



KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

Giedrė Jančiukaitė

TEORINIAI IR PRAKTINIAI
FRAKTALINIŲ INTERPOLIACINIŲ
FUNKCIJŲ SUDARYMO ASPEKTAI

Magistro darbas

Vadovas
doc. dr. J. Valantinas

KAUNAS, 2005



KAUNO TECHNOLOGIJOS UNIVERSITETAS
FUNDAMENTALIŲJŲ MOKSLŲ FAKULTETAS
TAIKOMOSIOS MATEMATIKOS KATEDRA

TVIRTINU
Katedros vedėjas
prof. dr. J.Rimas
2005 06 11

TEORINIAI IR PRAKTINIAI
FRAKTALINIŲ INTERPOLIACINIŲ
FUNKCIJŲ SUDARYMO ASPEKTAI

Taikomosios matematikos magistro baigiamasis darbas

Kalbos konsultantas
dr. J. Džežulskienė
2005 05 30

Vadovas
doc. dr. J. Valantinas
2005 06 03

Recenzentas
doc.dr. E.Valakevičius
2005 06 01

Atliko
FMMM 3 gr. stud.
G. Jančiukaitė
2005 05 27

KAUNAS, 2005

KVALIFIKCINĖ 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, profesorius (MII)

Zenonas Navickas, profesorius (KTU)

Arūnas Barauskas, UAB „Elsis“ generalinio direktoriaus pavaduotojas

Jančiukaitė G. Theoretical and practical aspects of fractal interpolation function analysis: Major's work in applied mathematics / supervisor dr. J.Valantinas; Department of Applied mathematics, Faculty of Fundamental Sciences, Kaunas University of Technology. – Kaunas, 2005. – 51 p.

SUMMARY

This thesis introduces fractal interpolation functions, exposes advantages of fractal interpolation of real world objects and presents some newly developed procedures, associated with fractal interpolation process.

The work briefly presents the context needed for introduction of fractal approach and relevant definitions. Also, the detailed description of fractal generating algorithms (deterministic, random iteration, “escape time”) as well as fractal classifications is presented. Since the research object is theoretical and practical aspects of fractal interpolation function analysis, special attention is paid to geometric fractals, obtained using systems of iterated functions (IFS).

The notion of a fractal interpolation function is introduced in the work. The author shows that it is possible to generate fractal interpolation functions for various types of data. The generated functions are “close” (in the sense of Hausdorff dimension) to the data under processing, i.e., it is possible to ensure that the fractal interpolation graph dimension were equal to the fractal dimension of experimental data (graph). The random iteration algorithm is used for the analysis of fractal interpolation functions, since it is relatively simple and fast enough. The author makes an attempt to analyze and solve the problem of choosing interpolation points (general case). A few approaches are proposed, namely the uniform distribution of interpolation points (for the interactive use) and collage. On the other hand, the most valuable developments – procedures for the determination of „skewness“ parameters for the affine transformations, entering the IFS.

The experiment was applied to the real company's trade object. The experimental analysis results showed that the INTERPO software program is the tool which can be put into practice for the generation of fractal interpolation functions for the earlier mentioned type of data (object).

TURINYS

ĮVADAS.....	8
1. BENDROJI DALIS.....	10
1.1. Fraktalai, fraktalų sintezė.....	10
1.1.1. Iteruotųjų funkcijų sistemos ir jų atraktoriai (fraktalai).....	10
1.1.2. Fraktalų sintezės algoritmai.....	12
1.1.3. Fraktalų tipai.....	16
1.2. Fraktalinės dimensijos sąvoka.....	17
1.3. Fraktalinės interpoliacinės funkcijos, jų specifika.....	21
1.4. Kai kurie fraktalų taikymo pavyzdžiai.....	25
2. TIRIAMOJI DALIS.....	28
2.1. INTERPO—fraktalinio interpoliavimo įrankis.....	28
2.2. Interpoliavimo taškų parinkimo kriterijai ir jų nustatymas.....	32
2.2.1. Interaktyviam naudojimui orientuotas tolygus interpoliavimo taškų išdėstymas.....	32
2.2.2. Koliažo organizavimas.....	36
2.3. Interpoliuojamo objekto fraktalinės dimensijos įvertinimas.....	39
2.4. Eksperimento rezultatų analizė.....	41
IŠVADOS.....	50
LITERATŪRA.....	51
1 PRIEDAS INTERPO ALGORITMAS C KALBA.....	52
2 PRIEDAS MATCAD PROGRAMŲ TEKSTAI.....	57

Paveikslų sąrašas

1.1 pav. Determinuotojo IFS atraktoriaus sintezės algoritmo veikimo rezultatas.....	12
1.2 pav. Iteruotųjų funkcijų sistemų atraktoriai (fraktaliniai vaizdai), gauti taikant atsitiktinių iteracijų algoritmą.....	13
1.3 pav. Mandelbroto ir Žulija aibės; $\{C; z^2-\lambda\}$ (centre—Mandelbroto (parametro reikšmių λ , kurias atitinkančios Žulija aibės yra jungios) aibė).....	15
1.4 pav. Van Kocho kreivė (fraktalas).....	17
1.5 pav. Sierpinskio trikampis.....	19
1.6 pav. Dalimis tiesinės interpoliacinės funkcijos, einančios per interpoliavimo taškus, grafikas; šis grafikas sutampa su $IFS\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktoriumi.....	21
2.1 pav. Pradinis INTERPO ekranas – jame siūloma pasirinkti monitoriaus tipą.....	28
2.2 pav. Duomenų aibės (Sample.fip), kai $D=1$, interpoliavimo rezultatas.....	29
2.3 pav. Duomenų aibės pateikimas.....	29
2.4 pav. FIF grafikas; Pvz2a.fip, kai $D=1$	29
2.5 pav. FIF grafikas; Pvz2a.fip, kai $D=1.61$	29
2.6 pav. Diskretizuotos kreivės pavyzdys.....	32
2.7 pav. Pirminis intervalo dalijimas pusiau.....	33
2.8 pav. Naujai dalijamos srities pasirinkimas.....	33
2.9 pav. Koliažo organizavimo ypatumai.....	37
2.10 pav. Įmonės apyvartos kaitos diagrama.....	41
2.11 pav. FIF kreivė „dim1“ duomenims; (Reikšmių sk. 93; skalė 1×900000 ; $D=1$).....	42
2.12 pav. FIF kreivė „dim2“ duomenims; (Reikšmių sk. 93; skalė 1×900000 ; $D=1,2$).....	42
2.13 pav. FIF kreivė „dim3“ duomenims (Reikšmių sk. 93; skalė 1×900000 ; $D=1,4$).....	43
2.14 pav. FIF kreivė „dim4“ duomenims (Reikšmių sk. 93; skalė 1×900000 ; $D=1,6$).....	43
2.15 pav. FIF kreivė „dim5“ duomenims (Reikšmių sk. 93; skalė 1×900000 ; $D=1,8$).....	43
2.16 pav. FIF kreivė „dim6“ duomenims (Reikšmių sk. 93; skalė 1×900000 ; $D=2$).....	44
2.17 pav. FIF kreivė („dim 4“; $\lambda=29$); Reikšmių sk. 61; skalė 1×900000 ; $D=1,6$	44
2.18 pav. FIF kreivė („dim4“; $\lambda=29$; suvidurkintos reikšmės); Reikšmių sk. 61; skalė 1×900000 ; $D=1,6$	45
2.19 pav. FIF kreivė („dim4“; $\lambda=39$; suvidurkintos reikšmės) Reikšmių sk. 45; skalė 1×900000 ; $D=1,6$	46
2.20 pav. Tikrieji ir glodinti duomenys („Mathcad Professional“ funkcija „medsmooth“)......	46
2.21 pav. FIF kreivė („dim4“; $\lambda=29$; „glodinti“ duomenys); Reikšmių sk. 61; skalė 1×900000 ; $D=1,6$	47

2.22 pav. Gauta FIF kreivė („dim4“; $\lambda=39$; „glodinti“ duomenys); Reikšmių sk. 45; skalė 1x900000; D=1,6.....	47
2.23 pav. FIF kreivė („dim4“; $\lambda=49$; „glodinti“ duomenys); Reikšmių sk. 35; skalė 1x900000; D=1,6.....	48

IVADAS

Sąvokos „fraktalas“ ir „fraktalinė geometrija“ atsirado aštuntojo dešimtmečio pabaigoje, o nuo devintojo dešimtmečio vidurio jau ilgam įsitvirtino matematikų ir programuotojų žodyne. Terminas „fraktalas“ paimtas iš lotynų kalbos: „fractus“ reiškia nereguliarų fragmentą. Beje, šį terminą 1975 m. pasiūlė amerikiečių matematikas Benua Mandelbrotas (Benoit Mandelbrot) nereguliarims, bet panašioms į save, struktūroms pažymėti. Fraktalinės geometrijos gimimas siejamas su 1977 m. pasirodžiusia B. Mandelbroto knyga „Fraktalinė gamtos geometrija“ [7]. Knygoje panaudoti ir kitų mokslininkų, dirbusių šioje srityje 1875—1925 metų laikotarpiu (Dž.Kantoro (Georg Cantor), Dž.Peano (Giuseppe Peano), D.Hilberto (David Hilbert), H.Kocho (Helge von Koch), V.Sierpinskio (Waclaw Sierpinski), G.Žulia (Gaston Julia), F.Hausdorfo (Felix Hausdorff), A.Bezicovičiaus (Abram Besicovitch), P.Fatu (Pierre Fatou) mokslinių tyrimų rezultatai. Svarbu tai, kad tik pastaruoju metu (B. Mandelbroto dėka) visi tie darbai buvo apjungti į vientisą sistemą.

Visos figūros, kurias B.Mandelbrotas tyrė ir vadino fraktalais, turėjo savybę – nereguliaros, bet atsikartojančios savyje, t.y. žiūrint pro padidinamąjį stiklą į atskirą figūros fragmentą, pastarasis „primindavo“ visą figūrą. Be to, fraktalus (kaip aibes) sudaro be galo daug taškų, kurių tarpusavio išsidėstymas toks sudėtingas, kad neįmanoma nei perprasti, nei aprašyti jų „geometrijos“.

Fraktalų įvairovę patogiu pateikti, atsižvelgiant į visuotinai priimtą jų klasifikaciją, būtent: geometriniai fraktalai, algebriniai fraktalai, stochastiniai fraktalai.

Geometriniai fraktalai vaizdžiausi. Dvimačiu atveju, jie gaunami kokios tai laužtės (arba paviršiaus, trimačiu atveju), vadinamos generatoriumi, pagalba. Kiekviename algoritmo žingsnyje, laužtę sudaranti atkarpa keičiama laužte—generatoriumi (atitinkamai parinkus mastelį). Daugkartinio šios procedūros taikymo rezultatas—geometrinis fraktalas.

Kompiuterinėje grafikoje geometriniai fraktalai yra būtina priemonė medžių, krūmų, kalnų (kranto) linijos ir kitiems vaizdams sukurti. Dvimačiai geometriniai fraktalai naudojami erdvinėms tekstūroms (objektų paviršiaus piešiniam) išgauti.

Algebriniai fraktalai—tai pati didžiausia fraktalų grupė. Jie generuojami (gaunami) netiesinių iteracinių procesų n —matėse erdvėse pagalba. Geriausiai išanalizuoti yra dvimačiai iteraciniai procesai. Interpretuojant tokių netiesinių iteracinių procesų kaip diskrečiąją dinaminę sistemą, galime naudotis šių sistemų teorijos terminija: fazinis portretas, nusistovėjęs procesas, atraktorius ir panašiai.

Žinoma, jog netiesinės dinaminės sistemos turi kelias stabilias būsenas (padėtis). Būsena, kurioje atsidūrė dinaminė sistema po baigtinio iteracijų skaičiaus, priklauso nuo jos pradinės padėties. Todėl kiekviena stabili būsena (dar vadinama atraktoriumi) turi fiksuotą aibę pradinių padėčių, iš kurių

sistema būtinai pereina į analizuojamą galutinę stabilią būseną. Tokiu būdu, fazinė sistemos erdvė gali būti suskaidyta į atraktorių pritraukimo zonas. Jei fazinė erdvė—dvimatė erdvė, tai, parinkdami pritraukimo zonoms skirtingas spalvas, galime nesunkiai gauti spalvinį fazinį šios sistemos (iteracinio proceso) portretą.

Dar vieną fraktalų klasę sudaro stochastiniai fraktalai. Jie gaunami, iteracinio proceso metu atsitiktinai keičiant vienus ar kitus proceso parametrus. Generuojami objektai labai panašūs į gamtinius objektus—nesimetrinius medžius, išraižytas kranto linijas ir kita. Dvimačiai stochastiniai fraktalai, paprastai, naudojami modeliuojant vietovės reljefą bei jūros paviršių.

Daugumos ženklių mokslo apie fraktalus laimėjimų nebūtų buvę įmanoma pasiekti be skaičiavimo matematikos priemonių, kurios tiesiogiai remiasi šiuolaikinių kompiuterių taikymu. Milžiniški kompiuterių skaičiavimo pajėgumai leido pakankamai išsamiai apžvelgti įvairias fraktalines struktūras bei suprasti jų atsiradimo priežastis. Kita vertus, gana dažnai teorinis šių struktūrų modeliavimas aplenkė eksperimentinius metodus, kuriais tiriami realūs sudėtingos formos gamtiniai objektai.

Fraktalinės geometrijos srityje taip pat daug yra nuveikę Klifordas A. Pikoveris (Clifford A. Pickover), Džeimsas Gleikas (James Gleick), G. O. Peitgenas (H. O. Peitgen) bei kiti. Šie mokslininkai stengėsi išplėsti naująją geometriją taip, kad ją būtų galima taikyti nuo vertybinių popierių biržos prognozių iki naujų atradimų teorinės fizikos srityje. JAV fraktalinių modelių tyrimu užsiima specialus centras NCSA (Nacionalinis papildymų superkompiuteriams centras).

Šiame darbe pristatomos ir analizuojamos fraktalinės interpoliacinės funkcijos, atskleidžiami realaus pasaulio objektų fraktalinio interpoliavimo privalumai, pateikiamos paties interpoliacinio proceso organizavimo procedūros.

1. BENDROJI DALIS

Fraktalinė geometrija — tai šiuolaikinės matematikos šaka, apjungianti klasikinę (Euklido) geometriją, topologiją, mato bei dinaminų sistemų teorijas. Benua Mandelbrotas — fraktalų geometrijos pradininkas — šią geometriją pavadino gamtos geometrija [7]. Būtent jisai surado nišą įvairiems „patologiniams“ (klasikinei matematinei analizei „nepaklūstantiems“) objektams bei reiškiniams (Kantoro dulkės, Peano kreivės, Vejerštraso funkcijos ir kt.). Būtent B. Mandelbroto dėka fraktalinė geometrija tapo taikomuoju mokslu. Fraktalų panaudojimo sritys nuolat plečiasi. Jeigu prieš dešimt metų didžiausias dėmesys buvo skiriamas realaus pasaulio objektų (debesys, kalnai, jūros paviršius ir pan.) modeliavimui, tai dabar fraktalinės geometrijos, kaip matematinio įrankio, taikymo sričių spektras kur kas platesnis. Tai duomenų suglaudėjimas (kompiuterinė grafika), techninė kainų analizės teorija (ekonomika), kietųjų kūnų paviršių analizė ir sintezė (fizika), biosensorinių tarpusavio poveikių tyrimas (medicina) ir t. t.

Žemiau glaustai pristatomas kontekstas, reikalingas fraktalo sąvokos įvedimui, pateikiami pakankamai detalūs fraktalų sintezės (generavimo) algoritmų aprašai bei komentuojami fraktalų tipai.

1.1. Fraktalai, fraktalų sintezė

Kadangi tiriamojo darbo objektas yra teoriniai ir praktiniai fraktalinių interpoliacinių funkcijų sudarymo aspektai, tai pagrindinį dėmesį toliau skirsime geometriniams fraktalams, gaunamiems taikant iteruotąsias (afiniųjų) funkcijų sistemas.

1.1.1. Iteruotųjų funkcijų sistemos ir jų atraktoriai (fraktalai)

Tarkime, turime metrinę erdvę (X, d) ir joje apibrėžtą transformaciją $f: X \rightarrow X$. Pastebėsime, jog $f(S) = \{f(x) \mid x \in S\}$, kai $S \subset X$. Transformacija f yra apgręžiama, jeigu ji yra abipusiškai vienareikšmė ir $f(X) = X$. Šiuo atveju, galima apibrėžti atvirkštinę transformaciją $f^{-1}: X \rightarrow X$ tokią, kad $f^{-1}(y) = x$, ir $x \in X$ yra vienintelis taškas, su kuriuo $f(x) = y$.

Transformacijos $f: X \rightarrow X$ iteracijomis pirmyn vadinamos transformacijos $f^{0n}: X \rightarrow X$, apibrėžiamos lygybėmis:

$$f^{00}(x) = x, \quad f^{01}(x) = f(x), \quad f^{0(n+1)}(x) = f(f^{0n}(x)), \quad \text{su visais } n = 0, 1, 2, \dots$$

Jeigu f yra apgręžiama, tai transformacijos f iteracijomis atgal vadinamos transformacijos $f^{0(-m)}: X \rightarrow X$, apibrėžiamos lygybėmis:

$$f^{0(-1)}(x) = f^{-1}(x), \quad f^{0(-m)}(x) = (f^{0m})^{-1}(x), \quad \text{su visais } m = 1, 2, 3, \dots$$

Praktiniuose taikymuose, svarbiausia yra akcentuoti ryšį tarp transformacijos apibūdinančių formulių ir jų poveikyje atsirandančių geometrinių pasikeitimų (ištempimų, poslinkių, lenkimų ir pan.) erdvėse. Be to, svarbu tai, kaip transformacijos veikia ne atskirus erdvės X taškus, o jos poaibius.

Kalbant apie geometrinius fraktalus, jų prigimtį, paprastai, imamos Euklido metrinės erdvės – (\mathbb{R}, d) , (\mathbb{R}^2, d) arba (\mathbb{R}^3, d) – ir joje veikiančios afiniosios transformacijos.

Bendru atveju, afinioji transformacija, veikianti (tarkime, dvimatėje) Euklido erdvėje (\mathbb{R}^2, d) , užrašoma taip - $\omega: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, ir apibrėžiama lygybe - $\omega(x) = \omega(x_1, x_2) = (ax_1 + bx_2 + e, cx_1 + dx_2 + f)$, su visais $x = (x_1, x_2) \in \mathbb{R}^2$; čia a, b, c, d, e, f yra realieji skaičiai (afiniosios transformacijos ω parametrai). Afinosios transformacijos turi daug svarbių geometrinių ir algebrinių savybių. Jų pagalba galima realizuoti posūkio, atspindžio, panašumo, pražulniąsias ir kitas transformacijas Euklido erdvėje (\mathbb{R}^2, d) , [14].

Jeigu su visais $x, y \in \mathbb{R}^2$ teisinga nelygybė $d(\omega(x), \omega(y)) \leq s \cdot d(x, y)$, $0 \leq s < 1$, tai afinioji transformacija $\omega: \mathbb{R}^2 \rightarrow \mathbb{R}^2$ vadinama suspaudžiančiąja, o realusis skaičius s , tenkinantis šią sąlygą, — afiniosios transformacijos suspaudimo koeficientu.

