VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
MATEMATIKOS METODIKOS KATEDRA

Modestas Rimkus

LOGO GALIMYBĖS FRAKTALAMS KURTI

Magistro baigiamasis darbas

Vadovė
prof. dr. Valentina Dagienė

VILNIUS   2006

Turinys

Įvadas

Fraktalai mus supa visur. Tai debesys, kalnai, liepsnos liežuviai, medžiai ar sudėtinga mūsų kraujagyslių sistema. Vieni jų nuolat kinta, kiti išlaiko susiformavusią struktūrą. Tokie kasdieniai ir, atrodytų, paprasti dalykai per pastaruosius trisdešimt metų tapo intensyvių tyrinėjimų objektu ir nuostabūs meniški vaizdai tapo sudėtingomis matematinėmis formulėmis. Tačiau tas sudėtingumas nemažina noro toliau domėtis fraktalais ir už jų slypinčia matematika. Priešingai, fraktalai – puiki priemonė skatinti domėjimąsi matematika. Ypač jei juos galima pateikti paprastomis, kiekvienam suprantamomis priemonėmis, integruoti į kitus dalykus. Viena tokių priemonių – Logo.

Logo – tai mokymo filosofija, o taip pat ir programavimo kalbų šeima, naudojanti ir perteikianti konstruktyvistines šios filosofijos idėjas. Tradicinė Euklido geometrija pagrįsta abstrakcijomis: taško negalima išmatuoti, tiesė neturi pločio. Tą sunku suprasti jaunesniems mokiniams. Logo programose naudojamas vėžliukas – konkretus objektas, kurį galima matyti ekrane ir valdyti. Logo geometrija tinkama ir jaunesniems, ir vyresniems mokiniams. Žinoma, ja nereikia pakeisti tradicinės geometrijos, tačiau tai puiki priemonė pradėti mokytis geometrijos ir matematikos apskritai.

Logo geometrija skatina naudoti kompiuterius mokymui ir mokymuisi. Jaunesnieji mokiniai, paprastomis komandomis valdydami vėžliuką, o vyresnieji ir rimčiau programuodami Logo kalba, gali perprasti ne tik darbo kompiuteriu principus, bet kartu susidomėti ir kitais dėstomais dalykais, nagrinėti kitų dalykų objektus Logo vėžliuko pagalba. Logo – puiki priemonė sudominti mokinius matematika, paprastai ir vaizdžiai pademonstruoti nepaprastą fraktalų grožį.

Šio darbo tikslai – ištirti paprasčiausius fraktalus, išnagrinėti fraktalų savybių realizavimo ir fraktalų atvaizdavimo Logo priemonėmis galimybes, sudaryti fraktalų dėstymo mokiniams metodiką naudojant Logo. Šiems tikslams įgyvendinti iškeliami darbo uždaviniai – užrašyti matematinius fraktalų sudarymo algoritmus Logo kalba bei pateikti keletą fraktalų realizavimo Logo kalba pavyzdžių.

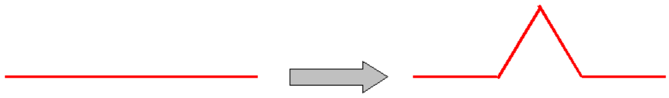
Darbą sudaro dvi dalys. Pirmojoje pateikiamas fraktalų apibrėžimas, jų savybės, trumpa fraktalų istorija, detaliau nagrinėjami žinomi fraktalai – Julijaus ir Mandelbroto aibės. Antrojoje dalyje nagrinėjamos fraktalų vaizdavimo Logo priemonėmis galimybės, pateikiami Logo programų fraktalams kurti pavyzdžiai.

1. Bendroji dalis

1.1. Kas yra fraktalas?

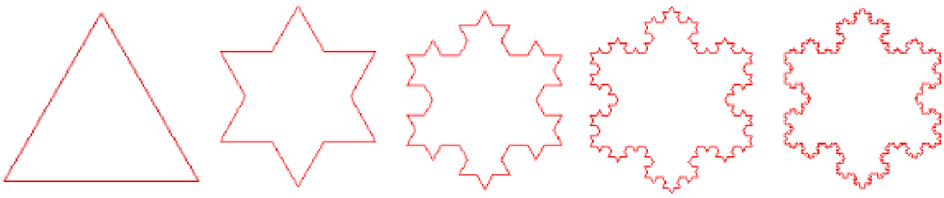
Prieš apibrėždami, kas yra fraktalas, susipažinsime su netradicine simetrijos rūšimi, vadinama *mastelio simetrija*. Mastelio simetriją paaiškinsime keletu pavyzdžių [7].

