
                    fscanf ( F, "%d%d%d%d\t",&K.Lytis,&K.Amzius, &K.Svoris, &K.Ugis, &K.Sv);
                    fscanf ( F, "%s\t",K.Poz);
                    Poz = K.Poz;
                    fscanf ( F,
"%d%d%d%d%d%d%d\t",&K.l_dydis,&K.Nmax,&K.HR_0,&K.HR_50,&K.HR_100,&K.HR_150,&K.HR_200,&K.H
R_250, &K.HR_300,&K.HR_1,&K.HR_2,&K.HR_3,&K.HR_4,&K.HR_5 );
                    fscanf ( F, "%d%d%d%d%d\t",&K.S_0,&K.S_50,&K.S_100,&K.S_150,&K.S_200,&K.S_250,
&K.S_300,&K.S_1,&K.S_2,&K.S_3,&K.S_4,&K.S_5 );
                    fscanf ( F, "%d%d%d%d\t",&K.D_0,&K.D_50,&K.D_100,&K.D_150,&K.D_200,&K.D_250,
&K.D_300,&K.D_1,&K.D_2,&K.D_3,&K.D_4,&K.D_5 );
                    fscanf ( F,
"%f%f%f%f%f%f%f\t",&K.JT_0,&K.JT_50,&K.JT_100,&K.JT_150,&K.JT_200,&K.JT_250,
&K.JT_300,&K.JT_1,&K.JT_2,&K.JT_3,&K.JT_4,&K.JT_5 );
                    fscanf ( F,
"%f%f%f%f%f%f%f\t",&K.ST_0,&K.ST_50,&K.ST_100,&K.ST_150,&K.ST_200,&K.ST_250,
&K.ST_300,&K.ST_1,&K.ST_2,&K.ST_3,&K.ST_4,&K.ST_5 );
                    fscanf ( F,
"%f%f%f%f%f%f%f\n",&K.QRS_0,&K.QRS_50,&K.QRS_100,&K.QRS_150,&K.QRS_200,&K.QRS_250,
&K.QRS_300,&K.QRS_1,&K.QRS_2,&K.QRS_3,&K.QRS_4,&K.QRS_5 );

                if (!DBDM1->PacientaiT->Locate( "Pavarde;Vardas",VarArrayOf (OPENARRAY (Variant, ( Pavarde,
Vardas))), Opts)) {
                    DBDM1->PacientaiT->Append();
                    DBDM1->PacientaiT->FieldByName("GydytojoNumeris")->AsInteger = 1;
                    DBDM1->PacientaiT->FieldByName("PacientoNumeris")->AsString = PacientoNr+1;
                    PacientoNr = DBDM1->PacientaiT->FieldByName("PacientoNumeris")->AsInteger;
                    DBDM1->PacientaiT->FieldByName("Pavarde")->AsString = Pavarde;
                    DBDM1->PacientaiT->FieldByName("Vardas")->AsString = Vardas;
                    DBDM1->PacientaiT->FieldValues["PavVard"] = Pavarde + " " + Vardas;
                    DBDM1->PacientaiT->FieldByName("Lytis")->AsInteger = K.Lytis;
                    DBDM1->PacientaiT->FieldByName("Pozymis")->AsString = Poz;
                    DBDM1->PacientaiT->Post();
                }

                if (!DBDM1->IrasaiT->Locate(
"PacientoNumeris;GydytojoNumeris;Tyrimas;Data",VarArrayOf (OPENARRAY (Variant, ( DBDM1->PacientaiT-
>FieldByName("PacientoNumeris")->AsString, DBDM1->PacientaiT->FieldByName("GydytojoNumeris")->AsString,
K.Tyrimas, StrToDate(Tdata))), Opts)) {
                    DBDM1->IrasaiT->Append();
                    DBDM1->IrasaiT->FieldByName("PacientoNumeris")->AsString = DBDM1->PacientaiT-
>FieldByName("PacientoNumeris")->AsString;

```