Imkime afinijų suspaudžiančių transformacijų $\omega_i: \mathbb{R}^2 \rightarrow \mathbb{R}^2$, $i = 1, 2, \dots, N$, rinkinį atskirų afinijų transformacijų suspaudimo koeficientus pažymėkime s_i , $i = 1, 2, \dots, N$. Tada, Euklido erdvė (\mathbb{R}^2, d) su joje veikiančių suspaudžiančių afinijų transformacijų rinkiniu vadinama iteruotųjų funkcijų sistema ir žymima - $\text{IFS}\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$. IFS suspaudimo koeficientu laikomas skaičius $s = \max\{s_1, s_2, \dots, s_N\}$.

Apibrėžkime dar vieną metrinę erdvę $(H(\mathbb{R}^2), h)$ tokiu būdu: $H(\mathbb{R}^2)$ - visų netuščių uždarujų aibės \mathbb{R}^2 poaibių aibė; h — metrika, nusakanti atstumą tarp bet kurių dviejų aibės $H(\mathbb{R}^2)$ elementų (aibės \mathbb{R}^2 poaibių), būtent:

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

$$\text{čia } d(A, B) = \max_{x \in A} \{ \min_{y \in B} \{d(x, y)\} \}; \quad d(B, A) = \max_{y \in A} \{ \min_{x \in B} \{d(y, x)\} \}.$$

Įvestoji erdvė $(H(\mathbb{R}^2), h)$ vadinama „fraktalu“ erdve arba tiesiog, fraktaline erdve.

Perkelkime IFS sudarančias afiniąsias transformacijas į erdvę $(H(\mathbb{R}^2), h)$. Tada $\omega_i: H(\mathbb{R}^2) \rightarrow H(\mathbb{R}^2)$, apibrėžiama lygybe $\omega_i(B) = \{\omega_i(y) | y \in B\}, \forall B \in H(\mathbb{R}^2)$, yra suspaudžiančioji transformacija erdvėje $(H(\mathbb{R}^2), h)$, ir jos suspaudimo koeficientas s_i , $i = 1, 2, \dots, N$.

Apibrėžkime dar vieną transformaciją fraktalinėje erdvėje $(H(\mathbb{R}^2), h)$, būtent:

$W : H(\mathbb{R}^2) \rightarrow H(\mathbb{R}^2)$; čia $W(B) = \omega_1(B) \cup \omega_2(B) \cup \dots \cup \omega_N(B) = \bigcup_{i=1}^N \omega_i(B)$, $\forall B \in H(\mathbb{R}^2)$

Galima įsitikinti, jog pastaroji transformacija irgi yra suspaudžiančioji, t.y. $h(W(B), W(C)) \leq s \cdot h(B, C)$, su visais $B, C \in H(\mathbb{R}^2)$; be to, $s = \max\{s_1, s_2, \dots, s_N\}$.

Vienintelis nejudamasis transformacijos $W : H(\mathbb{R}^2) \rightarrow H(\mathbb{R}^2)$ taškas A ($A \in H(\mathbb{R}^2)$) toks, kad $A = W(A) = \bigcup_{i=1}^N \omega_i(A) = \lim_{n \rightarrow \infty} W^{0n}(B)$, $\forall B \in H(\mathbb{R}^2)$, vadinamas IFS atraktoriumi (arba fraktalu), [14]. Daug iteruotųjų funkcijų sistemų bei jų atraktorių pavyzdžių galima rasti literatūroje, [14].

1.1.2. Fraktalų sintezės algoritmai

Yra žinomi ir plačiau praktikoje taikomi keli IFS atraktorių sintezės (generavimo) algoritmai. Tai – determinuotasis algoritmas, atsitiktinių iteracijų algoritmas bei „pabėgimo laiko“ algoritmas.

Žemiau pateikiami detalesni šių algoritmų aprašai.

Determinuotojo fraktalų (IFS atraktorių) generavimo algoritmo „veikimas“ tiesiogiai remiasi IFS atraktoriaus apibrėžimu.

Tarkime, kad $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ yra iteruotųjų funkcijų sistema (IFS); čia ω_i ($i = 1, 2, \dots, N$) yra suspaudžiančiosios afiniosios transformacijos. Parenkame (pradinę) uždaroją aibę $A_0 \subset \mathbb{R}^2$ ($A_0 \in H(\mathbb{R}^2)$) ir nuosekliai formuojame aibių seką $\{A_n\}$; čia:

$$A_n = W^{0n}(A_0) = W(A_{n-1}) = \bigcup_{i=1}^N \omega_i(A_{n-1}), \text{ su visais } n = 1, 2, \dots. \text{ Įrodoma, jog tai (Koši) seka, kuri}$$

konverguoja į IFS atraktorių A . Kai n yra pakankamai didelis skaičius, aibė A_n vizualiai yra „artima“ aibei (fraktalui, IFS atraktoriui) A .

Visiškai nesvarbu, kokia imama pradinė uždaroji aibė A_0 . Iteracinės procedūros rezultatas yra vienas ir tas pats — IFS atraktorius A . Kitaip sakant, aibė A pilnai nusakoma afinių transformacijų $\omega_i, i = 1, 2, \dots, N$, veikiančių fraktalinėje erdvėje $(H(\mathbb{R}^2), h)$, išraiškomis (2.1 pav.).

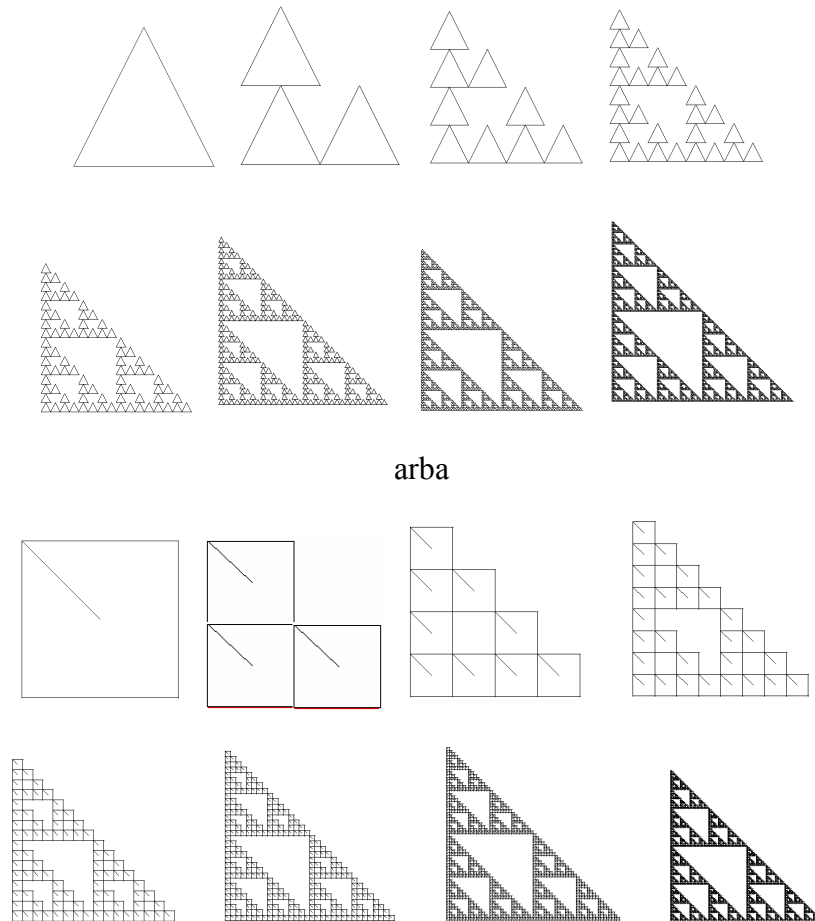
Pateiksime atsitiktinių iteracijų algoritmo aprašą. Tarkime, kad $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ yra iteruotųjų funkcijų sistema. Kiekvienai suspaudžiančiajai afiniajai transformacijai ω_i ($i = 1, 2, \dots, N$) yra priskiriamas teigiamas skaičius (tikimybė) p_i taip, kad $p_1 + p_2 + \dots + p_n = 1$. Jeigu

$$\omega_i = \omega_i \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} = A_i \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + T_i, i = 1, 2, \dots, N, \text{ tai apytikslės tikimybių } p_i \text{ reikšmės}$$

randamos iš formulės

$$p_i \cong \frac{|a_i d_i - b_i c_i|}{\sum_{j=1}^N |a_j d_j - b_j c_j|};$$

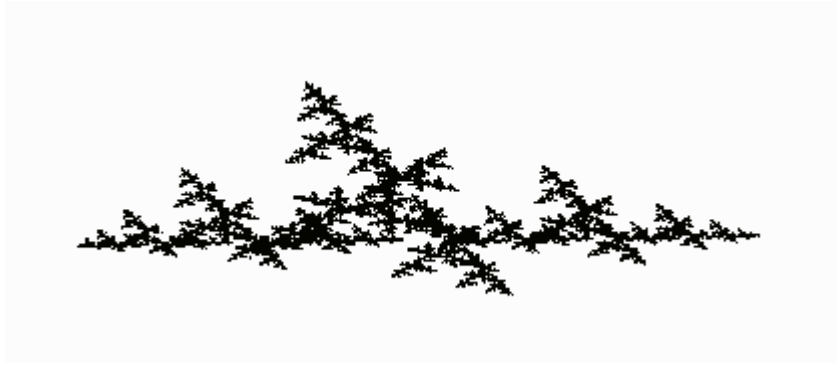
Jeigu su kuria nors reikšme i , $a_i d_i - b_i c_i = 0$, tai p_i yra pakankamai mažas teigiamas skaičius (tarkim, $p_i = 0.001$).



1.1 pav. Determinuotojo IFS atraktoriaus sintezės algoritmo veikimo rezultatas

Parenkamas pradinis taškas x_0 ($x_0 \in \mathbb{R}^2$). Rekursyviai ir nepriklausomai konstruojama erdvės \mathbb{R}^2 taškų seka $\{x_n\}_{n=0}^{\infty}$; čia $x_n \in \{\omega_1(x_{n-1}), \omega_2(x_{n-1}), \dots, \omega_N(x_{n-1})\}$, su visais $n = 1, 2, \dots$, ir įvykio, jog x_n įgys reikšmę $\omega_i(x_{n-1})$, tikimybė yra lygi p_i , t. y. $P\{x_n = \omega_i(x_{n-1})\} = p_i$, $i = 1, 2, \dots, N$.

Atsitiktinių iteracijų algoritmo „veikimas“ yra grindžiamas fraktalo (IFS atraktoriaus) taškams būdinga chaotiška judesio dinamika. Keletas IFS kodų bei juos atitinkančių fraktalinių vaizdų parodyti 1.2 pav.



(a) IFS $\{X, \omega_1, \omega_2, \omega_3\}$ atraktorius, gautas taikant atsitiktinių iteracijų algoritmą; IFS kodas:

$$\begin{aligned} &\{0.48 \ 0.01 \ -0.01 \ 0.47 \ 0.80 \ -0.34 \ 0.26 \\ &0.47 \ 0.01 \ -0.01 \ 0.47 \ -1.01 \ -0.28 \ 0.25 \\ &-0.28 \ 0.38 \ -0.96 \ -0.19 \ 0.03 \ -0.06 \ 0.48\} \end{aligned}$$


(b) IFS $\{X, \omega_1, \omega_2\}$ atraktorius, gautas taikant atsitiktinių iteracijų algoritmą; IFS kodas:

$$\begin{aligned} &\{-0.46 \ -0.50 \ 0.58 \ -0.58 \ 0.46 \ -0.24 \ 0.60 \\ &0.52 \ -0.45 \ 0.45 \ 0.32 \ -0.85 \ 0.43 \ 0.40\} \end{aligned}$$

1.2 pav. Iteruotųjų funkcijų sistemų atraktoriai (fraktaliniai vaizdai), gauti taikant atsitiktinių iteracijų algoritmą

„Pabėgimo laiko“ algoritmas žinomas daugiau kaip teorinis rezultatas (idėja), jo praktinė realizacija iki šiol dar kelia tam tikrų problemų.

Pateikiame pačią idėją (algoritmą).

Tarkime, kad $\text{IFS}\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktorius yra aibė A (akivaizdu, $A = \omega_1(A) \cup \omega_2(A) \cup \dots \cup \omega_N(A)$). Tarkime, taipogi, jog visos afinosios transformacijos yra abipusiškai vienareikšmės, t. y. apgrėžiamos.

Duotajai IFS konstruojama dinaminė poslinkių sistema $\{A; S\}$, kai poslinkio transformacija $S: A \rightarrow A$ apibrėžiama (kiekvienam taškui $a \in A$) taip: $S(a) = \omega_i^{-1}(a)$, jeigu $a \in \omega_i(A)$ ($i \in \{1, 2, \dots, N\}$) ir $a \notin \bigcup_{j=1(j \neq i)}^N \omega_j(A)$; sakoma, jog taškas a yra transformacijos ω_i^{-1} veikimo

zonoje; $S(a) = \omega_i^{-1}(a)$, jeigu $a \in \bigcap_{s=1}^r \omega_{i_s}(A)$; čia $t \in \{i_1, i_2, \dots, i_r\} \subset \{1, 2, \dots, N\}$; sakoma, jog taškas a yra bet kurios iš transformacijų ω_i^{-1} veikimo zonoje.

Su duotąja IFS susijusi dinaminė poslinkių sistema $\{A; S\}$ praplečiama į visą erdvę \mathbf{R}^2 , t. y. konstruojama nauja dinaminė sistema $\{\mathbf{R}^2; \hat{S}\}$. Naujoji poslinkio transformacija $\hat{S}: \mathbf{R}^2 \rightarrow \mathbf{R}^2$ apibrėžiama lygybe $\hat{S}(x) = \hat{S}(x) = \omega_i^{-1}(x)$, kai taškas $x \in \mathbf{R}^2$ patenka į i -tosios atvirkštinės afiniosios transformacijos ω_i^{-1} ($i \in \{1, 2, \dots, N\}$) veikimo zoną. Aišku, kad \hat{S} sutampa su S , kai taškas $x \in A$.

Toliau, tarkime, kad IFS atraktorius A patenka į stačiakampį R , t. y. $A \subset R \subset \mathbf{R}^2$. Kiekvienam stačiakampio R taškui x konstruojama orbita $\{\hat{S}^{0n}(x)\}_{n=1}^{\tau}$; čia τ —iš anksto apibrėžtas iteracijų skaičius. Fiksuokime skritulį su centru stačiakampio įstrižainių susikirtimo taške (pažymėkime O) ir spinduliu R ; beje, skritulys talpina savyje stačiakampį R .

Jeigu $d(\hat{S}^{0\tau}, O) \leq R$, t. y. po τ iteracijų taško $x \in R$ orbita nepalieka skritulio, daroma išvada, jog taškas x priklauso IFS atraktoriui, t. y. $x \in A$; priešingu atveju $d(\hat{S}^{0\tau}, O) > R$, $x \notin A$.

„Pabėgimo laiko“ algoritmo „veikimas“ remiasi tuo, jog IFS atraktorius (aibė A) yra transformacijos $\hat{S}: H(\mathbf{R}^2) \rightarrow H(\mathbf{R}^2)$ atstumiantysis nejudamasis taškas. Kitaip tariant, taškų, esančių arčiau (metrikos h prasme) atraktoriaus A , orbitos ilgiau „užsibūna“ skritulyje, negu orbitos taškų, labiau nutolusių nuo atraktoriaus.

Pagrindinė kliūtis, dėl kurios šis algoritmas sunkokai pritaikomas praktikoje, — tai nebuvimas afinių transformacijų veikimo zonų atskyrimo kriterijaus.

Praktiniuose taikymuose ypatingas vaidmuo tenka Koliažo teoremai (M.Barnsley, 1985), kuri čia pateikiama be įrodymo.

Teorema / Koliažo /. Tarkime, kad (X, d) yra pilnoji metrinė erdvė, $T \in H(X)$ yra duotoji aibė ir $\varepsilon \geq 0$. Parinkime IFS (arba IFS su kondensacija) $\{X; (\omega_0), \omega_1, \omega_2, \dots, \omega_N\}$; kurios suspaudimo koeficientas s ($0 \leq s < 1$), taip, kad

$$h\left(T, \bigcup_{i=1}^N \omega_i(T)\right) \leq \varepsilon.$$

$$\text{Tada } h(T, A) \leq \frac{1}{1-s} \cdot h\left(T, \bigcup_{\substack{i=1 \\ (i=0)}}^N \omega_i(T)\right) \leq \frac{\varepsilon}{1-s}; \text{ čia } A \text{ yra IFS atraktorius}$$

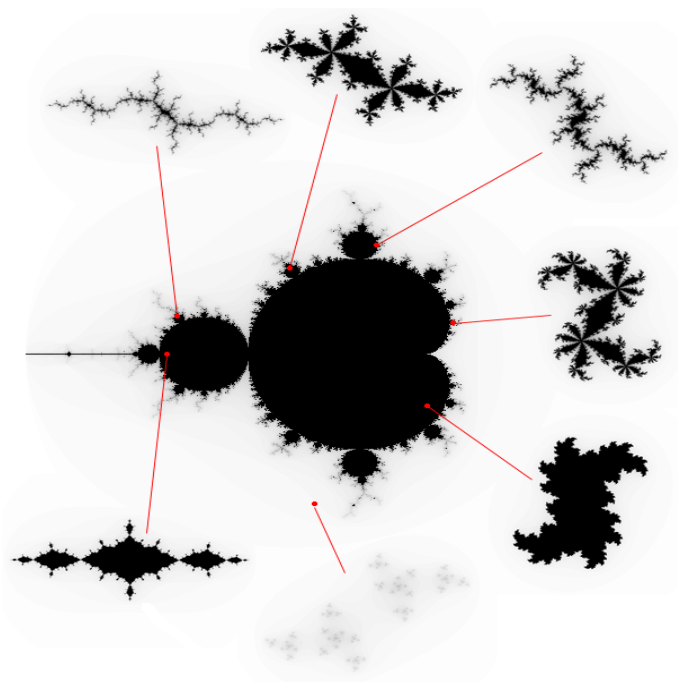
Šiame darbe, fraktalinių interpoliacinių funkcijų analizei mes naudojome atsitiktinių iteracijų algoritmą. Būtent šis algoritmas pasižymi spartumu bei nebloga sintezuojamų fraktalinių vaizdų kokybe

1.1.3. Fraktalų tipai

Fraktalinėje geometrijoje išskiriami šie fraktalų tipai — geometriniai fraktalai, algebriniai fraktalai, stochastiniai fraktalai.

Geometriniai fraktalai siejami su ankstyvuoju fraktalinės geometrijos (kaip mokslo šakos) vystymosi periodu. Faktiškai, tai iteruotųjų funkcijų sistemų, charakterizuojamų afiniųjų transformacijų, veikiančių įvairaus matavimo Euklido erdvėse, rinkiniais, atraktorių aibė. Šio tipo fraktalai yra vaizdžiausi ir, galima sakyti, geriausiai išanalizuoti.

Algebriniai fraktalai priklauso pačiai didžiausiai fraktalų grupei. Jie siejami su netiesinėmis dinaminėmis sistemomis, veikiančiomis kompleksinėje plokštumoje. Šio tipo fraktalų sintezei gana efektyviai panaudojamas anksčiau aptartas „pabėgimo laiko“ algoritmas (1.1.2 skyrelis). Vienas ryškiausių šio tipo fraktalų yra Mandelbroto aibė ir su ja susijusios Žulija aibės (1.3 pav.). Pastarosios generuojamos, tiriant netiesinės parametrizuotos sistemos $\{C; f_\lambda\}$, $f_\lambda(z) = z^2 - \lambda$, dinamiką. Čia pat pastebėsime, jog algebriniai fraktalai nėra pilnai ištirti, vis dar slepia savyje daug paslapčių.