Imkime bet kokio dydžio lygiakraštį trikampį. Laikykime, kad trikampio kraštinių ilgiai yra vienetiniai, o trikampio plotą pažymėkime raide A. Kiekvieną trikampio kraštinę pakeiskime 1 pav. pavaizduota laužte, sudaryta iš vienodo ilgio atkarpų.



1 pav. Kiekvieną atkarpą (trikampio kraštinę) keičiame laužte

Gauname žvaigždės formos figūrą, turinčią 12 kraštinių, kurių kiekvienos ilgis yra 1/3. Atlikdami kitą žingsnį, kiekvienai iš 12 naujosios figūros kraštinių kartojame tą pačią 1 pav. pavaizduotą procedūrą: atkarpą keičiame laužte. Gautoji figūra jau turi 48 kraštines, ir kiekvienos ilgis yra 1/9. Atlikdami tolesnius žingsnius, tą pačią procedūrą kartojame be galo. Tokiu būdu gauname geometrinį objektą, kurį atrado Helge von Kochas[1]. Jis vadinamas Kocho snaige (2 pav.).



2 pav. Pirmieji 5 Kocho snaigės sudarymo žingsniai

Žinoma, galutinio bei tobulo Kocho snaigės piešinio pavaizduoti neįmanoma, nes tikrajai Kocho snaigei sudaryti reikėtų be galo daug žingsnių.

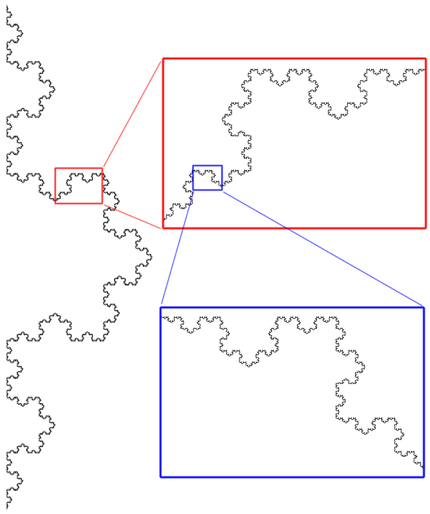
Panagrinėkime kitą pavyzdį. Imkime bet kokį juodą trikampį, sujunkime šio trikampio kraštinių vidurio taškus. Gausime keturis trikampius, panašius į pradinį. Pašalinkime vidurinįjį trikampį. Su kiekvienu iš likusių trijų trikampių atliekame tą pačią procedūrą: sujungę atkarpomis kiekvieno trikampio kraštinių vidurio taškus, iš kiekvieno trikampio, padalinto į keturis į jį panašius trikampius, pašaliname viduriniuosius. Atlikę be galo daug tokių žingsnių, gausime geometrinį objektą, kurį pirmasis pasiūlė V. Sierpinskis[2]. Šis objektas vadinamas Sierpinskio nėriniu (3 pav.).

1 Niels Fabian Helge von Koch (1870–1924) – švedų matematikas.
2 Wacław Franciszek Sierpiński (1882–1969) – lenkų matematikas, plėtojęs aibių teoriją, skaičių teoriją, funkcijų teoriją ir topologiją.

3 pav. Pirmieji 5 Sierpinskio trikampio sudarymo žingsniai

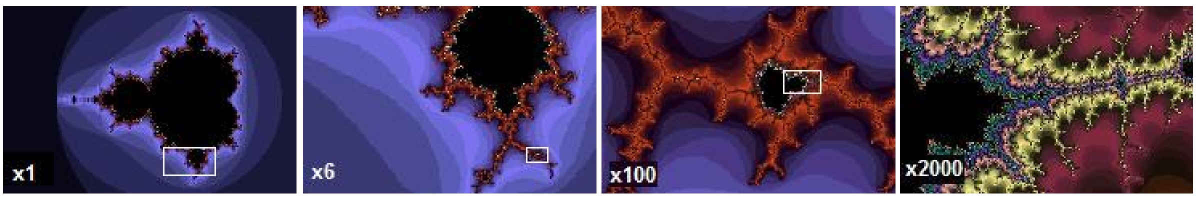
Kocho snaigės ir Sierpinksio nėrinio sudarymas yra rekursinio proceso pavyzdys. Tai toks procesas, kai kas žingsnį vėl ir vėl taikome tas pačias pagrindines taisykles, o kiekvieno žingsnio pabaiga tampa kito žingsnio pradžia. Čia aprašyti geometriniai rekursiniai procesai panašūs į kitus rekursinius procesus, kurių objektai yra skaičiai. Pavyzdžiui, Fibonačio seka sudaroma taip pat rekursinio proceso būdu: 1) turime $F_1=1$ ir $F_2=1$; 2) $F_n=F_{N-1}+F_{N-2}$ visiems $N>2$.