```

DBDM1->IrasaiT->FieldByName("GydytojoNumeris")->AsInteger = 1;
DBDM1->IrasaiT->FieldByName("Tyrimas")->AsInteger = K.Tyrimas;
DBDM1->IrasaiT->FieldByName("Data")->AsString = Tdata;
DBDM1->IrasaiT->FieldByName("Lytis")->AsInteger = DBDM1->PacientaiT->FieldByName("Lytis")-
>AsInteger;
DBDM1->IrasaiT->FieldByName("Amzius")->AsInteger = K.Amzius;
DBDM1->IrasaiT->FieldByName("Svoris")->AsInteger = K.Svoris;
DBDM1->IrasaiT->FieldByName("Ugis")->AsInteger = K.Ugis;
DBDM1->IrasaiT->FieldByName("Sv")->AsInteger = K.Sv;
DBDM1->IrasaiT->FieldByName("L_dydis")->AsInteger = K.l_dydis;
DBDM1->IrasaiT->FieldByName("Nmax")->AsInteger = K.Nmax;
DBDM1->IrasaiT->FieldByName("Pozymis")->AsString = Poz;

DBDM1->IrasaiT->FieldByName("HR_0")->AsInteger = K.HR_0;
DBDM1->IrasaiT->FieldByName("HR_50")->AsInteger = K.HR_50;
DBDM1->IrasaiT->FieldByName("HR_100")->AsInteger = K.HR_100;
DBDM1->IrasaiT->FieldByName("HR_150")->AsInteger = K.HR_150;
DBDM1->IrasaiT->FieldByName("HR_200")->AsInteger = K.HR_200;
DBDM1->IrasaiT->FieldByName("HR_250")->AsInteger = K.HR_250;
DBDM1->IrasaiT->FieldByName("HR_300")->AsInteger = K.HR_300;
DBDM1->IrasaiT->FieldByName("HR_1")->AsInteger = K.HR_1;
DBDM1->IrasaiT->FieldByName("HR_2")->AsInteger = K.HR_2;
DBDM1->IrasaiT->FieldByName("HR_3")->AsInteger = K.HR_3;
DBDM1->IrasaiT->FieldByName("HR_4")->AsInteger = K.HR_4;
DBDM1->IrasaiT->FieldByName("HR_5")->AsInteger = K.HR_5;

DBDM1->IrasaiT->FieldByName("S_0")->AsInteger = K.S_0;
DBDM1->IrasaiT->FieldByName("S_50")->AsInteger = K.S_50;
DBDM1->IrasaiT->FieldByName("S_100")->AsInteger = K.S_100;
DBDM1->IrasaiT->FieldByName("S_150")->AsInteger = K.S_150;
DBDM1->IrasaiT->FieldByName("S_200")->AsInteger = K.S_200;
DBDM1->IrasaiT->FieldByName("S_250")->AsInteger = K.S_250;
DBDM1->IrasaiT->FieldByName("S_300")->AsInteger = K.S_300;
DBDM1->IrasaiT->FieldByName("S_1")->AsInteger = K.S_1;
DBDM1->IrasaiT->FieldByName("S_2")->AsInteger = K.S_2;
DBDM1->IrasaiT->FieldByName("S_3")->AsInteger = K.S_3;
DBDM1->IrasaiT->FieldByName("S_4")->AsInteger = K.S_4;
DBDM1->IrasaiT->FieldByName("S_5")->AsInteger = K.S_5;