1.3 pav. Mandelbroto ir Žulija aibės; $\{C; z^2 - \lambda\}$ (centre—Mandelbroto (parametro reikšmių λ , kurias atitinkančios Žulija aibės yra jungios) aibė)

Stochastinius fraktalus, tam tikra prasme, galima būtų priskirti geometriniais fraktalams. Skirtumas tas, jog vietoj įprastinių iteruotųjų funkcijų sistemų imamos parametrizuotos sistemos. Iteracinio proceso metu keičiamos parametrų reikšmės, ir tai daroma atsitiktinai (atsitiktine tvarka). Rezultatas — gaunami fraktalai labai panašūs į realaus pasaulio objektus („asimetriniai“ medžiai,

karpytos pakrantės linijos ir pan.). Dvimačiai stochastiniai fraktalai naudojami, modeliuojant vietovės reljefą, jūros paviršių ir kt.

1.2. Fraktalinės dimensijos sąvoka

Skaičiai, charakterizuojantys fraktalus ir skirti pastarųjų palyginimui, paprastai, vadinami fraktalinėmis dimensijomis. Tai gana svarbios kiekybinės fraktalų charakteristikos. Jos leidžia įvertinti subjektyvų požiūrį apie tai, kaip „tirštai“ (tankiai) realaus pasaulio objektas (fraktalas) užpildo erdvę, kurios poaibis jis pats yra. Fraktalinė dimensija—tai objektyvi priemonė fraktalų palyginimui.

Istoriškai, svarbu apibrėžti Hausdorfo ir Bezicovičiaus matą bei dimensiją, ir kartu atskleisti su jų apskaičiavimu susijusius sunkumus, [14].

Imkime poaibį $A \subset \mathbb{R}^n$; čia (\mathbb{R}^n, d) - n -matė Euklido erdvė. Tegu $|A| = \text{diam}(A) = \sup\{d(x, y) \mid x, y \in A\}$ žymi poaibio A diametrą.

Fiksuokime teigiamą skaičių $\varepsilon > 0$. Imkime skaičių poaibių A_i ($|A_i| \leq \varepsilon$) aibę $\{A_i\}$ ir tokią, kad ji padengtų poaibį A , t.y. $A \subset \bigcup_{i=1}^{\infty} A_i$; rinkinys (aibė) $\{A_i\}$ vadinamas poaibio A ε -denginiu.

Toliau, tarkime, kad $p > 0$, ir apibrėžkime dydį $H_\varepsilon^p(A)$ tokiu būdu:

$$H_\varepsilon^p(A) = \inf \left\{ \sum_{i=1}^{\infty} |A_i|^p \mid \{A_i\} \text{ yra poaibio } A \text{ } \varepsilon\text{-denginys} \right\}.$$

Faktiškai, tai minimizavimo uždavinys, kai stengiamasi minimizuoti dengiančiųjų aibių diametrų, pakeltų p —tuoju laipsniu, sumą (pagal visus įmanomus poaibio A ε -denginius). Kita vertus, dydis $H_\varepsilon^p(A)$, kaip funkcija, yra nemažėjantis, kai ε artėja prie nulio. Pereidami prie ribos, kai $\varepsilon \rightarrow 0$, apibrėšime Hausdorfo ir Bezicovičiaus matą (poaibiui A) tokiu būdu

$$H^p(A) = \lim_{\varepsilon \rightarrow 0} H_\varepsilon^p(A).$$

Dydžio $H^p(A)$ priklausomybės nuo parametro p grafikas rodo, jog yra kritinė p reikšmė, kai $H^p(A)$ „šoka“ nuo ∞ prie 0, [14]. Tą kritinį tašką atitinkanti p reikšmė vadinama Hausdorfo ir Bezicovičiaus dimensija (poaibiui A), t.y.

$$D_H(A) = \sup\{p \mid H^p(A) = \infty\} = \inf\{p \mid H^p(A) = 0\}.$$

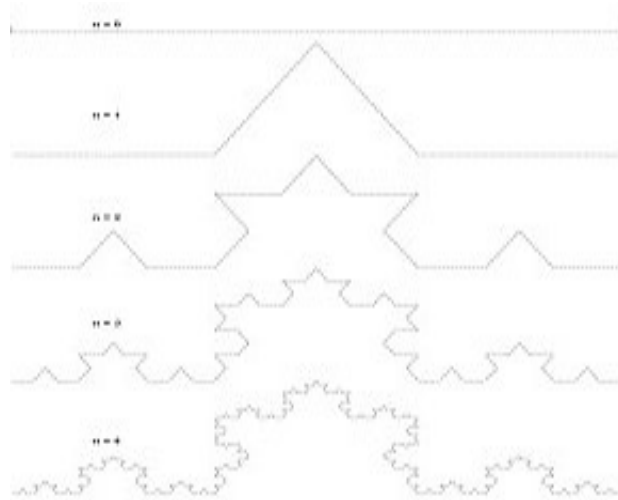
Pastebėsime, jog šią dimensiją apskaičiuoti netgi „paprastoms“ aibėms (fraktalams) yra pakankamai sudėtinga. Kaip pavyzdį, panagrinėkime Kocho kreivę, [14]. Po k iteracijų kreivę sudaro 4^k segmentai, kurių kiekvieno ilgis 3^{-k} . Todėl kreivę galima padengti (tiksliai) 4^k aibėmis, kurių diametrai lygūs 3^{-k} . Tokiu būdu,

$$H_{\varepsilon}^p(A) = H_{3^{-k}}^p(A) = 4^k (3^{-k})^p.$$

Pertvarkykime šį rezultatą taip:

$H_{\varepsilon}^p(A) = H_{3^{-k}}^p(A) = 3^{k(D-p)}$; čia $D = \ln 4 / \ln 3$. Pereidami prie ribos, kai dengiančiųjų aibių diametrai artėja prie nulio (t.y. $k \rightarrow \infty$), gauname

$$H^p = \lim_{k \rightarrow \infty} H_{3^{-k}}^p = \begin{cases} \infty, & p < D \\ 1, & p = D \\ 0, & p > D \end{cases}.$$



1.4 pav. Van Kocho kreivė (fraktalas)

Taigi, galima teigti, jog Van Kocho kreivės Hausdorfo ir Bezicovičiaus dimensija lygi $D = \ln 4 / \ln 3 \cong 1.262$.

Vėliau, B. Mandelbrotas šią dimensiją pavadino fraktaline dimensija. Faktiškai, tai nauja erdvės (poaibių) matavimo priemonė.

Daugelyje šaltinių, fraktalinės dimensijos sąvokos įvedimui naudojamas šiek tiek supaprastintas požiūris (interpretacija).

Imkime metrinę erdvę (\mathbb{R}^n, d) . Tegul $A \in \mathcal{H}(\mathbb{R}^n)$. Tarkime, kad $\varepsilon > 0$ ir $N(A, \varepsilon)$ žymi mažiausią uždarytųjų rutulių $B(x, \varepsilon) = \{y \in \mathbb{R}^n \mid d(x, y) \leq \varepsilon\}$, sudarančių baigtinį poaibio A denginį, skaičių, t.y. $N(A, \varepsilon)$ - mažiausias sveikasis skaičius toks, kad $A \subset \bigcup_{i=1}^{N(A, \varepsilon)} B(x_i, \varepsilon)$; čia $\{x_1, x_2, \dots, x_{N(A, \varepsilon)}\} \subset \mathbb{R}^n$.

Fraktalinės dimensijos sąvoka įvedama, remiantis intuityvia idėja, jog aibė A ($A \subset \mathbb{R}^n$) turi fraktalinę dimensiją D , jeigu

$$N(A, \varepsilon) \approx C\varepsilon^{-D}; \quad (1.1)$$

čia: C — tam tikra konstanta; „ \approx “ reiškia, jog $\lim_{\varepsilon \rightarrow 0} (\ln f(\varepsilon) / \ln g(\varepsilon)) = 1$, kai $f(\varepsilon) \approx g(\varepsilon)$.

Dabar, išsprendę (1.1) „lygybę“ (D atžvilgiu), gauname

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

Perėję prie ribos ($\varepsilon \rightarrow 0$), turime:

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

Ši riba vadinama poaibio A ($A \subset \mathbf{R}^n$) fraktaline dimensija ir žymima $D(A) = D$.

Žemiau pateikiame keletą svarbesnių teiginių, skirtų teoriniam ir eksperimentiniam fraktalinės Euklido erdvės poaibių dimensijos nustatymui.

1 teorema. Tegul $A \subset \mathbf{H}(\mathbf{R}^m)$; be to, $\varepsilon_n = Cr^n$ ($C > 0, 0 < r < 1, n = 1, 2, \dots$). Poaibis A turi fraktalinę dimensiją D , apibrėžiamą formule:

$$D(A) = D = \lim_{n \rightarrow \infty} \left\{ \frac{\ln N_n(A)}{\ln \left(\frac{1}{\varepsilon_n} \right)} \right\}.$$

Ši teorema leidžia tolydųjį kintamąjį ε pakeisti diskrečiuoju kintamuoju ε_n ; supaprastėja skaičiavimo procedūra. Teoremos (kaip ir visų kitų teiginių) įrodymą galima rasti knygoje [14].

2 teorema. Tarkime, kad $A \subset \mathbf{H}(\mathbf{R}^m)$; be to, erdvė \mathbf{R}^m padengta nesusikertančiais m —mačiais „kubiukais“ (dėžutėmis), kurių briaunos ilgis lygus $1/2^n$; tegul $N_n(A)$ žymi „kubiukų“, persidengiančių su poaibiu A , skaičių. Tada skaičius

$$D = \lim_{n \rightarrow \infty} \left\{ \frac{\ln N_n(A)}{\ln(2^n)} \right\}$$

vadinamas poaibio A fraktaline dimensija.

Būtent pastarasis teiginys sudaro (dažniausiai) pagrindą eksperimentiniam realaus pasaulio (fraktalinio) objekto dimensijos nustatymui (dimensijos įverčio gavimui).

Kaip pavyzdį, panagrinėkime fraktalą A (1.5 pav. Sierpinskio trikampis). Nesunku pastebėti, jog:

$$N_1(A) = 3, \text{ kai } n = 1;$$

$$N_2(A) = 9, \text{ kai } n = 2, \text{ t.y. kai } \varepsilon_2 = \frac{1}{2^2} = \frac{1}{4};$$

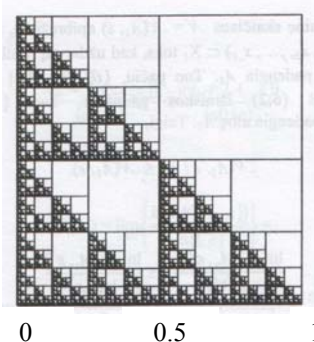
$$N_3(A) = 3^3 = 27, \text{ kai } n = 3 \text{ ir t.t.}$$

Pagaliau

$$N_n(A) = 3^n, \text{ kai } \varepsilon_n = \frac{1}{2^n}.$$

Taigi, fraktalinė Sierpinskio trikampio dimensija lygi:

$$D = \lim_{n \rightarrow \infty} \frac{\ln 3^n}{\ln 2^n} = \frac{\ln 3}{\ln 2} \cong 1.585.$$



1.5 pav. Sierpinskio trikampis

3 teorema. Jeigu $A, B \in \mathcal{H}(\mathbb{R}^n)$ ir $A \subset B$, tai $D(A) \leq D(B)$; be to, $D(A) \geq 0$, $D(B) \leq n$.

4 teorema. Jeigu $A, B \in \mathcal{H}(\mathbb{R}^n)$ ir $D(A), D(B)$ bei $D(A \cup B)$ žymi atitinkamai aibių A, B ir $A \cup B$ fraktalines dimensijas, tai $D(A \cup B) = D(A)$, kai $D(B) < D(A)$.

5 teorema. Tarkime, kad $A \subset \mathbb{R}^n$ sutampa su visiškai nejungios $\text{IFS}\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$, sudarytos iš afiniųjų panašumo transformacijų, atraktoriumi. Tada, poaibio A fraktalinė dimensija D yra lygties

$$\sum_{i=1}^N s_i^D = 1$$

sprendinys; čia s_i yra i -tosios afinėsios transformacijos suspaudimo koeficientas, $i=1, 2, \dots, N$.

Pavydžiui, Kocho kreivės atveju (1.4 pav.), pastarojo teiginio taikymas leidžia gauti tokį rezultatą. Kadangi Kocho kreivė yra $\text{IFS}\{\mathbb{R}^2; \omega_1, \omega_2, \omega_3, \omega_4\}$ atraktorius, be to, $s_i = 1/3$, su visais

$$i = 1, 2, 3, 4, \text{ tai } \sum_{i=1}^4 \left(\frac{1}{3}\right)^D = 1; \text{ iš čia } 4 \cdot \left(\frac{1}{3}\right)^D = 1; \left(\frac{1}{3}\right)^D = \frac{1}{4}; D = D = \frac{\ln 4}{\ln 3} \cong 1.262.$$

6 teorema. Tarkime, kad $D(A)$ ir $D_H(A)$ žymi kokio nors apręžto poaibio $A \subset \mathbb{R}^n$ atitinkamai fraktalinę bei Hausdorfo ir Bezicovičiaus dimensijas. Tada

$$0 \leq D_H(A) \leq D(A) \leq n.$$

Galima būtų teigti, jog $D_H(A)$ „šiek tiek subtiliau“ charakterizuoja poaibį A , negu $D(A)$.

Yra ir daugiau požiūrių bei interpretacijų, pateikiant fraktalines Euklido erdvės poabių dimensijas. Tai — koreliacijos dimensija, informacijos dimensija, Liapunovo dimensija ir panašiai. Plačiau apie jas čia nekalbėsime.

1.3. Fraktalinės interpoliacinės funkcijos, jų specifika

Elementariosios funkcijos (tokios kaip sinusas, kosinusas, daugianariai ir panašiai.) sudaro tradicinio, visuotinai priimto eksperimentinių duomenų analizės metodo pagrindą. Tarkime, kad eksperimento metu matuojamos tam tikros realiosios funkcijos $F(x)$ reikšmės. Eksperimento rezultatas - duomenų rinkinys $\{(x_i, F_i) \mid i=0,1,\dots,N\}$; čia $F_i = F(x_i)$, $i=0,1,\dots,N$, ir $x_0 < x_1 < \dots < x_N$. Šie duomenys (kaip erdvės \mathbb{R}^2 poabis) pavaizduojami grafiškai ir analizuojami, t. y. stengiamasi parinkti galimai žemesnio lygio daugianarį (atskiru atveju, „laužtę“) tokį, kuris „neblogai modeliuotų“ duomenis segmente $[x_0, x_N]$, t.y. kurio grafikas „eitų“ per taškus (x_i, F_i) , $i=0,1,\dots,N$.

Tačiau, grafikos sistemoms kartais keliami šiek tiek didesni reikalavimai, būtent: galimybė modeliuoti realaus pasaulio (fizinius) objektus, tokius kaip debesys, kalnų masyvo profilis, kabantys stalaktitai ir panašiai. Šiems fiziniams objektams (sistemoms) būdinga tai, kad kiekvienos sistemos dalies struktūra dažnai „atkartoja“ visos sistemos struktūrą. Tai galima aiškinti tuo, jog sistemą įvairiuose jos lygiuose veikiančios ir formuojančios jėgos yra panašios. Čia Euklido geometrijos ir elementariųjų funkcijų čia jau nebepakanka.

Pateiksime fraktalinės interpoliacinės funkcijos sąvoką, parodysime, jog minėto tipo duomenims galima parinkti funkcijas, kurios (Hausdorfo metrikos prasme) būtų „artimos“ tiems duomenims; be to, įmanoma užtikrinti, kad fraktalinės interpoliacinės funkcijos grafiko dimensija sutaptų su eksperimento duomenų (grafiko) fraktaline dimensija.

Tarkime, kad taškų rinkinys $\{(x_i, F_i) \in \mathbb{R}^2 \mid i=0,1,\dots,N\}$ sudaro duomenų aibę; čia $x_0 < x_1 < \dots < x_N$. Tolydžioji funkcija $f: [x_0, x_N] \rightarrow \mathbb{R}$ tokia, kad $f(x_i) = F_i$, $i=0,1,\dots,N$, vadinama interpoliacine funkcija, atitinkančia duomenų aibę. Taškai (x_i, F_i) , $i=0,1,\dots,N$, vadinami interpoliavimo taškais.

Pasirodo, jog erdvėje \mathbb{R}^2 galima sukonstruoti IFS, kurios atraktorius (aibė A) sutaptų su tolydžiosios duomenis interpoliuojančios funkcijos $f: [x_0, x_N] \rightarrow \mathbb{R}$ grafiku. Tam pakanka imti pražulniąsias (nėra posūkio y —ašies atžvilgiu) afiniāsias transformacijas, t.y.

$$\omega = \omega(x, y) = \begin{pmatrix} r_1 \cos \Theta^\circ & -r_2 \sin 0^\circ \\ r_1 \sin \Theta^\circ & r_2 \cos 0^\circ \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix} = \begin{pmatrix} a & 0 \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}, \text{ kai } (x, y) \in \mathbb{R}^2.$$

Kita vertus, afinioji transformacija

$$\omega_i = \omega_i(x, y) = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}, \quad i \in \{1, 2, \dots, N\},$$

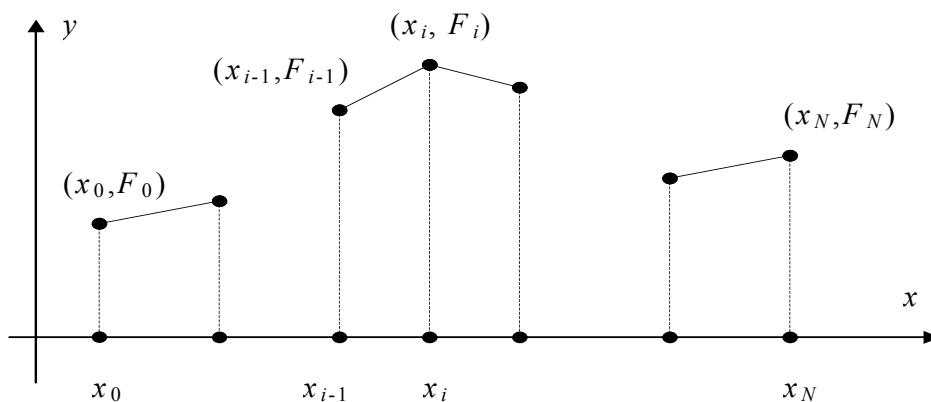
turi atvaizduoti duomenų aibę (nuo x_0 iki x_N) į „juostą“ (nuo x_{i-1} iki x_i), t.y. turi būti tenkinamos sąlygos:

$$\begin{cases} a_i x_0 + e_i = x_{i-1} \\ a_i x_N + e_i = x_i \\ c_i x_0 + d_i F_0 = f_i = F_{i-1} \\ c_i x_N + d_i F_N + f_i = F_i \end{cases}$$

Nesunku pastebėti, jog šioje tiesinių algebrainių lygčių sistemoje vienas kintamasis (tarsime, kad tai vertikalųjį mastelį keičiantis parametras d_i) yra laisvai pasirenkamas. Tada

$$\begin{aligned} a_i &= \frac{x_i - x_{i-1}}{x_N - x_0}, \quad c_i = \frac{F_i - F_{i-1}}{x_N - x_0} - d_i \frac{F_N - F_0}{x_N - x_0} \\ e_i &= \frac{x_N x_{i-1} - x_0 x_i}{x_N - x_0}, \quad f_i = \frac{x_N F_{i-1} - x_0 F_i}{x_N - x_0} - d_i \frac{x_N F_0 - x_0 F_N}{x_N - x_0}. \end{aligned} \quad (1.2)$$

Jeigu parinktume $d_i=0$, su visais $i=1, 2, \dots, N$, tai tokios IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$, atraktorius A sutaptų su laužte (dalimis tiesine interpoliacine funkcija, kurios grafikas „eina“ per interpoliavimo taškus (1.6 pav.))