Pastebime, kad ir Kocho kreivė, ir Sierpinskio nėrinys pasižymi tokia savybe: artinant tam tikros jų dalies vaizdą, jis atrodo lygiai taip pat, kaip ir pradinis objektas. Kad ir kiek daug vaizdą artintume, niekas nepasikeis. Ši savybė yra vadinama mastelio simetrija. Tai skirtingų mastelių simetrija, simetrija tarp didelio ir mažo, tarp mažo ir dar mažesnio.



4 pav. Kocho kreivės mastelio simetrija

Mastelio simetrija gali ir nebūti visiškai tiksli. Mandelbroto aibė (ją išsamiau nagrinėsime kitame skyrelyje), pavadinta B. Mandelbroto[3], pirmojo, ėmusio nagrinėti šią aibę ir mastelio simetriją, garbei, turi mastelio simetriją, tačiau didinamos aibės vaizdo dalys yra labai panašios į pradinį vaizdą, bet ne visiškai identiškos (5 pav.).



5 pav. Net ir 2000 kartų padidinus Mandelbroto aibę, gaunami vaizdai, panašūs į pradinį [10]

3 Benoît B. Mandelbrot (g. 1924) – prancūzų-amerikiečių matematikas, fraktalų geometrijos pradininkas; Jeilio universiteto matematikos mokslų Sterlingo profesorius, emeritas; IBM Thomas J. Watson tyrimų centro mokslinis bendradarbis; Šiaurės vakarų Ramiojo vandenyno nacionalinės laboratorijos mokslinis bendradarbis.

Terminą „fraktalas“ (iš lotyniško žodžio *fractus – sudužęs, suskilęs*) 1975 metais pasiūlė B. Mandelbrotas, norėdamas viena sąvoka aprašyti tokius skirtingus objektus, kaip Kocho kreivę, Sierpinskio nėrinį ir Mandelbroto aibę. Šie objektai yra fraktalai ir turi keletą bendrų savybių. Viena iš jų – ką tik aprašyta mastelio simetrija.

Norėdami fraktalus apibrėžti matematiškai (taip, kaip juos apibrėžė B. Mandelbrotas), turime susipažinti su dviem sąvokomis: Hausdorfo dimensija ir topologine dimensija.

Tegu Θ – aprėžtasis aibės $\mathbb{R}^n$ poaibis, o $N_\Theta(\epsilon)$ – mažiausias atvirųjų aibių, kurių skersmuo ϵ, skaičius, reikalingas visiškai užpildyti poaibį Θ. Tada galima apibrėžti poaibio Θ Hausdorfo dimensiją d_H:

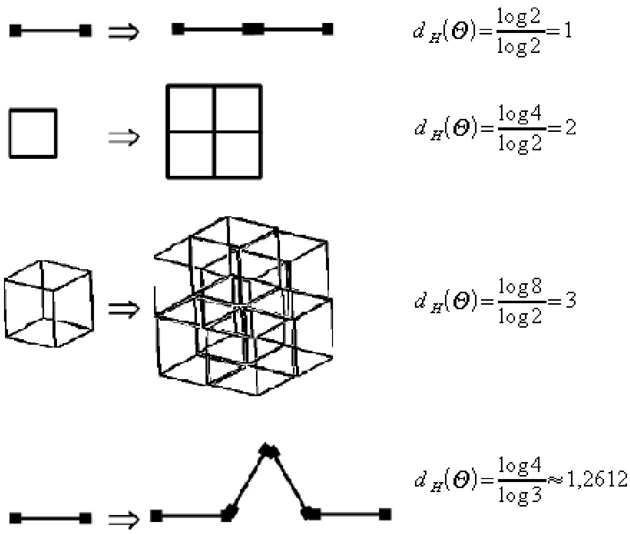
$$d_H(\Theta) = -\lim_{\epsilon \to 0} \frac{\log N_\Theta(\epsilon)}{\log \epsilon}.$$