DBDM1->IrasaiT->FieldByName("D_0")->AsInteger = K.D_0;
DBDM1->IrasaiT->FieldByName("D_50")->AsInteger = K.D_50;
DBDM1->IrasaiT->FieldByName("D_100")->AsInteger = K.D_100;
DBDM1->IrasaiT->FieldByName("D_150")->AsInteger = K.D_150;
DBDM1->IrasaiT->FieldByName("D_200")->AsInteger = K.D_200;
DBDM1->IrasaiT->FieldByName("D_250")->AsInteger = K.D_250;
DBDM1->IrasaiT->FieldByName("D_300")->AsInteger = K.D_300;
DBDM1->IrasaiT->FieldByName("D_1")->AsInteger = K.D_1;
DBDM1->IrasaiT->FieldByName("D_2")->AsInteger = K.D_2;
DBDM1->IrasaiT->FieldByName("D_3")->AsInteger = K.D_3;
DBDM1->IrasaiT->FieldByName("D_4")->AsInteger = K.D_4;
DBDM1->IrasaiT->FieldByName("D_5")->AsInteger = K.D_5;

if (K.JT_0 != 999 ) DBDM1->IrasaiT->FieldByName("JT_0")->AsString =
FloatToStrF(K.JT_0,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_0")->AsString =999;
if (K.JT_50 != 999 ) DBDM1->IrasaiT->FieldByName("JT_50")->AsString =
FloatToStrF(K.JT_50,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_50")->AsString =999;
if (K.JT_100 != 999 ) DBDM1->IrasaiT->FieldByName("JT_100")->AsString =
FloatToStrF(K.JT_100,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_100")->AsString =999;
if (K.JT_150 != 999 ) DBDM1->IrasaiT->FieldByName("JT_150")->AsString =
FloatToStrF(K.JT_150,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_150")->AsString =999;
if (K.JT_200 != 999 ) DBDM1->IrasaiT->FieldByName("JT_200")->AsString =
FloatToStrF(K.JT_200,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_200")->AsString =999;
if (K.JT_250 != 999 ) DBDM1->IrasaiT->FieldByName("JT_250")->AsString =
FloatToStrF(K.JT_250,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_250")->AsString =999;
if (K.JT_300 != 999 ) DBDM1->IrasaiT->FieldByName("JT_300")->AsString =
FloatToStrF(K.JT_300,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_300")->AsString =999;
if (K.JT_1 != 999 ) DBDM1->IrasaiT->FieldByName("JT_1")->AsString =
FloatToStrF(K.JT_1,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_1")->AsString =999;
if (K.JT_2 != 999 ) DBDM1->IrasaiT->FieldByName("JT_2")->AsString =
FloatToStrF(K.JT_2,ffGeneral,5,2);
else DBDM1->IrasaiT->FieldByName("JT_2")->AsString =999;

```





```

        if (K.QRS_5 != 999 )DBDM1->IrasaiT->FieldByName("QRS_5")->AsString =
FloatToStrF(K.QRS_5,ffGeneral,5,2);
        else DBDM1->IrasaiT->FieldByName("QRS_5")->AsString = 999;
        DBDM1->IrasaiT->Post();
        Button17Click(Sender);    }
    }
    fclose (F);    }
    DBDM1->GydytojaiT->Edit();
    DBDM1->GydytojaiT->FieldByName("Pacientai")->AsInteger = PacientoNr;
    DBDM1->GydytojaiT->Post(); }
trina=false;
}
//-----

void __fastcall TPagrl::Button15Click(TObject *Sender)
{
if (DBDM1->PacientaiT->RecordCount && DBDM1->IrasaiT->RecordCount ) {
SaveDialog1->Filter = "Text files (*.txt)|*.TXT";

if ( SaveDialog1->Execute()){

AnsiString failas = SaveDialog1->FileName;
FILE *F;

F = fopen ( failas.c_str(), "w");
if ( F ){
    int kiek = DBDM1->PacientaiT->RecordCount;
    DBDM1->PacientaiT->First();
    for (int i=0; i<kiek; i++) {
        int kiekis = DBDM1->IrasaiT->RecordCount;
        DBDM1->IrasaiT->First();
        for (int j=0; j<kiekis; j++) {