1.6 pav. Dalimis tiesinės interpoliacinės funkcijos, einančios per interpoliavimo taškus, grafikas; šis grafikas sutampa su IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktoriumi

Jeigu parametru d_i ($i=1, 2, \dots, N$) reikšmės nėra nulinės, tai IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktorius išlieka duomenis $\{(x_i, F_i) \in \mathbb{R}^2 \mid i=0, 1, \dots, N\}$ interpoliuojančiąja funkcija, tačiau tai jau nėra dalimis tiesinė funkcija. To priežastys akivaizdžios - IFS sudaro pražulniosios afiniosios transformacijos, t.y. bet kokia tiesė (atkarpa), lygiagrečiai ašiai O_y , atvaizduojama $(\omega_i, i \in \{1, 2, \dots, N\},$ pagalba) į tiesę (atkarpa), lygiagrečią ašiai O_x . Be to, abiejų atkarpų (prieš ir po vaizdavimo) ilgių santykis

išreiškiamas skaičiumi $|d_i|$. Nesunku pastebėti, jog parametrai d_i ($i=1, 2, \dots, N$) įtakoja IFS atraktoriaus (fraktalinės interpoliacinės funkcijos) fraktalinę dimensiją.

Pateikiame vieną (šia prasme labai svarbų) teiginį.

1 teorema. Tarkime, kad $\text{IFS}\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ yra susieta su duomenų aibe $\{(x_i, F_i) \in \mathbb{R}^2 \mid i = 0, 1, \dots, N\}$; be to,

$$\omega_i(x, y) = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix};$$

čia koeficientai a_i , c_i , e_i ir f_i apibrėžiami (1.2) išraiškomis, su visais $i = 1, 2, \dots, N$; $0 \leq d_i < 1$, $i = 1, 2, \dots, N$. Tarkim, kad A yra fraktalinės interpoliacinės funkcijos grafikas (IFS atraktorius).

Jeigu $\sum_{i=1}^N |d_i| > 1$ ir interpoliavimo taškai nepriklauso vienai tiesei, tai A turi fraktalinę dimensiją

D , kuri yra vienintelis realusis lygties $\sum_{i=1}^N |d_i| \cdot a_i^{D-1} = 1$ sprendinys; priešingu atveju, $D(A) = 1$.

Šios teoremos ir kitų įdomių teiginių įrodymus galima rasti knygoje [14].

Pastebėsime, jog tuo atveju, kai interpoliavimo taškai yra pasiskirstę segmente $[x_0, x_N]$ tolygiai, t.y. $x_i = x_0 + x_i = x_0 + \frac{i}{N}(x_N - x_0)$, $i = 1, 2, \dots, N$, koeficientų a_i ($i = 1, 2, \dots, N$) reikšmės lygios $1/N$. Gauname

$$\sum_{i=1}^N |d_i| = N^{D-1}.$$

Pastarąją lygtį spęsdami D atžvilgiu, randame, jog:

$$D = 1 + \frac{\ln\left(\sum_{i=1}^N |d_i|\right)}{\ln N}.$$

Vadinasi, manipuluojant parametru d_i ($i = 1, 2, \dots, N$) reikšmėmis, dydžiui $\sum_{i=1}^N |d_i|$ galima priskirti bet kurią reikšmę iš intervalo $[1, N]$, tuo pačiu fraktalinės interpoliacinės funkcijos (IFS atraktoriaus A) fraktalinę dimensiją prilyginti bet kuriai reikšmei iš intervalo $[1, 2]$. Tai labai svarbus momentas, kadangi atsiranda galimybė „priderinti“ fraktalines interpoliacinių funkcijų dimensijas prie duomenų aibės (eksperimento rezultatų) fraktalinės dimensijos.

Įdomu ir tai, kad fraktalinė dimensija nepriklauso nuo reikšmių $\{F_i \mid i = 1, 2, \dots, N\}$, išskyrus tą faktą, jog interpoliavimo taškai neturi priklausyti vienai tiesei. Taigi, galima kalbėti apie fraktalinių interpoliacinių funkcijų, turinčių tą pačią dimensiją D , rinkinį. Tam pakanka išpildyti sąlygą:

$$\sum_{i=1}^N |d_i| = N^{D-1}.$$

Nors fraktalinės interpoliacinės funkcijos (FIF) buvo formalizuotos daugiau nei prieš dešimtmetį, tačiau tik pastaraisiais metais imta plačiau jas taikyti kompiuterinėje grafikoje. FIF taikomos kalbos signalų interpoliavimui, modeliuojant kalno vaizdą, seisminius duomenis, elektrokardiogramas. Modeliuojant įvairius paviršius taikomos dvi sudėtinės FIF dalys: FIS (fraktalinės interpoliacinės sistemos) ir FIV (fraktaliniai interpoliaciniai paviršiai).

Be įprastinės FIF dar taikomos ir RFIF (Recurrent Fractal Interpolation Functions) ir PFIF (Partitioned Fractal Interpolation Functions). Rekurentinės FIF — FIF apibendrinimas, sudarantis lankstesnes sąlygas „grubių“ kreivių formavimui. RFIF leidžia naudoti atitinkamos srities dalių sąjungą, o PFIF supaprastina šį apribojimą ir leidžia naudoti laisvai pasirenkamą srities dalį.

Tad reikalingi metodai, skirti fraktalinių funkcijų, interpoliuojančių duotą duomenų rinkinį, formavimui. Tuo pačiu, FIF sudaro galimybes naudoti naujus eksperimentinių duomenų aproksimavimo metodus. W. O. Kochranas (W. O. Cochran), J. C. Hart Patrikas (J. C. Hart Patrik) bei J. Flainas (J. Flynn) savo straipsnyje apie „grubių“ kreivių aproksimavimą, naudojant fraktalines funkcijas, siūlo aproksimavimo problemoms spręsti taikyti Hugh'o (Hough) transformacijų, vaizdų suspaudimo bei mažiausių kvadratų metodus, [2].

Darbe bandoma analizuoti ir spręsti interpoliavimo taškų parinkimo (bendru atveju) problemą. Taipogi, siūlomas naujas požiūris į pražulniųjų afinių transformacijų, sudarančių IFS, parinkimą.

1.4. Kai kurie fraktalų taikymo pavyzdžiai

Nors fraktalų teorija taikoma tik keletą dešimtmečių, tačiau ja paremti tyrimai (eksperimentai) leido mokslininkams paaiškinti galaktikų evoliuciją, gardelės vystymąsi, kalnodaros ir debesų susidarymo procesus, biržos kainų kaitos procesus ir netgi visuomenės bei šeimos vystymąsi, t.y. vis daugėja sričių, kuriose yra taikomi fraktalinės geometrijos teiginiai ir metodai.

Fraktalai pirmiausiai buvo pritaikyti kompiuterinėje grafikoje (vaizdų suspaudimas, vaizdų modeliavimas ir pan.), kompiuterinėse technologijose (duomenų suspaudimas), fizikoje, medicinoje bei ekonomikoje.

1991 m. M. Barnsley pateikė fraktalinio vaizdų suspaudimo algoritmą, kurio esmė ta, kad IFS parametrų pagalba vaizdas saugomas daug kompaktiškiau. Be to, galima išskirti keletą fraktalinio vaizdų suspaudimo privalumų: suspaustas vaizdas užima nedaug vietos ir jo atkūrimas užtrunka trumpiau, nei taikant kitus metodus; aukšta atkurto vaizdo kokybė; video suspaudimo galimybė. Savo ruožtu, fraktalinio vaizdų suspaudimo algoritmas leidžia (kaip ir JPEG atveju) užduoti suspaudimo laipsnį (suspaudimas su pradimu). Algoritmo esmė: paieška didelių vaizdo fragmentų, kurie būtų panašūs į mažesnius fragmentus. Faile saugomos tik rastų atitikmenų poros.

Fraktalų panašumas į kalnus, gėles, medžius taikomas kai kuriuose grafiniuose redaktoriuose (fraktaliniai debesys— 3D Studio MAX, fraktaliniai kalnai— World Builder ir kt.).

Be to, eksperimentais yra įrodyta, kad norint kompiuteryje pavaizduoti trimatę gamtą, būtina kurti trimates fraktalines figūras, nes fraktalai pateikia didžiausią realių detalių kiekį, esant mažiausiam kompiuterio atmintyje saugomų duomenų kiekiui. Vietoj atmintyje saugomų visų kalno detalių galima aprašyti objektą fraktaline lygtimi, arba saugoti tik atraktorių, nusakantį visą objektą (analogiškai, trimačiams upės, jūros pakrančių vingių vaizdams gauti ir kt.).

Fraktalų panašumo savybė itin naudinga, formuojant naujas tekstūras. Juk kiek bedidintumėm, fraktalinis vaizdas nėra „stambinamas“, praktiškai, nuolat atstatomas pradinis vaizdas. Be to, fraktalo „neribotumas“ tiek į plotį, tiek į gylį, leidžia, labai sumažinus ar padidinus fraktalą, gauti, praktiškai, naujus paveikslėlius.

Fraktalinės geometrijos taikymas (IFS, ar FIF) — tai naujas požiūris į procesų analizę bei prognozes. JAV žemdirbystės departamente (US Department of Agriculture) tyrinėjami fraktalų taikymai fotografijoms (darytoms iš oro), siekiant išanalizuoti augalininkystės rūšių plėtros galimybes. M. Barnsley vienas pirmųjų pasiūlė fraktalinę geometriją taikyti iš palydovų gautų vaizdų analizei, upių pakrančių ir jų baseinų struktūros analizei. Pastaruoju metu fraktalai vis plačiau taikomi modeliuojant socialinių arba civilizacinių procesų modelius.

XX a. 80—ųjų pradžioje analitikas Džonas Merfis sukūrė techninės kainų analizės teoriją, kurios tikslas – kainų raidos prognozavimas. Nors buvo manoma, kad kainų kaita —chaotiškas darinys, matematikas B. Mandelbrotas sukūrė universalų techninės kainų analizės metodą, kurio pagalba galima prognozuoti įvairių prekių ir prekybos kainas, kadangi ir kaitos taisyklės, ir kainų grafiko struktūra— nekinta.

Techninė analizė taip pat pagrįsta fraktalų geometrijos teiginiais. Apskritai, egzistuoja trys techninės analizės metodų grupės: grafiniai metodai, filtravimo metodai ir ciklų teorija . Tipinis techninės analizės fraktalo pavyzdys yra Elioto bangos.

Fraktalai, pritaikyti labai sudėtingoje antenų konstravimo srityje, pagerino antenų matricų kokybę. Be to, antenoms suteikus fraktalo formą, pagerėjo antenų parametrai. Pavyzdžiui, antenoms suteikus Kocho kreivės ar Serpinskio trikampio formą, antenų plotas sumažėjo 6 kartus; buvo sukurta elektrinė talpa bei indukcija. To pasekoje, derinant anteną ar didinant jos priimamų dažnių juostos plotį, nebereikia išorinių komponentų. Fraktalų antenos sėkmingai naudojamos Motorola mobiliuosiuose telefonuose.

Nuo 1980—ųjų svarbiausia fraktalinės geometrijos taikymo sritimi tampa biologija ir medicina. Šį procesą įtakojo tas faktas, kad fraktalinėmis savybėmis pasižymi praktiškai visos gyvo organizmo sistemos: kraujo indai ir kapiliarai, bronchai ir plaučių bronchiolės (suformuoja „medį“ su daugybe išsišakojimų), kvėpavimo takai (jų struktūra nekinta, keičiant mastelį), limfmazgiai, nervų sistema, virškinamasis traktas (plonosios žarnos stambios ir smulkios detalės panašios) ir kita. Smegenų, širdies funkcijų dinamika taip pat fraktalinė. Pastaraisiais metais daugiausia analizuoti buvo kvėpavimo takai. Čia savo indėlių įnešė įvairių šalių mokslininkai: E. Uebelas, D. Gomesas, O. Raabe, Uestas, Goldbergeris, V. Bchargava, T. Nelsonas ir kiti. Širdies arterijų ir venų fraktalinę struktūrą tyrinėja Dž. Bassingtueitas, Ch. Fon Beekas (pagrindinis tyrimų tikslas —kraujotakos anomalijų paaiškinimas). Neuronus, kaip fraktalines struktūras (dendritai išsišakoja į vis plonesnes gijas) pastaraisiais metais yra tyrinėję G. Majeris—Kresas, P. Repas, A. Bablojanis bei kiti. Svarbiausieji šios srities pasiekimai nuolat aptariami specializuotose tarptautinėse konferencijose (pirmoji įvyko 1993 m. Budapešte), kuriose pateikiama fraktalinė kardiologijos ir encefalografijos fraktalinių procesų apdoravimo metodų analizė. Taip pat yra leidžiamas specializuotas žurnalas „Fraktalai“.

Apskritai, pastaraisiais dešimtmečiais fraktalai vis dažniau taikomi chemijos, biologijos moksluose, kalbotyros moksluose bei muzikoje. Įvairių fraktalinių dydžių analizė ir fraktalinių objektų modeliavimas, seniai nebe fizikų ir programuotojų prerogatyva, randa pačias netikėčiausias taikymo sritis. Tiriama tekstų ir hipertekstų, muzikinių kūrinių, biržos kotiruočių, smegenų procesų dinamikos, komunikavimo socialinėse sistemose darinių fraktalinės struktūros, globalių informacinių tinklų dinaminiai procesai“ [6].

Fraktalinės geometrijos kalba būtina, taipogi, tiriant spinduliavimo elementariose terpėse sklidimą ir sugėrimą, ryškios turbulencijos charakterizavimui, kietųjų kūnų paviršių savybių modeliavimui, žaibo ir elektros smūgio aprašymui, objektų senėjimo procesų analizėje, difuzijos ir agregacijos įtakos kūnams tyrimui, kvantinėje mechanikoje— banginių funkcijų geometrinių struktūrų Andersono metalo— dielektriko perėjimo taške aprašymui. Astrofizikoje— aprašant Visatos galaktikų klasterizacijos procesus; kartografijoje— tiriant pakrančių linijų formas, upių tinklą; biologijoje— kraujotakos sistemos struktūros analizei arba sudėtingų paviršių ląstelių membranų apžvalgai. Be to, tiriamos fraktalinės tekstų ir hipertekstų struktūros, muzikos kūriniai, biržos kotiruotės, smegenų procesų požymių dinamikoje, auklėjimo atraktoriai, socialinių sistemų komunikacijos struktūros, globalių informacinių tinklų dinaminiai procesai.

Apskritai, fraktalams atsirado vieta įvairiuose moksluose. Pavyzdžiui, susiformavo fraktalinių įtrūkimų teorija, fraktalinių paviršių trinties teorija, medžio—polimerų pavidalo kompozitų fraktalinė mechanika, fraktalinė optika bei kitos sritys. Fraktalų pagalba mokslininkai aiškina galaktikų evoliuciją, kalnodaros procesus, debesų susidarymą, biržos kainų pokyčius, visuomenės ir šeimos vystymąsi.

2. TIRIAMOJI DALIS

FRAKTALINIŲ INTERPOLIACINIŲ FUNKCIJŲ ANALIZĖ: TEORINIAI IR PRAKTINIAI ASPEKTAI

Plečiantis fraktalinių interpoliacinių funkcijų (FIF) praktinio panaudojimo sferai reikalinga priemonė, kurios pagalba būtų galima analizuoti (interpoliuoti) ne vien duomenis apie geografinius landšafto objektus, bet ir ekonominius, statistinius, medicininius ir kt. pobūdžio duomenis. Atskirų tipų duomenų analizei priemonių yra sukurta nemažai (Fractan, Market Way Pro, Investor's Dream, SKO bei kt.), bet vienos universalios priemonės, kurią būtų galima taikyti įvairiarūšei informacijai analizuoti, naudoti neteko. Tam tikru universalumu pasižymi fraktalinio interpoliavimo įrankis INTERPO, [8].

Šiame darbe analizuojami priemonės, skirtos FIF generavimui, taikymo metodai bei apžvelgiama konkretaus objekto FIF kreivių generavimo eksperimento rezultatai. Eksperimento metu pastebėtos darbo su esamu INTERPO variantu problemos privertė pabandyti priemonę adaptuoti.

2.1. INTERPO—fraktalinio interpoliavimo įrankis

Fraktalinės interpoliacinės funkcijos (FIF)—tai naujas eksperimentinių duomenų perteikimo įrankis. Fraktalinės interpoliacinės funkcijos gali būti naudojamos tokių eksperimentinių duomenų, kaip laiko funkcijos (elektroencefalogramos ir pan.), modeliavimui. FIF grafo fraktalinė dimensija „atitinka“ duomenis tam tikrame intervale [1].

FIF generavimas rankiniu būdu yra sudėtingas, o remiantis interpoliacijos principais FIF paiešką galima kompiuterizuoti. Viena iš tokių priemonių ir yra D. Oliverio sukurta priemonė Fractal Vision [8].

Produkto pavadinimas: **Fractal Vision**

Autorius: **Dick Oliver:**

Priemonės sudėtis:

Fractal Vision – fraktalų generavimui skirta priemonė (**fv.exe**);

Interpo – fraktaliniam interpoliavimui skirta priemonė (**interpo.exe**);

Life – žaidimas (**life.exe**).

Produkto anotacija:

Produktas sukurtas 1990 m. balandžio mėn.; pradinė jo paskirtis – fraktalų generavimas.

INTERPO – *Fractal Vision* sudėtinė dalis, skirta fraktalų interpoliacijos tyrimams. Priemonei pritaikyti reikalingas pradinis *fip* tipo failas, kuriame pateikiami duomenys apie interpoliavimo taškus.

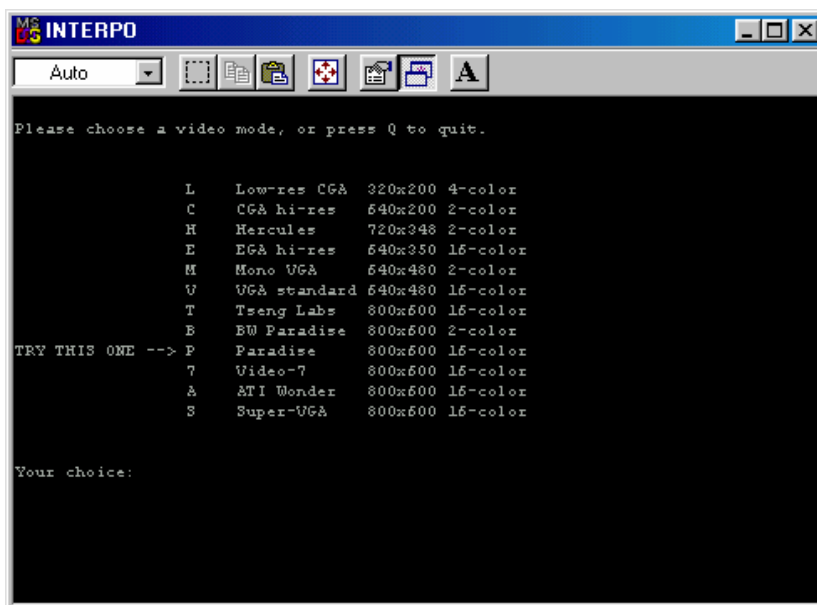
Vykdamt programą, pirmiausiai nurodomas vaizduoklio tipas, po to – nurodomas duomenų failas bei dimensija. Priemonė ekrane pateikia:

- Interpoliuojamą duomenų aibę bei kitus parametrus;
- interpoliavimo taškais apribotoje srityje gautą fraktalinę kreivę.

„Interpo skaito tekstinį (ASCII) failą, kuriame saugomos x , y reikšmės, spausdina gautus rezultatus ir pateikia ekrane grafinį vaizdą. Taip gaunamos fraktalinės interpoliacinės funkcijos, einančios per duomenų faile nurodytus taškus, vaizdas. Kreivės išlinkimai priklauso nuo skaitinės vertikalųjų mastelių keičiančio parametro reikšmės“, [8].

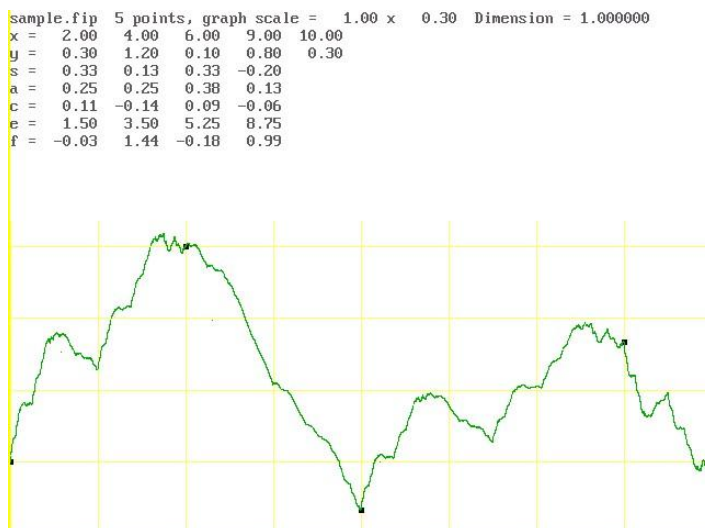
Kadangi pradinė Interpo.c versija sukurta rinkiniui, sudarytam iš mažiau nei 100 elementų, tai norėdami ją taikyti didesniai duomenų kiekiui turime keisti pradinę programą. Tad duomenų rinkinį turėtų nusakyti taškai, geriausiai atspindintys kreivės svyravimus.

INTERPO pertvarko bet kokį duomenų, esančių tarp dviejų nuoseklių intervalo galų, rinkinį. Pavyzdžiui, jei užduosime 100 taškų, tai kreivę gausime po 99 pertvarkių. FIF atitikimas realiai situacijai bus tuo tikslesnis, kuo tinkamiau bus parinkti interpoliavimo taškai. Kaip juos parinkti? „Idealiausia, kad tai būtų kraštiniai dalių, panašių į visumą taškai. Jei vizualiai nustatyti panašumą sunku, galima parinkti kas 10—tą, kas 50—tą ar 500—tąjį tašką, priklausomai nuo to, kiek duomenų turime apskritai.“ [8]



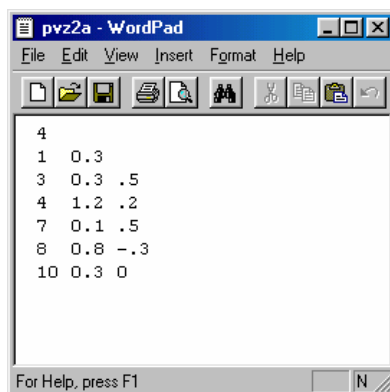
2.1 pav. Pradinis INTERPO ekranas – jame siūloma pasirinkti monitoriaus tipą

Toliau nurodome reikiamos bylos pavadinimą ir pasirenkame interpoliuojamo objekto fraktalinės dimensijos D reikšmę (nuo 1 iki 2).



2.2 pav. Duomenų aibės (Sample.fip), kai $D=1$, interpoliavimo rezultatas

Norėdami sudaryti savo fraktalą, turime suformuoti naują **fip** tipo bylą. Tą padaryti galime su paprasčiausia teksto redagavimo priemone (Notepad, ar WordPad):

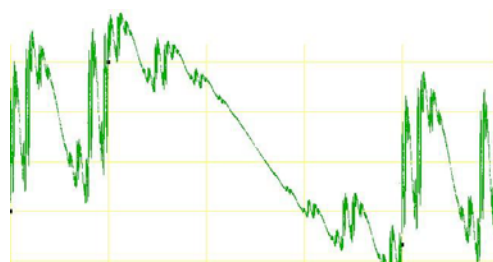


2.3 pav. Duomenų aibės pateikimas

Šiame dokumente nurodome interpoliavimo taškų kiekį, skalę (tinklėlių), kintamųjų x ir y bei vertikalųjų mastelių keičiančio parametro s reikšmes. Pagal pateiktus duomenis, papildomai nurodę dimensiją, gausime atitinkamą fraktalinę kreivę (FIF grafiką):



2.4 pav. FIF grafikas; Pzv2a.fip, kai $D=1$



2.5 pav. FIF grafikas; Pzv2a.fip, kai $D=1.61$

Parametras s apsprendžia kreivės „grubumą“. Pavyzdžiui, jei pirmojo interpoliavimo intervalo parametras $s = 0,5$, o visų kitų— $0,2$, tai kraštinė kairioji kreivės dalis bus aukštesnė ir „grubesnė“ nei likusioji dalis. Neigiama parametro s reikšmė nusako atvirkščią kreivės linkį horizontalios ašies atžvilgiu. „Jei parametro s reikšmės nežinome, pradiniam eksperimentui renkamės $0,5$, o vėliau, įvertinę gautą rezultatą vizualiai, reikšmes tiksliname. Apskritai s reikšmės priklauso intervalui $(-1; 1)$ “ [8].

Tinklelį parenkame priklausomai nuo aprašomų duomenų kaitos intervalų. Pavyzdžiui, jei x ir y reikšmės yra nutolusios (kinta) per vieną vienetą, tinklelis 1×1 , jei x yra tarp 300 ir 500, o y tarp 0,01 ir 0,02, tai tinklelis gali būti $50 \times 0,001$. Šių duomenų reikia FIF tinkleliui nurodyti.

Taigi, INTERPO duomenų failas turėtų atrodyti taip:

- bendras interpoliavimo taškų skaičius
- tinklelis
- x, y , vertikalųjį mastelį keičiantis parametras s .

Paskutinėje duomenų eilutėje esantis parametras nebereikalingas, nes už paskutinio interpoliavimo taško kreivės nėra, bet kadangi programoje nurodyti trys skaitomi duomenys, tai ir paskutinėje duomenų failo eilutėje šis parametras įrašomas.

Vykdamas INTERPO, pasirinkta fraktalinės dimensijos reikšmė apsprendžia FIF kreivės vertikalų „grubumą“. Fraktalinei dimensijai nustatyti galime panaudoti papildomą priemonę –*Fracdim*. Ši priemonė leidžia stebėti FIF kreivės priklausomybę nuo dimensijos D kaitos. D reikšmes galime užduoti savarankiškai (pasirinkdami iš tam tikro intervalo, arba taikydami atsitiktine tvarka parinktas vertes). *Fracdim* ypač naudinga tais atvejais, kai nustatyti dimensiją programiškai neįmanoma (pavyzdžiui, analizuojamas skenuotas vaizdas), o matematiniai skaičiavimai —sudėtingi ir reikalaujantys daug laiko.

Pagrindinės problemos, su kuriomis gali susidurti vartotojas:

- Nenurodytas duomenų failo vardas;
- Neteisingas duomenų failo vardas;
- Klaidingas duomenų failo turinys;
- Nurodytu vardu pavadinto failo šiame diske nėra;
- Nenurodyta objekto dimensija;
- Klaidingai įvesta dimensija.

2.2. Interpoliavimo taškų parinkimo kriterijai ir jų nustatymas

Interpoliavimo taškų parinkimui šiame darbe siūlomi šie metodai:

- Interaktyviam naudojimui orientuotas tolygaus interpoliavimo taškų parinkimo metodas;
- Koliažas.

2.2.1. Interaktyviam naudojimui orientuotas tolygus interpoliavimo taškų išdėstymas

Objekto fraktalinės analizės procesą galima suskaidyti į keletą etapų:

- 1) Nustatoma objekto fraktalinė dimensija ;
- 2) Suformuojamas objektą nusakantis duomenų masyvas (duomenų aibė). Jei duomenų aibė sudaro daugiau nei 100 elementų—ją reikia sumažinti, taikant duomenų atrankos metodus (INTERPO leidžia apdoroti tik iki 100 elementų);
- 3) Nustatomos vertikalųjų mastelių keičiančių parametrų s reikšmės;
- 4) Stebima fraktalinės kreivės priklausomybė nuo fraktalinės dimensijos (priemonė Fracdim);
- 5) Formuojama fraktalinė interpoliacinė kreivė (su priemone INTERPO).

Šiuos etapus apžvelgsime detaliau.

1. Grafinio *.pcx tipo vaizdo fraktalinės dimensijos nustatymui galime pasinaudoti Fractal Vision arba rinktis pradinę atsitiktinę reikšmę, kurią vėliau bus galima koreguoti.

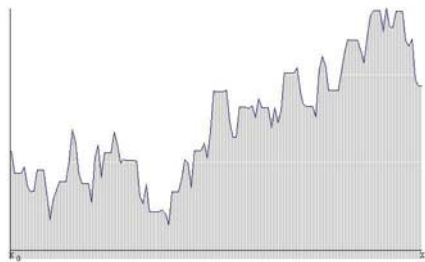
2. Objekto fraktalinis interpoliavimas priklauso nuo objektą nusakančių duomenų parinkimo metodų.

Grafinio objekto vaizdą geriausiai nusako koordinatinis duomenų masyvas (x_i, y_i) , $i=0, 1, \dots, N$. Šį masyvą galima suformuoti tiek naudojant *vizualinį metodą* (analizuojant geografinius atlasus ar žemėlapius, kai rezultatas priklauso nuo vietovės mastelio), tiek ir naudojant *programines priemones*. Pastarasis būdas leidžia objektą nusakyti tikslesniu duomenų rinkiniu, nei elementarus vizualinis būdas. Tačiau ir čia daug kas priklauso nuo grafinio objekto vaizdo pateikimo bei analizės priemonės. Galime naudoti skenuotą vaizdą, jį analizuodami kokio nors grafinio redaktoriaus pagalba. Tada rezultatas priklausys nuo pradinio vaizdo mastelio, skenerio dpi, ekrano raiškos bei priemonės — grafinio redaktoriaus —tinklelio tikslumo. Galutinį duomenų masyvą (duomenų aibę) galime suformuoti, pasirinkdami, tarkime, patį mažiausią (1x1) tinklelį ir įtraukdami jo vertikalės ir pradinį objektą nusakančios kreivės susikirtimo taškus.

Formuojant duomenų aibę, gaunamas duomenų masyvas gali būti sudarytas iš daygybės elementų, tad gali būti sudėtinga panaudoti šiuos duomenis fraktaliniam interpoliavimui su INTERPO, nes gautos aibės elementų skaičius gali būti didesnis nei 100 elementų.

Šią problemą išspręsimė, taikydami *duomenų atrankos* metodus.

Tarkime, turime fraktalinę kreivę, nusakytą duomenų rinkiniu (x_i, y_i) ($i = 0, 1, \dots, N$), atvaizduojančią fraktalinę funkciją $f: y_i = f(x_i)$, kai $i \in [0; N]$ ir $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$. Reikia parinkti rinkinį $f(x'_j)$, kad $0 < j < N$, $f(x'_j) = f(x_j)$ ir $f: \mathbb{R}^2 \rightarrow \mathbb{R}^2$.



2.6 pav. Diskretizuotos kreivės pavyzdys.

Kadangi duomenų masyve gali būti ne daugiau kaip 100 elementų, tai pats paprasčiausias (elementariausias) jų atrankos metodas būtų *atsitiktinės duomenų atrankos* metodas, kai iš turimo duomenų rinkinio atsitiktine tvarka būtų išrenkama 100 elementų (pavyzdžiui, 99 pirmieji elementai ir paskutinis ar pirmasis ir 99 paskutiniai elementai).

Tačiau tikimybė, kad optimaliausia, t.y. geriausiai atspindintį visumą, duomenų rinkinį pavyks gauti iš karto, labai maža, jei vizualiai negalime nustatyti interpoliavimo taškų. Tokiu atveju teks atranką pakartoti. Tad nuoseklus duomenų aibės formavimo eksperimentas gali tęstis gan ilgai (tokių derinių C_N^{100}).

Kitas, sąlyginai, atsitiktinės atrankos metodas—*atranka pagal pasirinktą požymį* (statistinių duomenų, biržos duomenų atranka pagal savaitės ar mėnesio dieną ir pan.)

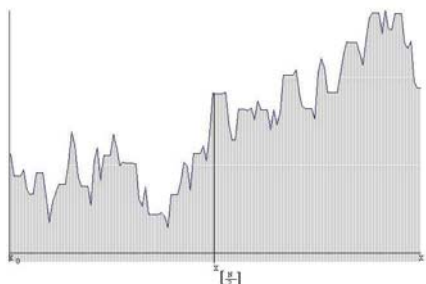
Tačiau efektyviausia bus duomenų atranka, pritaikant kokį nors *matematinį duomenų atrankos* metodą. Paprasčiausias iš jų —*aritmetinės progresijos principu paremtas* metodas, kai $x_j = x_{j-1} + h$, $j = 1, 2, \dots, N$. Tokiu būdu, priklausomai nuo pasirinkto žingsnio h , gausime duomenų rinkinį, kuriame bus $\left[\frac{N}{h} \right]$ elementų. Taigi, galutinis rezultatas priklauso nuo pasirinkto žingsnio dydžio.

Interpoliuojant su INTERPO, žingsniui h galioja vienintelė sąlyga—gautų elementų (taškų) skaičius negali būti didesnis nei 100. Dažniausiai tokį duomenų rinkinį $\{x_0, x_{0+h}, \dots, x_n \mid n \leq 100 < N\}$ sudaro kas 5 – tas, 10 – tas, 50 – tas, 100 – tasis ir pan. elementai. Statistiniai, biržos duomenys gali būti atrenkami ir imant kas 7 – tą (savaitė), 30 – tą (mėnuo) ar 365 – tą (metai) elementą.

Pagrindinė problema—kaip nustatyti žingsnį h ? Atsitiktinai parinktas h nevisada gali sudaryti sąlygas (net vizualiai vertinant) interpoliavimo metu gauti siektiną rezultatą (kaip ir, apskritai, atsitiktinės atrankos metodu gautas duomenų rinkinys).

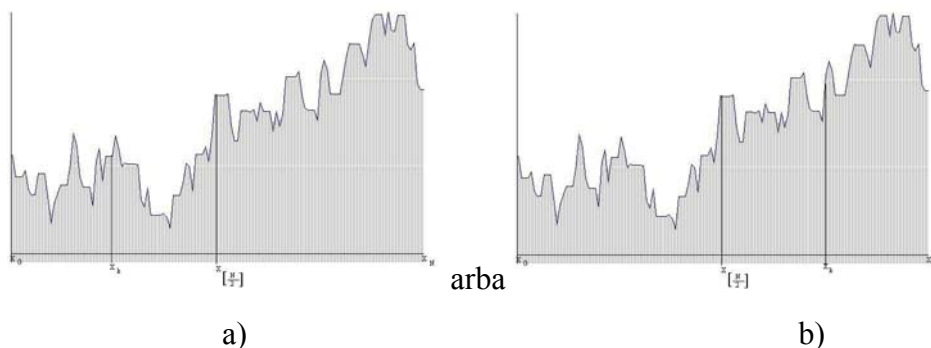
Dar vienas iš elementarių matematinių metodų yra *intervalo dalijimo pusiau metodas*. Intervalas $[0; N]$ dalijamas pusiau, ko pasekoje gaunami trys taškai, kuriems jau galima taikyti

interpoliavimą: $\left\{ x_0, x_{\left[\frac{N}{2}\right]}, x_N \right\}$.



2.7 pav. Pirminis intervalo dalijimas pusiau

Gautas interpoliavimo rezultatas greičiausiai (net vizualiai vertinant) nebus siektinas rezultatas. Todėl algoritmą reikėtų taikyti toliau, pasirinkus vieną iš gautų intervalų:



2.8 pav. Naujai dalijamos srities pasirinkimas

Dalijimo pusiau procedūra tęsiama tol, kol bus gautas pakankamai geras vaizdas. Jį įvertinti dažniausiai galime vizualiai. Statistinio rezultato tikslumą (patikimumą) galime įvertinti ir skaičiuodami statistinį patikimumo įvertį $\mu \left(\mu = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - y_{ii}}{y_i} \right| \cdot 100\% \right)$. Jei šis įvertis neviršija 10%, teigiama, kad gauti duomenys yra pakankamai patikimi.

Dar geresnius rezultatus galima būtų gauti, duomenų atrankai pritaikius sudėtingesnius matematinius metodus—*polinominės atrankos, splineų, bangų, greitą Furje atranką* bei kitus metodus.

3. Vertikalų mastelį keičiančių parametrų reikšmių nustatymas. Pradiniame etape, paprastai, pasirenkama ta pati reikšmė visiems parametrams, pavyzdžiui – $s_i = 0,5$ ($i = 1, 2, \dots, N$). Vėliau, dažniausiai vertinant vizualiai, s_i reikšmės koreguojamos, turint omeny tai, kad $s_i \in (-1; 1)$ ir neigiamos s_i reikšmės panaudojimo rezultatas – atitinkamos kreivės dalies atspindys x ašies atžvilgiu. Be to, didinant parametą, kreivės lankas („danteliai“) ilgės, mažinant – trumpės.

Šiuo etapu praktiškai baigiamas duomenų paruošimas ir jie perduodami INTERPO. INTERPO, kaip jau buvo minėta, pirmiausiai nurodomas vaizduoklio tipas bei duomenų failo pavadinimas (būtinai turi būti nurodytas ir tipas *.fip), įvedama dimensija ir gaunamas rezultatas. Toliau, duomenys gali būti koreguojami ir eksperimentas kartojamas iš naujo tol, kol bus gautas optimaliausias rezultatas —FIF kreivė —vizualiai panaši į analizuojamą objektą.

2.2.2. Koliažo organizavimas

Bazinę fraktalinių interpoliacinių funkcijų generavimo idėją (1.3 skyrelis) galima ir tikslinga apibendrinti panaudojant Koliažo teoremą lokalioms iteruotųjų funkcijų sistemoms. Intuityviai suprantama, kad toks apibendrinimas leistų parinkti (sugeneruoti fraktalines interpoliacines funkcijas atsižvelgiant į apdorojamų duomenų aibių pobūdį.

Žemiau pateikiame siūlomos koliažo organizavimo idėjos aprašą.

Tarkime, kad duomenų aibė

$$S = \{(x_j, F_j) \mid j = 0, 1, \dots, M\}$$

gauta diskretizuojant (skenuojant) realaus pasaulio objektą arba užrašant jau atlikto eksperimento rezultatus. Tarkime, taipogi, jog taškai x_j ($j = 0, 1, \dots, M$) yra išsidėstę tolygiai, t.y. vienodu atstumu vienas nuo kito; be to, skaičius M yra pakankamai didelis.