            fprintf ( F, "%d\t%s\t",DBDM1->PacientaiT->FieldByName("PacientoNumeris")->AsInteger,DBDM1->
PacientaiT->FieldByName("Pavarde")->AsString);
            fprintf ( F, "%s\t", DBDM1->PacientaiT->FieldByName("Vardas")->AsString);
            fprintf ( F, "%d\t",DBDM1->IrasaiT->FieldByName("Tyrimas")->AsInteger);
            fprintf ( F, "%s\t",DBDM1->IrasaiT->FieldByName("Data")->AsString);
            fprintf ( F, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t",DBDM1->PacientaiT->FieldByName("Lytis")->AsInteger,DBDM1->
IrasaiT->FieldByName("Amzius")->AsInteger, DBDM1->IrasaiT->FieldByName("Svoris")->AsInteger, DBDM1->
IrasaiT->FieldByName("Ugis")->AsInteger, DBDM1->IrasaiT->FieldByName("Sv")->AsInteger);
            fprintf ( F, "%s\t",DBDM1->IrasaiT->FieldByName("Pozymis")->AsString);
            fprintf ( F, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t",DBDM1->IrasaiT->
FieldByName("L_dydis")->AsInteger,DBDM1->IrasaiT->FieldByName("Nmax")->AsInteger,DBDM1->IrasaiT->
FieldByName("HR_0")->AsInteger,DBDM1->IrasaiT->FieldByName("HR_50")->AsInteger,DBDM1->IrasaiT->
FieldByName("HR_100")->AsInteger,
            DBDM1->IrasaiT->FieldByName("HR_150")->AsInteger,DBDM1->IrasaiT->FieldByName("HR_200")->
AsInteger,DBDM1->IrasaiT->FieldByName("HR_250")->AsInteger, DBDM1->IrasaiT->FieldByName("HR_300")->
AsInteger,DBDM1->IrasaiT->FieldByName("HR_1")->AsInteger,DBDM1->IrasaiT->FieldByName("HR_2")->
AsInteger,
            DBDM1->IrasaiT->FieldByName("HR_3")->AsInteger,DBDM1->IrasaiT->FieldByName("HR_4")->
AsInteger,DBDM1->IrasaiT->FieldByName("HR_5")->AsInteger );
            fprintf ( F, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t",DBDM1->IrasaiT->
FieldByName("S_0")->AsInteger,DBDM1->IrasaiT->FieldByName("S_50")->AsInteger,DBDM1->IrasaiT->
FieldByName("S_100")->AsInteger,
            DBDM1->IrasaiT->FieldByName("S_150")->AsInteger,DBDM1->IrasaiT->FieldByName("S_200")->
AsInteger,DBDM1->IrasaiT->FieldByName("S_250")->AsInteger, DBDM1->IrasaiT->FieldByName("S_300")->
AsInteger,DBDM1->IrasaiT->FieldByName("S_1")->AsInteger,DBDM1->IrasaiT->FieldByName("S_2")->AsInteger,
            DBDM1->IrasaiT->FieldByName("S_3")->AsInteger,DBDM1->IrasaiT->FieldByName("S_4")->
AsInteger,DBDM1->IrasaiT->FieldByName("S_5")->AsInteger );
            fprintf ( F, "%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t%d\t",DBDM1->IrasaiT->
FieldByName("D_0")->AsInteger,DBDM1->IrasaiT->FieldByName("D_50")->AsInteger,DBDM1->IrasaiT->
FieldByName("D_100")->AsInteger,
            DBDM1->IrasaiT->FieldByName("D_150")->AsInteger,DBDM1->IrasaiT->FieldByName("D_200")->
AsInteger,DBDM1->IrasaiT->FieldByName("D_250")->AsInteger, DBDM1->IrasaiT->FieldByName("D_300")->
AsInteger,DBDM1->IrasaiT->FieldByName("D_1")->AsInteger,DBDM1->IrasaiT->FieldByName("D_2")->AsInteger,
            DBDM1->IrasaiT->FieldByName("D_3")->AsInteger,DBDM1->IrasaiT->FieldByName("D_4")->
AsInteger,DBDM1->IrasaiT->FieldByName("D_5")->AsInteger );
            fprintf ( F, "%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t",DBDM1->IrasaiT->
FieldByName("JT_0")->AsFloat,DBDM1->IrasaiT->FieldByName("JT_50")->AsFloat,DBDM1->IrasaiT->
FieldByName("JT_100")->AsFloat,
            DBDM1->IrasaiT->FieldByName("JT_150")->AsFloat,DBDM1->IrasaiT->FieldByName("JT_200")->
AsFloat,DBDM1->IrasaiT->FieldByName("JT_250")->AsFloat, DBDM1->IrasaiT->FieldByName("JT_300")->
AsFloat,DBDM1->IrasaiT->FieldByName("JT_1")->AsFloat,DBDM1->IrasaiT->FieldByName("JT_2")->AsFloat,
            DBDM1->IrasaiT->FieldByName("JT_3")->AsFloat,DBDM1->IrasaiT->FieldByName("JT_4")->
AsFloat,DBDM1->IrasaiT->FieldByName("JT_5")->AsFloat );
            fprintf ( F, "%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t",DBDM1->IrasaiT->
FieldByName("ST_0")->AsFloat,DBDM1->IrasaiT->FieldByName("ST_50")->AsFloat,DBDM1->IrasaiT->
FieldByName("ST_100")->AsFloat,

```