Dėl įvairių priežasčių (2.1 skyrelis), interpoliavimo procedūra dažnai organizuojama ne visai duomenų aibei S , o tik tam tikram jos poaibiui S' (pavyzdžiui, imant kas antrą, kas trečią (ar panašiai) aibės S elementą), t.y.

$$S' = \{(x_j, F_j) \mid j = \lambda i, i = 0, 1, \dots, N\}$$

čia $\lambda \in \{2, 3, \dots\}$.

[veskime pažymėjimus:

$$x_j = x_{\lambda i} = x'_i, \quad F_j = F_{\lambda i} = F'_i,$$

su visais $i = 0, 1, \dots, N$. Tada, interpoliavimui imama duomenų aibė bus tokia

$$S' = \{(x'_i, F'_i) \mid i = 0, 1, \dots, N\}.$$

Fraktalinės interpoliacinės funkcijos (IFS atraktoriaus) generavimui pasinaudosime (lokaliaja) Koliažo teorema [14]. Ši teorema siejama su atveju, kai nagrinėjamam realaus pasaulio objektui (diskretizuotam jo analogui—duomenų aibei S) koliažas konstruojamas ne iš paties objekto sumažintų kopijų (2.1.2. skyrelis), bet iš to objekto fragmentų sumažintų kopijų.

Šiame kontekste (nagrinėjamasis objektas—duomenų aibė S) objekto fragmentus sudarys intervalai $[x'_{j-1}, x'_{j+1}]$, $j \in \{1, 2, \dots, N-1\}$, o jų sumažintų kopijų analogų bus ieškoma intervalų $[x'_{i-1}, x'_i]$, $i = 1, 2, \dots, N$, aibėje. Intervalus $[x'_{i-1}, x'_i]$ ($i = 1, 2, \dots, N$) vadinsime reikšmių srities intervalais, o intervalus $[x'_{j-1}, x'_{j+1}]$ ($j \in \{1, 2, \dots, N-1\}$)—apibrėžimo srities intervalais.

Fraktalinės interpoliacinės funkcijos (IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktoriaus) duomenų aibei S' generavimas suvedamas į geriausių (VKP prasme) porų „reikšmių srities intervalas – apibrėžimo srities intervalas“ paiešką.

Fiksuokime reikšmių srities intervalą $[x'_{i-1}, x'_i]$, $i \in \{1, 2, \dots, N\}$. Šiam intervalui randamas apibrėžimo srities intervalas $[x'_{j-1}, x'_{j+1}]$, $j \in \{1, 2, \dots, N-1\}$, kuriame apibrėžtos reikšmės $\{F_{\lambda(j-1)}, F_{\lambda(j-1)+1}, \dots, F_{\lambda(j+1)}\}$ geriausiai atvaizduojamos (konkrečios afiniosios transformacijos, tarkime, ω_i pagalba) į reikšmes $\{F_{\lambda(i-1)}, F_{\lambda(i-1)+1}, \dots, F_{\lambda i}\}$, apibrėžtas intervale $[x'_{i-1}, x'_i]$. Tokios afiniosios transformacijos (atvaizdžio ω_i) nustatymui ir realizavimui, visų pirma, dukart turi būti mažinamas su intervalu $[x'_{j-1}, x'_{j+1}]$ susietų reikšmių kiekis. Tai galima atlikti įvairiais būdais. Žemiau siūlomas variantas, kai dvi iš eilės einančios reikšmės keičiamos tų reikšmių aritmetiniu vidurkiu.

Naują (sumažintą) reikšmių, susietų su intervalu $[x'_{j-1}, x'_{j+1}]$, rinkinuką pažymėkime taip - $\{\tilde{F}_{\lambda(j-1)}, \tilde{F}_{\lambda(j-1)+1}, \dots, \tilde{F}_{\lambda j}\}$. Jeigu λ yra nelyginis skaičius, tai pastarosios reikšmės apskaičiuojamos tokiu būdu:

$$\begin{aligned} \tilde{F}_{\lambda(j-1)} &= F_{\lambda(j-1)} = F'_{j-1}, \quad \tilde{F}_{\lambda j} = F_{\lambda(j+1)} = F'_{j+1}; \\ \tilde{F}_{\lambda(j-1)+s} &= \frac{1}{2}(F_{\lambda(j-1)+2s-1} + F_{\lambda(j-1)+2s}), \quad \text{kai } s = 1, 2, \dots, (\lambda-1)/2, \end{aligned}$$

ir

$$\begin{aligned} \tilde{F}_{\lambda(j-1)+s} &= \frac{1}{2}(F_{\lambda(j-1)+2s} + F_{\lambda(j-1)+2s+1}), \quad \text{kai } s = (\lambda+1)/2, (\lambda+1)/2+1, \dots, \lambda-1; \\ \tilde{F}_{\lambda(j-1)+(\lambda-1)/2+1} &= F_{\lambda(j-1)+(\lambda-1)/2+1}. \end{aligned}$$

Jeigu λ yra lyginis skaičius, tai siūlome naudotis tokiomis išraiškomis:

$$\begin{aligned} \tilde{F}_{\lambda(j-1)} &= F_{\lambda(j-1)} = F'_{j-1}, \quad \tilde{F}_{\lambda j} = F_{\lambda(j+1)} = F'_{j+1}; \\ \tilde{F}_{\lambda(j-1)+s} &= \frac{1}{2}(F_{\lambda(j-1)+2s-1} + F_{\lambda(j-1)+2s}), \quad \text{kai } s = 1, 2, \dots, \lambda/2-1, \end{aligned}$$

ir

$$\begin{aligned} \tilde{F}_{\lambda(j-1)+s} &= \frac{1}{2}(F_{\lambda(j-1)+2s} + F_{\lambda(j-1)+2s+1}), \quad \text{kai } s = \lambda/2+1, \lambda/2+2, \dots, \lambda-1; \\ \tilde{F}_{\lambda(j-1)+\lambda/2} &= \frac{1}{3}(F_{\lambda j-1} + F_{\lambda j} + F_{\lambda j+1}). \end{aligned}$$

Toliau, konstruojama suspaudžiančioji (pražulnioji) afinioji transformacija

$$\omega_i = \omega_i(x, y) = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix},$$

„pervedanti“ tašką (x'_{j-1}, F'_{j-1}) į tašką (x'_{i-1}, F'_{i-1}) , ir tašką (x'_{j+1}, F'_{j+1}) - į tašką (x'_i, F'_i) . Pražulniąją transformaciją ω_i charakterizuojantys koeficientai randami pagal formules, pateiktas 1.3. skyrelyje, būtent:

$$a_i = \frac{x'_i - x'_{i-1}}{x'_{j+1} - x'_{j-1}}, \quad c_i = \frac{F'_i - F'_{i-1}}{x'_{j+1} - x'_{j-1}} - d_i \frac{F'_{j+1} - F'_{j-1}}{x'_{j+1} - x'_{j-1}},$$

$$e_i = \frac{x'_{j+1}x'_{i-1} - x'_{j-1}x'_i}{x'_{j+1} - x'_{j-1}}, \quad f_i = \frac{x'_{j+1}F'_{i-1} - x'_{j-1}F'_i}{x'_{j+1} - x'_{j-1}} - d_i \frac{x'_{j+1}F'_{j-1} - x'_{j-1}F'_{j+1}}{x'_{j+1} - x'_{j-1}}$$

Koeficiento d_i reikšmė apskaičiuojama, taikant mažiausiųjų kvadratų metodą, būtent:

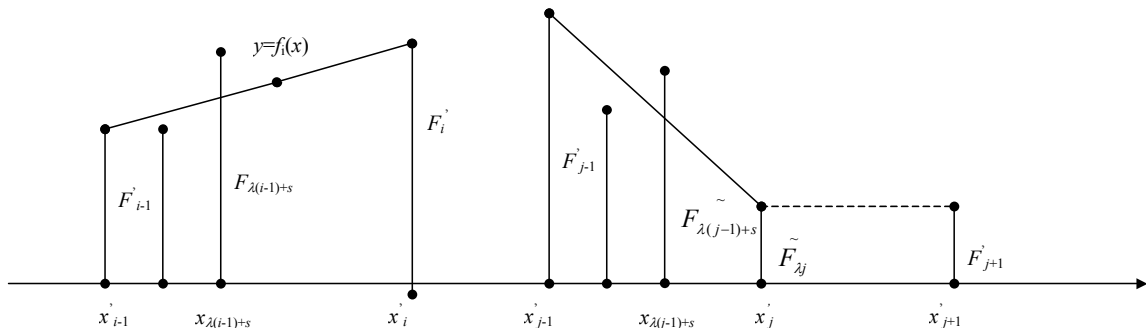
$$\sum_{s=1}^{\lambda-1} (F_{\lambda(i-1)+s} - d_i \tilde{F}_{\lambda(j-1)+s})^2 \rightarrow \min \quad (2.1)$$

Tai paprasčiausias minimizavimo uždavinys, todėl, prilyginę kairėje pusėje esančios išraiškos išvestinę (kintamojo d_i atžvilgiu) nuliui, nesunkiai randame

$$d_i = \frac{\sum_{s=1}^{\lambda-1} F_{\lambda(i-1)+s} \cdot \tilde{F}_{\lambda(j-1)+s}}{\sum_{s=1}^{\lambda-1} (\tilde{F}_{\lambda(j-1)+s})^2}$$

koks bebūtų $j \in \{1, 2, \dots, N\}$.

Būtina pabrėžti, jog konkrečiam reikšmių srities intervalui $[x'_{i-1}, x'_i]$ ($i \in \{1, 2, \dots, N\}$) fiksuojamas (kaip geriausias!) tas apibrėžimo srities intervalas (fragmentas) $[x'_{j-1}, x'_{j+1}]$, kuris suteikia minimumą (2.1) išraiškai (2.9 pav.).



2.9 pav. Koliažo organizavimo ypatumai

Kartais, siekiant padidinti gerų (VKP prasme) porų „reikšmių srities intervalas—apibrėžimo srities intervalas“ aptikimo tikimybę, fragmentams (intervalams) $[x'_{j-1}, x'_{j+1}]$ papildomai taikoma „atspindžio“ transformacija, t.y. reikalaujama, kad afinioji transformacija ω_i ($i \in \{1, 2, \dots, N\}$) pervestų tašką (x'_{j-1}, F'_{j-1}) į tašką (x'_i, F'_i) ir tašką (x'_{j+1}, F'_{j+1}) - į tašką (x'_{i-1}, F'_{i-1}) . Aišku, tokiu atveju atitinkamai turi būti koreguojamos afiniosios transformacijos koeficientų apskaičiavimui skirtos išraiškos.

Aprašytąją Koliažo organizavimo procedūrą galima nesunkiai apibendrinti bet kokiam nepersidengiančiųjų vienodo ilgio reikšmių srities intervalų skaičiui $r \in \{2, 3, \dots, N\}$ (aukščiau

išnagrinėtas atvejis, kai $r=N$) ir bet kokiam apibrėžimo srities intervalų skaičiui $p \in \{1, 2, \dots, N(1-2/r)+1\}$ (aukščiau išnagrinėtas atvejis, kai $p = N-1$). Toks apibendrinimas kartais leidžia gauti jeigu ne „geresnę“ fraktalinę interpoliacinę funkciją, tai bent paprastesnę (su mažesniu afinių transformacijų skaičiumi) IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_r\}$.

2.3. Interpoliuojamo objekto fraktalinės dimensijos įvertinimas

Vienas pagrindinių fraktalinio interpoliavimo privalumų yra tas, kad galima keisti fraktalinę generuojamų interpoliacinių funkcijų dimensiją, t.y. pastarąją galima parinkti taip, kad ji sutaptų su interpoliuojamų duomenų (objekto) fraktaline dimensija.

Pateikiame procedūrą, kuri mūsų manymu, garantuoja minėtojo privalumo įgyvendinimą.

Imkime pirminę duomenų aibę

$$S = \{(x_j, F_j) \mid j = 0, 1, \dots, M\}.$$

Tarkime, kad interpoliavimui naudojamas jos poaibis

$$S' = \{(x'_i, F'_i) \mid i = 0, 1, \dots, N\}.$$

čia $x'_i = x_{\lambda i} = x_j$, $F'_i = F_{\lambda i} = F_j$, su visais $i = 1, 2, \dots, N$; $\lambda \in \{2, 3, \dots\}$; $N < M$.

Naudojant tolygų interpoliavimo taškų pasiskirstymą bei anksčiau paminėtą fraktalinių interpoliacinių funkcijų generavimo (IFS pagalba) principą (1.3. skyrelis), belieka apibrėžti generavimo procese taikomų (tuo pačiu, ir INTERPO procedūroje) afinių transformacijų parametrų d_i ($i = 1, 2, \dots, N$) reikšmes. Pastebėsime dar kartą, jog šie parametrai keičia vertikalųjį mastelį vaizdavimo metu ir įtakoja galutinį interpoliavimo rezultatą—fraktalinę interpoliacinės funkcijos dimensiją.

Tarkime, kad duomenų aibę S' interpoliuojančioji funkcija sutampa su IFS $\{\mathbb{R}^2; \omega_1, \omega_2, \dots, \omega_N\}$ atraktoriumi A . Tada afinių (pražulniųjų) transformacijų išraiškos yra tokios (1.3. skyrelis):

$$\omega_i(x, y) = \begin{pmatrix} a_i & 0 \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix};$$

čia koeficientai a_i, c_i, e_i, f_i apskaičiuojami pagal (1.2) formules, pateiktas 1.3. skyrelyje. Vertikalųjį mastelį keičiančių koeficientų d_i reikšmių nustatymui siūlome tokią procedūrą.

Pirmiausia, užrašoma tiesės atkarpos, jungiančios taškus (x'_{i-1}, F'_{i-1}) ir (x'_i, F'_i) ($i \in \{1, 2, \dots, N\}$), lygtis (1.2.2. skyrelis):

$$y = f_i(x) = \frac{F'_i - F'_{i-1}}{x'_i - x'_{i-1}}(x - x'_{i-1}) + F'_{i-1}.$$

Kadangi į intervalą $[x'_{i-1}, x'_i]$ patenka, bendru atveju, daugiau nei du taškai x_j , $j \in \{0, 1, \dots, M\}$, apskaičiuojamas reikšmių

$$\{F'_{i-1} = F_{\lambda(i-1)}, F_{\lambda(i-1)+1}, F_{\lambda(i-1)+\lambda} = F_{\lambda i} = F'_i\}$$

vidurkis m_i bei jų išsibarstymo matas (dispersija) σ_i^2 tiesės $y = f_i(x)$ atžvilgiu. Tam panaudojamos išraiškos

$$m_i = \frac{1}{\lambda + 1} \sum_{s=0}^{\lambda} (F_{\lambda(i-1)+s} - y_s), \quad \sigma_i^2 = \frac{1}{\lambda + 1} \sum_{s=0}^{\lambda} (F_{\lambda(i-1)+s} - y_s)^2 - m_i^2; \quad (2.2)$$

čia $y_s = f_i(x_{\lambda(i-1)+s})$, $s = 0, 1, \dots, \lambda$.

Kadangi, esant fiksuotai (iš anksto apibrėžtai) fraktalinės interpoliacinės funkcijos (IFS atraktoriaus A) dimensijai (tarkime, kad $D(A)=D$) turi būti išpildoma sąlyga (1.3. skyrelis):

$$\sum_{i=1}^N |d_i| = N^{D-1},$$

tai koeficientų d_i ($i = 1, 2, \dots, N$) reikšmės („grubumo“ faktoriai), užtikrinančios lygybę $D(A)=D$, apskaičiuojamos taip:

$$d_i = \text{sign}(m_i) \cdot k \cdot \sigma_i^2$$

su visais $i = 1, 2, \dots, N$; čia

$$\text{sign}(m_i) = \begin{cases} 1, & m_i > 0, \\ 0, & m_i = 0, \\ -1, & m_i < 0; \end{cases}$$

ir $k = N^{D-1} / \sum_{i=1}^N \sigma_i^2$.

Be abejo, pastarosios išraiškos (reikšmių d_i radimui) netinka tuo atveju, kai fraktalinių interpoliacinių funkcijų radimui organizuojamas vienoks ar kitoks koliažas, aprašytas 2.2.2. skyrelyje. Pats koliažo organizavimo procesas leidžia teigti, jog fraktalinė IFS atraktoriaus (interpoliacinės funkcijos) dimensija bus artima nagrinėjamo realaus pasaulio objekto (diskretizuoto jo analogo— duomenų aibės S) fraktalinei dimensijai.

Panagrinėkime pavyzdį, pateiktą 2.1 skyrelyje.

Turime duomenų aibę, sudarytą iš penkių interpoliavimo taškų: $S' = \{(x'_i, F'_i) \in \{(3,0.3), (4,1.2), (7,0.1), (8,0.8), (10,0.3)\} \mid i = 0, 1, 2, 3, 4\}$. Tarkime, kad vertikalios mastelio keitimo parametro reikšmės yra: $s_i \in \{0.5, 0.2, 0.5, -0.3, 0\}$. Šiuos duomenis perdavę INTERPO, gausime 3.1 skyrelyje pateiktą interpoliavimo rezultatą (3.11 pav.). Pritaikykime duotai duomenų aibei šiame

skyrelyje išdėstyta metodiką ir patikrinkime, ar patikslintos parametro s reikšmės, gautos interpoliuojant su INTERPO ir d reikšmės, gautos taikant naująją metodiką sutampa.

Diskretizuokime kiekvieną duotą intervalą, raskime nuokrypius, apskaičiuokime vidurkius m_i bei jų išsibarstymo matą (dispersiją) σ_i^2 tiesės $y = f_i(x)$ atžvilgiu, kai $i \in \{1,2,3,4\}$. Tam panaudokime anksčiau pateiktas išraiškas (2.2). Gausime, kad

$$\begin{aligned} m_1 &= 0,255555555; & m_2 &= 0,111111111; & m_3 &= 0,315384615; & m_4 &= -0,06; \\ \sigma_1 &= 0,26034165; & \sigma_2 &= 0,105409255; & \sigma_3 &= 0,260915547; & \sigma_4 &= 0,260768096; \\ \sigma_{1p} &= 0,245452467; & \sigma_{2p} &= 0,099380799; & \sigma_{3p} &= 0,250679549; & \sigma_{4p} &= 0,233238075. \end{aligned}$$

Tada d reikšmės atitinkamai lygios: $d_1=0,29$; $d_2=0,118$; $d_3=0,29$; $d_4=-0,29$.