```

        DBDM1->IrasaiT->FieldByName("ST_150")->AsFloat, DBDM1->IrasaiT->FieldByName("ST_200")-
>AsFloat, DBDM1->IrasaiT->FieldByName("ST_250")->AsFloat, DBDM1->IrasaiT->FieldByName("ST_300")-
>AsFloat, DBDM1->IrasaiT->FieldByName("ST_1")->AsFloat, DBDM1->IrasaiT->FieldByName("ST_2")->AsFloat,
        DBDM1->IrasaiT->FieldByName("ST_3")->AsFloat, DBDM1->IrasaiT->FieldByName("ST_4")-
>AsFloat, DBDM1->IrasaiT->FieldByName("ST_5")->AsFloat );
        fprintf ( F, "%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\t%f\n", DBDM1->IrasaiT-
>FieldByName("QRS_0")->AsFloat, DBDM1->IrasaiT->FieldByName("QRS_50")->AsFloat, DBDM1->IrasaiT-
>FieldByName("QRS_100")->AsFloat,
        DBDM1->IrasaiT->FieldByName("QRS_150")->AsFloat, DBDM1->IrasaiT->FieldByName("QRS_200")-
>AsFloat, DBDM1->IrasaiT->FieldByName("QRS_250")->AsFloat, DBDM1->IrasaiT->FieldByName("QRS_300")-
>AsFloat, DBDM1->IrasaiT->FieldByName("QRS_1")->AsFloat, DBDM1->IrasaiT->FieldByName("QRS_2")->AsFloat,
        DBDM1->IrasaiT->FieldByName("QRS_3")->AsFloat, DBDM1->IrasaiT->FieldByName("QRS_4")-
>AsFloat, DBDM1->IrasaiT->FieldByName("QRS_5")->AsFloat );

DBDM1->IrasaiT->Next();
DBDM1->PacientaiT->Next();}

fclose (F);}
}
}
//-----
void TPagr1::Braizyti()
{
if ( DBDM1->IrasaiT->RecordCount ){

Series1->Clear();
Series2->Clear();
Series3->Clear();
Series4->Clear();

intervalai=0;
Duomeni_masyvai(0);

float *laikinas;

laikinas=Duomeni_masyvai(0);
for (int i=0; i<intervalai; i++) Series4->AddXY(i, laikinas[i], "", clTeeColor);

laikinas=Duomeni_masyvai(1);
for (int i=0; i<intervalai; i++) Series1->AddXY(i, laikinas[i], "", clTeeColor);

laikinas=Duomeni_masyvai(7);
for (int i=0; i<intervalai; i++) Series2->AddXY(i, laikinas[i], "", clTeeColor);