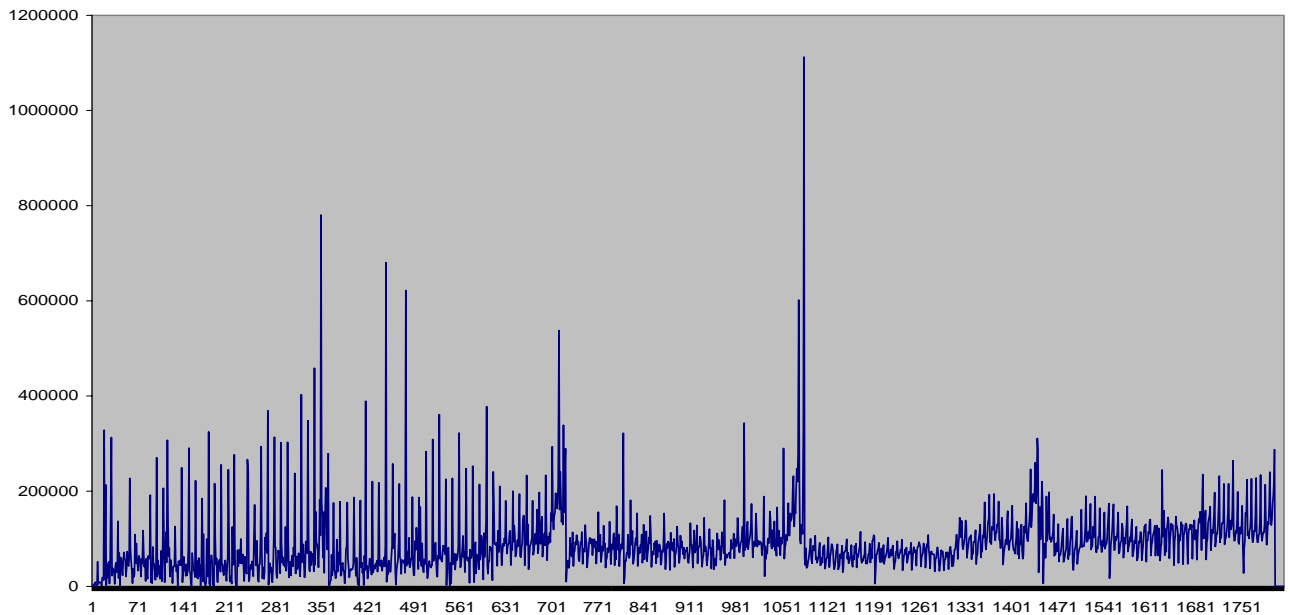
Analogiškai, s reikšmės, gautos interpoliuojant su INTERPO, yra lygios: $s_1 = 0.33$; $s_2 = 0.13$; $s_3 = 0.33$; $s_4 = -0.20$. Skirtumai nežymūs ir gauti tik dėl netikslių nuokrypių matavimų rezultatų, todėl galime daryti prielaidą, kad apskaičiuodami d reikšmes ir naudodami jas kaip s reikšmes mes gausime ne tik tikslesnį rezultatą, bet ir apčiuopiamai sumažinsime fraktaliniam interpoliavimui (INTERPO pagalba) skirtas laikines sąnaudas.

2.4. Eksperimento rezultatų analizė

Fraktalinių interpoliacinių funkcijų praktinio panaudojimo eksperimentas buvo atliktas naudojant fraktaliniam interpoliavimui skirtą priemonę INTERPO. Tyrimams buvo taikomas tolygus interpoliavimo taškų išdėstymas bei koliažas, kai „reikšmių srities intervalų“ skaičius lygus $[M/\lambda]$, o „apibrėžimo srities intervalų“ skaičius lygus $r=1$ (čia M – pradinės duomenų aibės galia). Eksperimentas taikytas realiai egzistuojančios įmonės prekybos objektui. Buvo siekiama apdoroti gautas reikšmių lenteles (duomenų aibes), sudarant įvairius interpoliavimo taškų masyvus (t.y. parenkant įvairias parametro λ reikšmes), nusakančius generuojamas FIF.

Tyrimo metu panaudoti duomenys – realios įmonės, prekiaujančios drabužiais, apyvartos kitimas. Buvo analizuojami pokyčiai per 2000 01 01 – 2005 04 01 laikotarpį.

Kadangi įmonės dienos apyvarta priklauso nuo daugelio sunkiai įvertinamų dalykų (nuo naujų kolekcijų pristatymo į parduotuves, nuo pirkėjų aktyvumo (mėnesio pradžioje, kai dažniausiai mokami atlyginimai pastebimas didesnis pirkėjų aktyvumas), nuo modelių populiarumo, nuo išpardavimų, nuo konkurentų vykdomų akcijų, galų gale nuo sezono oro – šalta vasara ir šilta žiema smarkiai sužlugdo sezoninių prekių pardavimus) galime daryti prielaidą, kad duomenys yra fraktalinės kilmės.



2.10 pav. Įmonės apyvartos kaitos diagrama

Kita vertus, pradinis duomenų rinkinys labai didelis (1800 dienų), todėl tam, kad galėtume naudotis INTERPO programa, būtina atlikti duomenų atrinkimą (duomenų praretinimą).

Galimi įvairūs interpoliavimo taškų parinkimo būdai:

- 1) naudojant žingsnį λ ($\lambda \in \{2,3,\dots\}$);
- 2) naudojant suvidurkintas reikšmes;
- 3) naudojant suglodintus duomenis.

1) Duomenų rinkinio formavimas naudojant pasirinktą žingsnį λ .

Remiantis turimais (pradiniais) duomenimis, pirmiausiai suformuojamas duomenų rinkinys pagal pasirinktą žingsnį λ . Sudarant duomenų failą (orientuotą į INTERPO panaudojimą), reikia įrašyti vertikalųjį mastelį keičiančius parametrus s_i . Kaip jau buvo užsiminta 2.3 skyrelyje, tam reikia apskaičiuoti d_i („grubumo“, faktorių įverčius) ir naudoti juos vietoj s_i reikšmių. Kad apskaičiuoti d_i , reikia žinoti interpoliuojamo objekto fraktalinę dimensiją. Deja tai padaryti sunku, kadangi FRACDIM programa nepriima kitokių bylų, kaip tik *.pcx bylų, sukurtų naudojantis paketu FV. Todėl vadinamosios „grubumo“ reikšmės s_i buvo apskaičiuotos fiksavus keletą skirtingų fraktalinių dimensijų, tuo pačiu, pasirinktam žingsniui λ buvo sudaryta keletas duomenų rinkinių.

Mūsų atveju (2.10 pav.pateiktiems duomenims) pasirinkus žingsnį $\lambda=19$, buvo suformuoti duomenų rinkiniai:

dim1: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=1$;

dim2: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=1,2$;

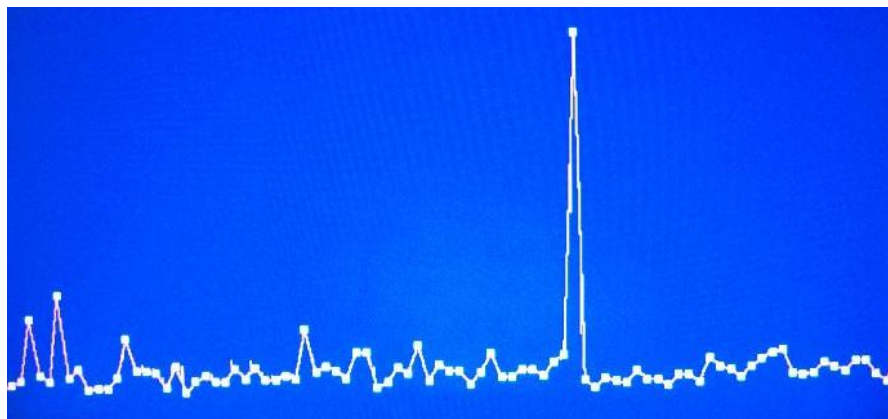
dim3: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=1,4$;

dim4: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=1,6$;

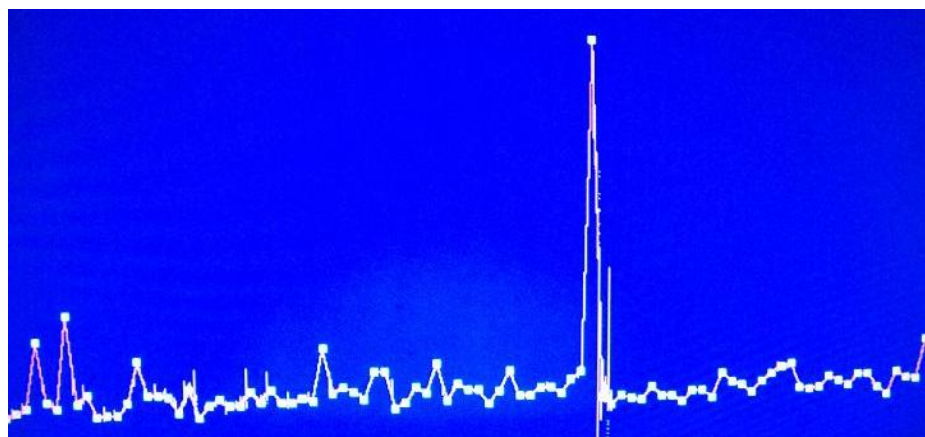
dim5: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=1,8$;

dim6: reikšmės d_i skaičiuojamos, kai fraktalinė dimensija $D=2$.

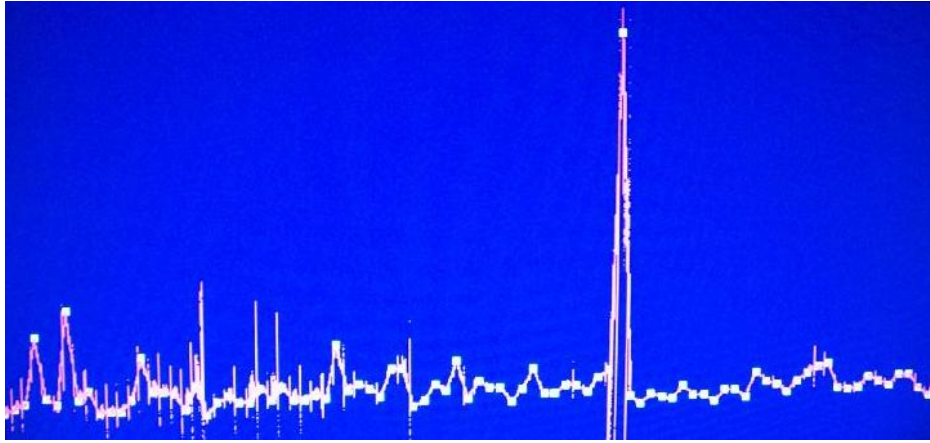
Kiekvienu atveju, gautąsias reikšmes d_i priskyrus „grubumo“ faktoriams s_i ir pasinaudojus INTERPO priemone, buvo gautos (sugeneruotos) fraktalinės interpoliacinės funkcijos (FIF). Pastarųjų grafikai pateikti 2.11 – 2.16 paveikslėliuose.



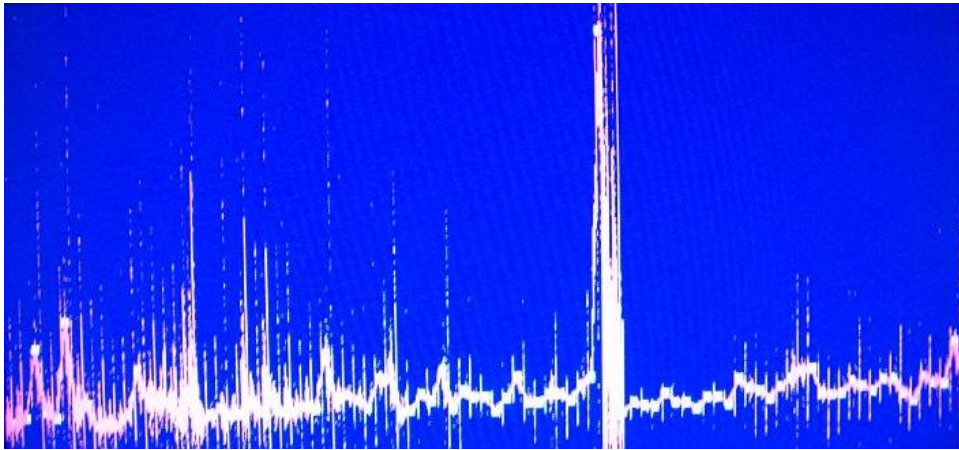
2.11 pav. FIF kreivė „dim1“ duomenims;
Reikšmių sk. 93; skalė 1×900000 ; $D=1$



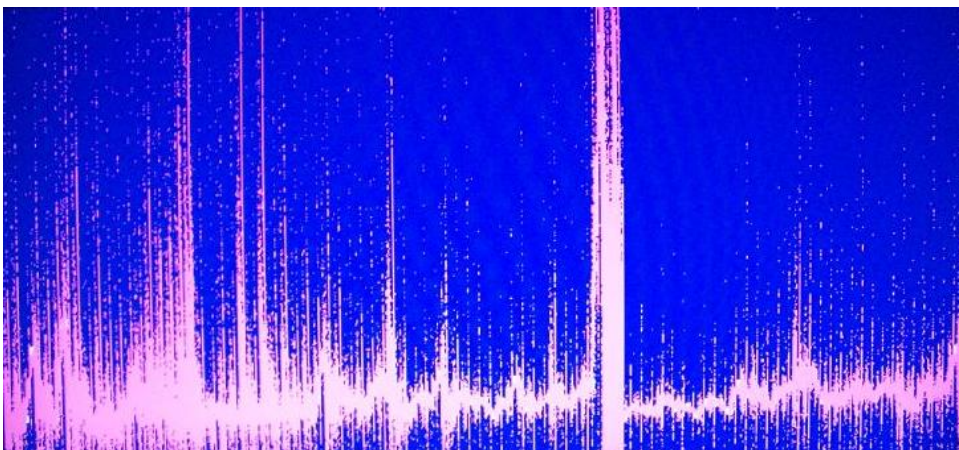
2.12 pav. FIF kreivė „dim2“ duomenims;
Reikšmių sk. 93; skalė 1×900000 ; $D=1,2$



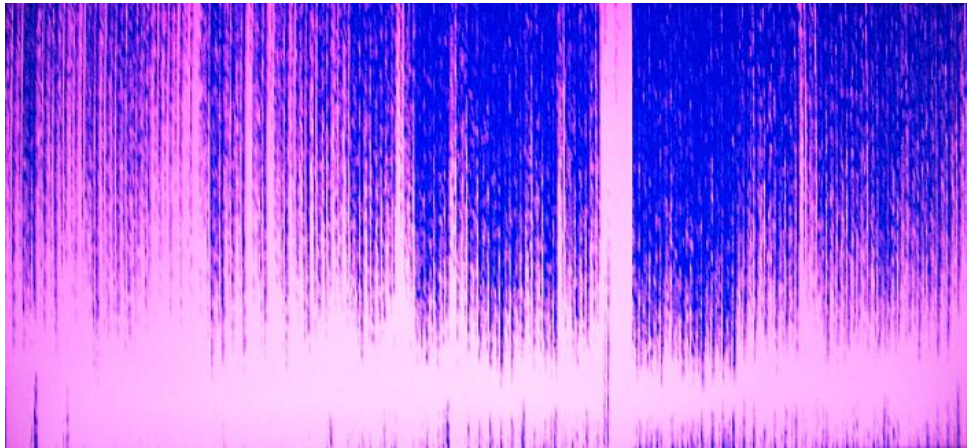
2.13 pav. FIF kreivė „dim3“ duomenims
Reikšmių sk. 93; skalė 1x900000; D=1,4



2.14 pav. FIF kreivė „dim4“ duomenims
Reikšmių sk. 93; skalė 1x900000; D=1,6



2.15 pav. FIF kreivė „dim5“ duomenims
Reikšmių sk. 93; skalė 1x900000; D=1,8

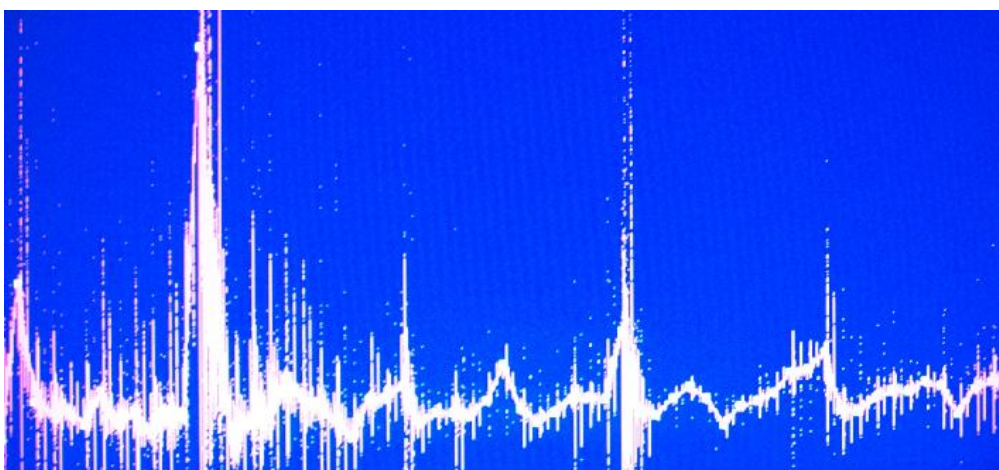


2.16 pav. FIF kreivė „dim6“ duomenims
Reikšmių sk. 93; skalė 1x900000; D=2

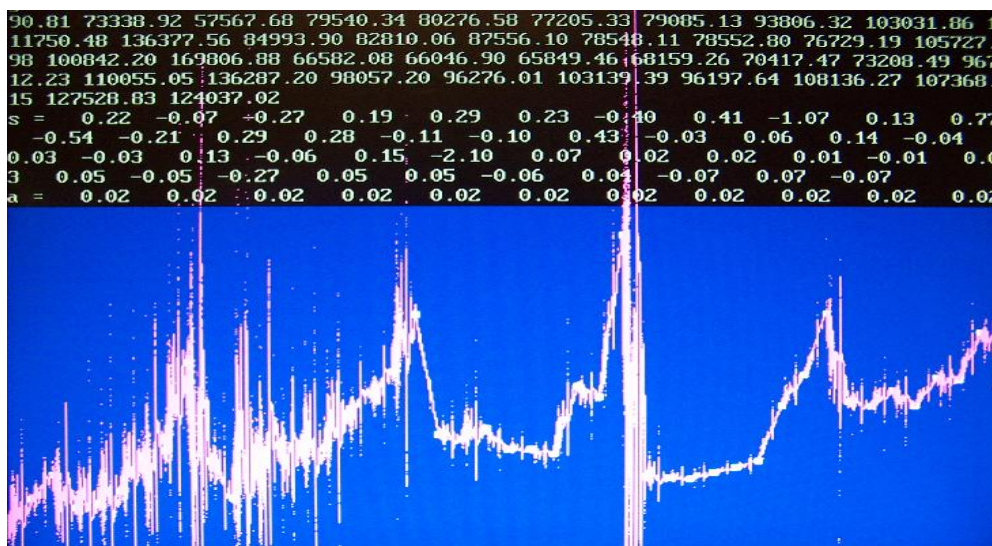
Vizualiai vertindami palyginame gautus rezultatus (2.11 – 2.16 pav.) su apyvartos kaitos grafine diagrama (2.10 pav.), matome, kad artimiausia analizuojamai kreivei yra FIF kreivė, gauta iš duomenų „dim4“ (2.14 pav.). Vadinasi, mūsų interpoliuojamo objekto fraktalinė dimensija $D \approx 1,6$. Ją ir naudosime tolimesniame tyrime.

Kitas eksperimentas buvo siejamas su žingsnio λ didinimu. Buvo siekiama mažinti interpoliavimui skirtos duomenų aibės galią, ir žiūrima, kokią tai įtaką turi galutiniam rezultatui (FIF kreivės formai)

Atskiru atveju, kai $\lambda=29$, atitinkamai modifikavus „dim4“ duomenis, buvo gauta FIF kreivė, parodyta 2.17 pav.