laikinas=Duomeni_masyvai(4);
for (int i=0; i<intervalai; i++) Series3->AddXY(i, laikinas[i], "", clTeeColor);
}
else {
Series1->Clear();
Series2->Clear();
Series3->Clear();
Series4->Clear(); }
}
//-----
void __fastcall TPagr1::FormShow(TObject *Sender)
{
TIniFile *ini;
ini = new TIniFile(ExtractFilePath(Application->ExeName)+DBDM1->GydytojaiT-
>FieldByName("GydytojoNumeris")->AsString+".INI" );
P->Height = ini->ReadInteger ("Pagr1", "PacientasA", 300 );
PSesijos->Width = ini->ReadInteger ("Pagr1", "PacientasP", 550 );
Chart2->Height = ini->ReadInteger ("Pagr1", "GrafikaiA", PSesijos->Height/2 );
delete ini;
Braizyti();
}
//-----
void __fastcall TPagr1::Button4Click(TObject *Sender)
{
    NaujK1 = new TNaujK1(this);
    NaujK1->ShowModal();
    delete NaujK1;
    NaujK1= NULL;
}
//-----
void __fastcall TPagr1::Button5Click(TObject *Sender)
{
if (DBDM1->IrasaiT->RecordCount) {
if (Application->MessageBox ( "Ar tikrai norite ištrinti šio tyrimo duomenis?"
, "Ispėjimas", MB_YESNO+MB_ICONINFORMATION+MB_DEFBUTTON1)==IDYES) {
    DBDM1->IrasaiT->Delete(); } }
}
}

```



```

        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_S")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_S")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2);
    }
    if (Nustatymail->CheckBox5->Checked) {
        ValdantiFrakt(2);
        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FID_SD")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_SD")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_SD")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
    if (Nustatymail->CheckBox6->Checked) {
        ValdantiFrakt(3);
        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FID_SDS")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_SDS")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_SDS")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
    if (Nustatymail->CheckBox7->Checked) {
        ValdantiFrakt(4);
        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FID_JT")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_JT")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_JT")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
    if (Nustatymail->CheckBox8->Checked) {
        ValdantiFrakt(5);
        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FID_RR")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_RR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_RR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
    if (Nustatymail->CheckBox9->Checked) {
        ValdantiFrakt(6);
        if (Nustatymail->CheckBox1->Checked) DBDM1->IrasaiT->FieldByName("FID_JTRR")->AsString =
FloatToStrF(IDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox2->Checked) DBDM1->IrasaiT->FieldByName("FHD_JTRR")->AsString =
FloatToStrF(CDn,ffGeneral,5,2);
        if (Nustatymail->CheckBox3->Checked) DBDM1->IrasaiT->FieldByName("FCoD_JTRR")->AsString =
FloatToStrF(CoDn,ffGeneral,5,2); }
        DBDM1->IrasaiT->Post();    }}
//-----
void __fastcall TPagr1::Button19Click(TObject *Sender)
{
    if (DBDM1->IrasaiT->RecordCount) Faz1->ShowModal();
//-----
float TPagr1::MazKv( float dim[], float ep[])
{
    float x2 = 0; float x = 0;float xy = 0; float y = 0; int sk = 4;
    for (int j=0; j<sk; j++) {
        x = x + ep[j];
        y = y + dim[j];
        x2 = x2 + ep[j]*ep[j];
        xy = xy + ep[j]*dim[j];    }
    float k= float(x*y-sk*xy)/float(x*x - sk *x2);
    return k;
}
//-----
float TPagr1::MazKv2( float dim[], float ep[], float *l)
{float x2 = 0; float x = 0; float xy = 0; float y = 0; int sk = 4;
for (int j=0; j<sk; j++) {
    x = x + ep[j];
    y = y + dim[j];
    x2 = x2 + ep[j]*ep[j];
    xy = xy + ep[j]*dim[j];    }
float k= float(x*y-sk*xy)/float(x*x - sk *x2);
*l= float(xy*x-y*x2)/float(x*x- sk*x2);
return k;
}
//-----
void TPagr1::Algoritmas(short ind1, short ind2)
{float xmin, xmax; float *x; float *y; float *yn;
float dim1[4]; float dim2[4]; float dim3[4]; float ep[4];
float dimK1[4]; float dimK2[4]; float dimK3[4];
intervalai=0;

```

```

Duomenu_masyvai(0);
y = new float[intervalai];
float *laikinas;
laikinas=Duomenu_masyvai(ind1);
for (int i=0; i<intervalai; i++) y[i] = laikinas[i];
if (ind1==0){ xmin = 50; xmax = 220; }
else if (ind1==1){ xmin = 80; xmax = 240; }
else if (ind1==2){ xmin = 20; xmax = 220; }
else if (ind1==3){ xmin = 0.2; xmax = 1; }
else if (ind1==4){ xmin = 0.15*12; xmax = 0.36*12; }
else if (ind1==5){ xmin = 0.2727; xmax = 1.2; }
else if (ind1==6){ xmin = 0.3*12; xmax = 0.65*12; }
vid = Vidurkis( y, intervalai );
x = new float[intervalai];
yn = new float[intervalai];
m = new float[intervalai];
for (int i=0; i<intervalai; i++){
    yn[i] =(float) (y[i]-vid)/(xmax-xmin);
    x[i]=i; }
Splainas(x,yn);
for (int i=0; i<4;i++) {
Iteracinis_zemelapis(x,yn,0.001 );
float hh;
if (i==0) hh = 0.005;
else if (i==1) hh = 0.01;
else if (i==2) hh = 0.05;
else if (i==3) hh = 0.1;
ep[i]=log(hh);
h1 = floor(min);
h2 = ceil(max);
int r = (float)(h2-h1)/hh+1;
kiek = new float*[r];
for (int j = 0; j < r; j++)
    kiek[j] = new float[r];
Kiek(hh,r);
if (ind2==0) { CDn = CD(hh,r); dim1[i]=-log(CDn); }
if (ind2==1) { IDn = ID(hh,r ); dim2[i]=IDn; }
if (ind2==2) { CoDn = CoD(hh,r ); dim3[i]=log(CoDn); }
for (int j = 0; j < r; j++) delete[] kiek[j];
delete[] kiek;}
float mm;
if (ind2==0) { CDn = MazKv2(dim1, ep, &mm); Dim1->StaticText69->Caption = FloatToStrF(CDn,
ffFixed,9,4); }
if (ind2==1) { IDn = MazKv2(dim2, ep, &mm); Dim1->StaticText69->Caption = FloatToStrF(IDn,
ffFixed,9,4); }
if (ind2==2) { CoDn = MazKv2(dim3, ep, &mm); Dim1->StaticText69->Caption = FloatToStrF(CoDn,
ffFixed,9,4); }
for (int i=0; i<4; i++) {
if (ind2==0) Dim1->Chart7->Series[1]->AddXY(ep[i],dim1[i],"",clTeeColor);
else if (ind2==1) Dim1->Chart7->Series[1]->AddXY(ep[i],dim2[i],"",clTeeColor);
else if (ind2==2) Dim1->Chart7->Series[1]->AddXY(ep[i],dim3[i],"",clTeeColor); }
for (int i=0; i<4; i++) {
if (ind2==0) Dim1->Chart7->Series[0]->AddXY(ep[i],ep[i]*CDn+mm,"",clTeeColor);
else if (ind2==1) Dim1->Chart7->Series[0]->AddXY(ep[i],ep[i]*IDn+mm,"",clTeeColor);
else if (ind2==2) Dim1->Chart7->Series[0]->AddXY(ep[i],ep[i]*CoDn+mm,"",clTeeColor); }
delete yn; delete x; delete m; delete xx; delete yy; delete y; }
//-----
void __fastcall TPagr1::Button7Click(TObject *Sender)
{
Grup1->ShowModal();}
//-----

```