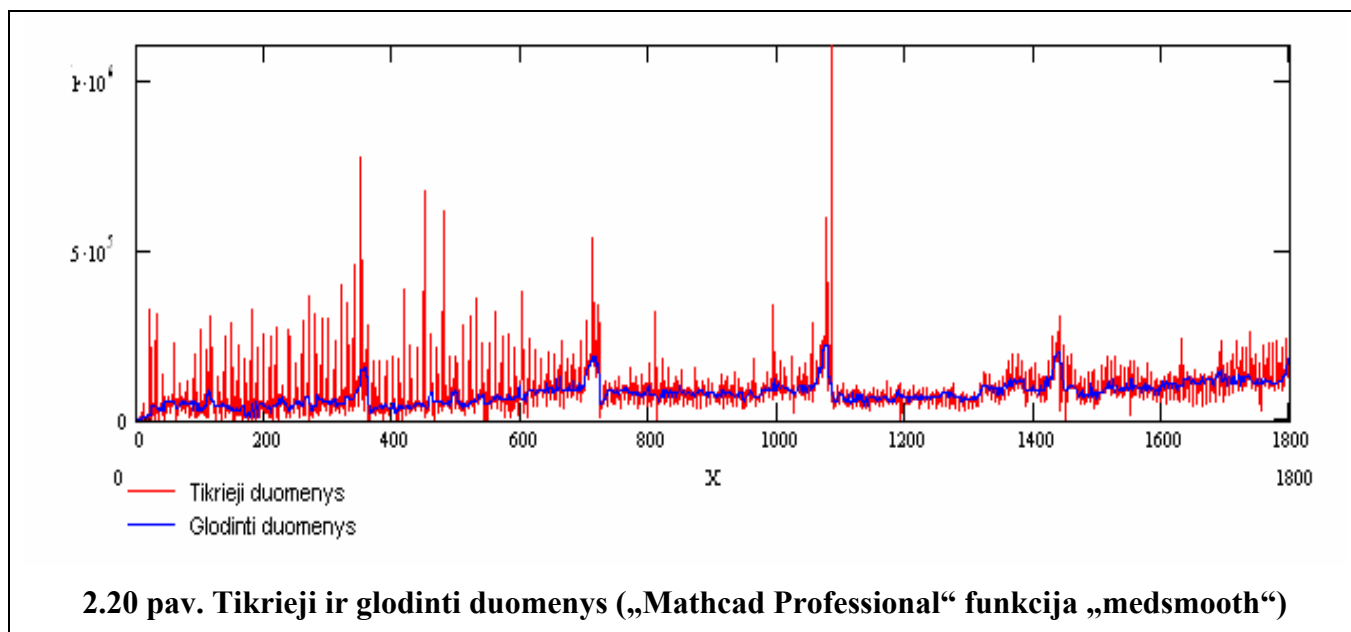
2.17 pav. FIF kreivė („dim 4“; $\lambda=29$);
Reikšmių sk. 61; skalė 1x900000; D=1,6



2.19 pav. FIF kreivė („dim4“; $\lambda=39$; suvidurkintos reikšmės)
Reikšmių sk. 45; skalė 1×900000 ; $D=1,6$

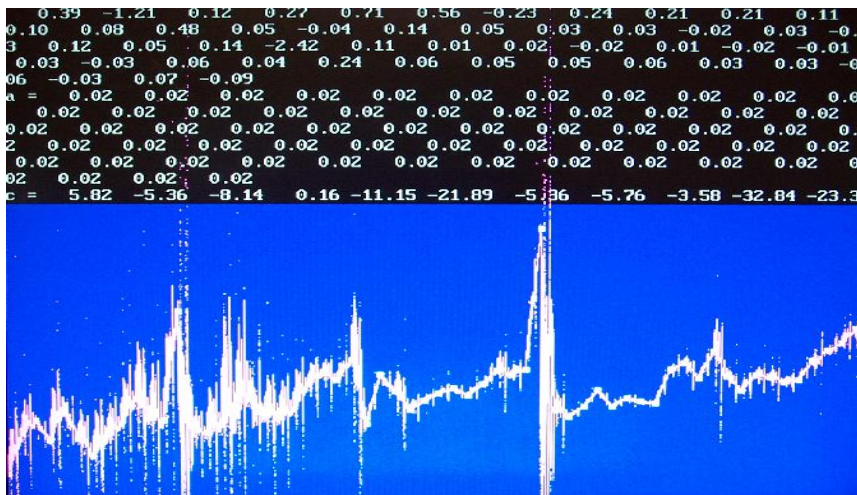
3) Duomenų aibės formavimas, naudojant pasirinktą žingsnį λ , bei glodintas duomenų reikšmes.

Pradinių duomenų glodinimui buvo panaudota matematinės programos „Mathcad Professional“ funkcija „medsmooth“, kuri skaičiuoja naujas reikšmes slenkančios medianos principu. 2.20 pav. pateikti tikrieji ir glodinti (pritaikius „Mathcad Professional“ funkciją „medsmooth“) duomenys.



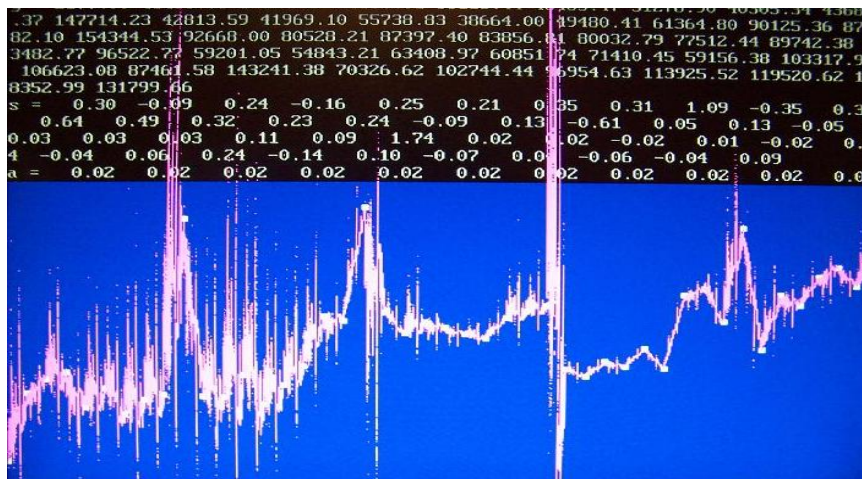
2.20 pav. Tikrieji ir glodinti duomenys („Mathcad Professional“ funkcija „medsmooth“)

Pasirenkame žingsnį $\lambda=29$. Imame naujai suformuotus „dim4“ duomenis. Rezultatas pateiktas 2.21 paveiksle.



2.21 pav. FIF kreivė („dim4“; $\lambda=29$; „glodinti“ duomenys);
Reikšmių sk. 61; skalė 1x900000; D=1,6

Pasirenkame žingsnį $\lambda=39$. Imame naujai suformuotus „dim4“ duomenis. Rezultatas pateiktas 2.22 paveiksle.



2.22 pav. Gauta FIF kreivė („dim4“; $\lambda=39$; „glodinti“ duomenys);
Reikšmių sk. 45; skalė 1x900000; D=1,6

Pasirenkame žingsnį $\lambda=49$. Imame naujai suformuotus „dim4“ duomenis. Rezultatas pateiktas 2.23 paveiksle.

Pastebime, kad geriausias rezultatas gaunamas, kai $\lambda=29$. Toliau didinant žingsnį, FIF kreivė darosi vis mažiau (vizualiai) panaši į interpoliuojamą objektą (2.10 pav.).

Vis dėlto, galima būtų teigti, jog šis duomenų aibės formavimo būdas pasiteisina labiausiai, kadangi su mažiausiu interpoliavimo taškų skaičiumi ($\lambda=29$, t.y. reikšmių skaičius 61) gaunamas gana neblogas rezultatas.



2.23 pav. FIF kreivė („dim4“; $\lambda=49$; „glodinti“ duomenys);
Reikšmių sk. 35; skalė 1×9000000 ; $D=1,6$

Atlikti eksperimentai parodė, jog tinkamai parinkus fraktaliniam interpoliavimui skirtos duomenų aibės formavimo metodą, galima interpoliuoti informaciją apie įmonės veiklos rezultatus per vienokį ar kitokį laikotarpį. Gautos FIF kreivės gana artimos realioms duomenims (interpoliuojamam objektui). Galima jau būtų kalbėti ir apie ekstrapoliavimo (prognozavimo) uždavinių sprendimą. Kita vertus, automatizavus koliažu grindžiamų FIF kreivių generavimo procedūras (2.2 skyrelis), galima būtų išsamiau iširti ir įvertinti fraktalinio interpoliavimo privalumus, bei gauti , neabejotinai, geresnius rezultatus, interpoliuojant vienokį ar kitokį fraktalinės prigimties objektą.

IŠVADOS

1. Išanalizavus fraktalų generavimo bei jų panaudojimo principus, buvo ištirti ir nustatyti reikalavimai fraktalinio interpoliavimo priemonėms.
2. Eksperimentų rezultatai parodė, kad INTERPO—priemonė, kurios pagalba galima generuoti FIF įvairių tipų duomenims, jų analizei. Iškilusios problemos (vaizdumo, rezultatų išsaugojimo galimybių stoka) leidžia daryti išvadą, kad sėkmingesniai priemonės taikymui ją būtina atnaujinti, priderinant prie naujų kompiuterių techninių parametrų ir naudojamų OS (būtina pakeisti vartotojo terpę bei išplėsti duomenų sąrašą).
3. Sudaryta ir pasiūlyta metodika, leidžianti priderinti FIF kreivės dimensiją prie analizuojamo (interpoliavimui skirtą) objekto fraktalinės dimensijos.
4. Teoriniai ir eksperimentiniai rezultatai rodo, kad suformuoti interpoliavimo taškų masyvai galėtų būti naudojami ekstrapoliavimui (kreivės-fraktalo perspektyvos generavimui (prekybinės prognozės, ligos eigos prognozės, meteorologinės, biržų prekybos bei kt. prognozės)). Kita vertus, tokie masyvai leistų sutaupyti kompiuterio resursus – kreivės (fraktalo) aprašui būtų naudojama mažiau atminties (tarkim, objektas, nusakomas daugiau nei 1400 įrašais, gali būti nusakytas 100, ar bent 700, įrašais). Arba, FIF kreivės, nusakančios širdies ritmogramą, leistų stebėti tarpinius (šiai nefiksuojamus duomenis) ir laiku diagnozuoti širdies ritmo sutrikimus, kurie susirgimo pradžioje etape šiai jau būna gan reti ir ne visada pavyksta juos užfiksuoti įprastais metodais.
5. FIF kreivių generavimui skirtą koliažo organizavimas ir jo realizacija yra imlūs laiko sąnaudoms, todėl perspektyvus ir sėkmingas jo taikymas įmanomas tik automatizavus visą procesą.

LITERATŪRA

1. M. BARNSLEY, *Fractals Everywhere*, Second edition, Academic Press Professional, Cambridge 1993.
2. W. O. COCHRAN, J. C. HART PATRICK, J. FLYNN, *On Approximating Rough Curves With Fractal Functions* School of EECS Washington State University Pullman, WA
3. *Fractal Vision: Put Fractals to Work For You* Authors/ Dick Oliver: Publisher; 1992.— 485 p.
4. J. FEDER, *Fractals* Plenum Press New York 1988
5. N. KOSINOV, *Fraktal protona—novy predstavitel v semeistve fraktalov.*
6. N. KOSINOV, *Protono ir deitrono fraktalai – mikropasaulio fraktalų atstovai*
7. B. B. MANDELBROT, *The Fractal Geometry of Nature.* 1977
8. D. OLIVER, *Fractal Vision*, Sams Publishing (A division of Prentice Hall Computer Publishing, Carmel, Indiana). 1992.
9. W. PAULSON, *Introduction to Fractals and IFS*; 1998
10. H. PEITGEN, P. RICHTER, *Krasota fraktalov. Obrazy kompleksnykh dinamičeskikh sistem.—* M.:Mir, 1993.
11. H. PEITGEN, P. RICHTER, *The Beauty of Fractals*, Springer—Verlag, Berlin, New York. 1986.
12. P. TANNENBAUMAS, R. ARNOLDAS, *Kelionės į šiuolaikinę matematiką*; V.: TEV, 1995.
13. V. TARASENKO, *Ossobennosti vvedeniya poniatija fraktala.*
14. J. VALANTINAS, *Fraktalinė geometrija.—* K.: Technologija, 1999.—p.10— 69,121— 153
15. B. VILJAMS, *Trading Chaos.*

1 PRIEDAS INTERPO ALGORITMAS C KALBA

```

/* INTERPO.C
*
* Ši programa skirta fraktalinės interpoliacinės funkcijos (FIF)
* kreivės generavimui, kai duomenys skaitomi iš ASCII failo
* duomenų failo pavyzdys: failas SAMPLE.FIP
*
*/

#include <stdio.h>    /* Įvedimo/išvedimo standartas */
#include <stdlib.h>
#include <string.h>   /* strcpy */
#include <math.h>     /* mat f-jos */
#define MAIN        /* Pagrindinė. Dydžio D skaičiavimui */
#include "fv.h"      /* Instrukcija */
#define drawclr 7    /* Kreivės linijos spalva */
#define pointclr 11  /* duotų taškų spalva */
#define gridclr 2    /* grafo linijos spalva */
#define NPOINTS 100 /* Maksimalus duomenų kiekis */
float x[NPOINTS], y[NPOINTS], /* duomenų failo duomenys */
      scale[NPOINTS],          /* y mastelio keitimo parametras */
      a[NPOINTS - 1], b, c[NPOINTS - 1], /* IFS kodas */
      e[NPOINTS - 1], f[NPOINTS - 1];
void main(int nargs, char **arg) /* komandinės eilutės paruošimas */
{ int vidmode, i, j;           /* monitoriaus tipo nurodymas */
  float x1, y1, x2, y2,       /* brėžti kreivei */
        xscale, yscale,      /* monitoriaus taškų skalė */
        maxx, maxy, minx, miny, /* max ir min x bei y */
        dim, sum, ratio,     /* dimensijos skaičiavimui */
        xgrid, ygrid;       /* grafo formavimas pagal įvestus duomenis */
  int xoffset, yoffset, yadjust, /* ekrano pozicionavimas */
      npoints;                /* duomenų taško eilė duomenų faile */
  FILE *diskfile;            /* failo vardui */
  char filename[20];

```

```

vidmode = pickmode(detectmode()); /* video moduliui */
if (nargs > 1) strcpy(filename, arg[1]);
else /* jei failo vardas kom eil nenurodytas */
{ printf("\n\n");
  printf("Programa skirta fraktalinei kreivei generuoti"
    " pagal duomenis\n");
  printf("iš tekstinio failo: taško x,y. Duomenų įvedimo "
    "pavyzdys yra faile SAMPLE.FIP.");
  printf("\n\nĮveskite duomenų tipo pavadinimą: ");
  scanf("%s", filename);
}
if (nargs > 2) dim = atof(arg[2]);
else /* jei dimensija nenurodyta */
{ printf("\n\nPasirinkite fraktalinės dimensijos "
  "reikšmę (iš intervalo [1;2]: ");
  scanf("%f", &dim);
}
if ((dim > 2.0) || (dim < 1.0)) /* dimensija nurodyta tinkamai? */
{ printf("Fraktalinė dimensija-skaičius "
  "iš intervalo [1;2] (pvz. 1.2).");
  exit(0);
}
if ((diskfile = fopen(filename, "r")) == NULL)
{ printf("Klaidingai nurodėte failo vardą %s.", filename);
  exit(0);
}
fscanf(diskfile, "%d", &npoints); /* duomenų nuskaitymas */
fscanf(diskfile, "%f %f", &xgrid, &ygrid);
for (i = 0; (i < npoints) && (i < NPOINTS); i++)
  fscanf(diskfile, "%f %f %f", x + i, y + i, scale + i);
fclose(diskfile); /* kol nėra klaidų!! */
if (setvidmode(vidmode, 1) != 1)
{ printf("\nPakeiskite monitoriaus tipą.\n");
  exit(0);
}

```

```

/* vaizdo formavimas */
yadjust = scrny / 20;      /* tinklelis */
maxx = x[0]; minx = x[0];
maxy = y[0]; miny = y[0];
for (i = 1; i < npoints; i++) /* plačiausia/siauriausia */
{
  if (x[i] > maxx) maxx = x[i];
  if (y[i] > maxy) maxy = y[i];
  if (x[i] < minx) minx = x[i];
  if (y[i] < miny) miny = y[i];
}
xscale = scrnx / (maxx - minx); /* skaič. x, y */
yscale = (scrny / (maxy - miny)) / 2;
xoffset = - (int) (minx * xscale); /* perkėlimas */
yoffset = scrny - yadjust + (int) (miny * yscale);
/* Skaičiuojami transformacijų
   koeficientai ir vertik. mastelį keič. parametro suma */
b = x[npoints - 1] - x[0]; /* b reikšmės */
sum = 0.0;      /* v.m.k. parametro sum */
for (i = 0; i < npoints - 1; i++) sum += fabs(scale[i]);
ratio = pow(npoints - 1, dim - 1) / sum;
for (i = 0; i < npoints - 1; i++) /* nustatomas IFS kodas */
{
  scale[i] *= ratio;
  a[i] = (x[i + 1] - x[i]) / b;
  e[i] = (x[npoints - 1] * x[i] - x[0] * x[i + 1]) / b;
  c[i] = (y[i + 1] - y[i] - scale[i] *
          (y[npoints - 1] - y[0])) / b;
  f[i] = (x[npoints - 1] * y[i] -
          x[0] * y[i + 1] -
          scale[i] *
          (x[npoints - 1] * y[0] -
           x[0] * y[npoints - 1])) / b;
}

/* Rezultatų spausdinimas */
printf("%s %d points, "

```

```

    "graph scale = %6.2f x %6.2f Dimension = %f",
    filename, npoints, xgrid, ygrid, dim);
printf("\nx =");
for (i = 0; i < npoints; i++) printf(" %6.2f", x[i]);
printf("\ny =");
for (i = 0; i < npoints; i++) printf(" %6.2f", y[i]);
printf("\ns =");
for (i = 0; i < npoints - 1; i++) printf(" %6.2f",
    scale[i]);

printf("\na =");
for (i = 0; i < npoints - 1; i++) printf(" %6.2f", a[i]);
printf("\nc =");
for (i = 0; i < npoints - 1; i++) printf(" %6.2f", c[i]);
printf("\ne =");
for (i = 0; i < npoints - 1; i++) printf(" %6.2f", e[i]);
printf("\nf =");
for (i = 0; i < npoints - 1; i++) printf(" %6.2f", f[i]);
    getch();
settextmode();
    /* brėžiama kreivė */
setcolor(gridclr);
xgrid = xgrid * xscale; ygrid *= yscale;
for (x1 = (float) xoffset; x1 < scrnx; x1 += xgrid)
    drawline((int) x1, scrny / 2 - yadjust * 2,
        (int) x1, scrny - 1);
for (y1 = (float) yoffset;
    y1 >= (float) scrny / 2 - yadjust * 2;
    y1 -= ygrid)
    drawline(0, (int) y1, scrnx - 1, (int) y1);
/* tinklelis */
for (i = 0; i < npoints; i++)
{   x1 = x[i] * xscale + xoffset;
    y1 = yoffset - y[i] * yscale;
    fillrect((int) x1 - 2, (int) y1 - 2,
        (int) x1 + 2, (int) y1 + 2, pointclr);
}

```

```
}
/* kreivė pateikiama ekrane*/
x1 = (float) xoffset;          /* pradžia */
y1 = (float) yoffset;
while(!kbhit())              /* ciklas */
{ j = rand() % (npoints - 1); /* chaosas */
  x2 = a[j] * x1 + e[j];      /* atraktorius */
  y2 = c[j] * x1 + scale[j] * y1 + f[j];
  x1 = x2, y1 = y2;
  putpixel((int) (x1 * xscale + xoffset),
           (int) (yoffset - y1 * yscale), drawclr);
}
/* pabaiga */
getch();
settextmode();
closedown();
}
```


2 PRIEDAS MATCAD PROGRAMŲ TEKSTAI