Hausdorfo dimensiją nesunku apskaičiuoti paprastiems fraktalams. Pavyzdžiui, sudarydami Kocho kreivę vieną atkarpą pakeičiame keturiomis, kurių kiekvienos ilgis yra lygus 1/3 pradinės atkarpos ilgio, taigi šiuo atveju $N_\Theta(\epsilon)=4$, $\epsilon=3$. Tada Kocho kreivės Hausdorfo dimensija bus lygi $d_H(\Theta)=\frac{\log 4}{\log 3}\approx 1{,}26$.

Sierpinskio nėrinio atveju kiekviename žingsnyje imame atskirus trikampius, kuriais uždengiamas pradinis trikampis. *n*-tajame žingsnyje turėsime 3^n trikampių su spinduliu $\frac{1}{2^n}$. Taigi Sierpinskio nėrinio Hausdorfo dimensija neviršys $-\frac{n\log 3}{n\log 1/2}=\frac{\log 3}{\log 2}$ ir galima sakyti, kad ji lygi $\frac{\log 3}{\log 2}\approx 1{,}58$.

6 paveiksle palyginama paprasčiausių euklidinės geometrijos objektų ir Kocho kreivės Hausdorfo dimensija. Matome, kad euklidinės geometrijos objektų Hausdorfo dimensija yra sveikasis skaičius, o Kocho kreivės, kaip, beje, ir kitų fraktalų – trupmeninis. Tai siejasi ir su pačiu terminu „fraktalas“ (angl. *fraction – trupmena*).

Topologinė dimensija – tai sveikasis skaičius, charakterizuojantis tam tikras aibes. Kreivių topologinė dimensija yra 1, paviršių – 2, kūnų – 3.

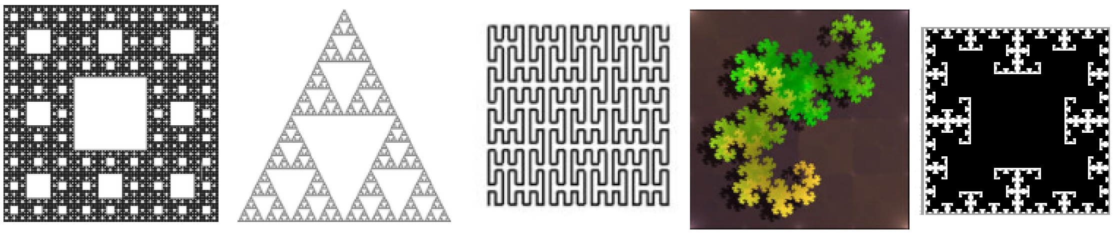


6 pav. Euklidinės geometrijos objektų ir Kocho kreivės Hausdorfo dimensija

Dabar galime pateikti B. Mandelbroto fraktalų apibrėžimą: fraktalas – tai aibės $\mathbb{R}^n$ poaibis, kurio Hausdorfo dimensija yra didesnė už topologinę dimensiją. Vėliau Hausdorfo dimensiją B. Mandelbrotas pavadino fraktaline dimensija.

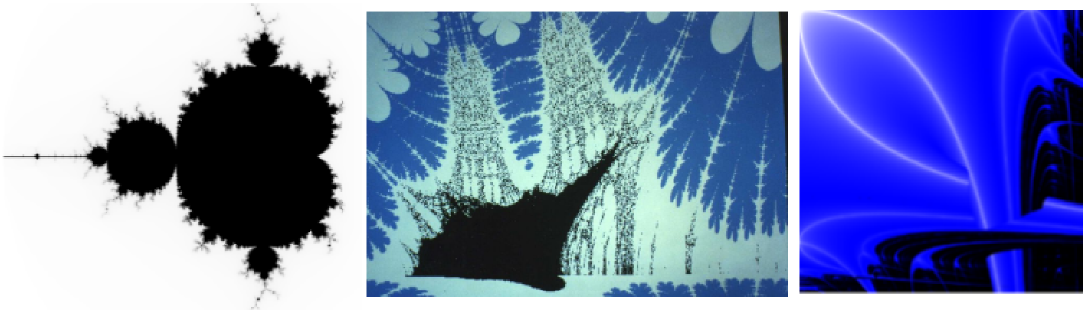
Pagal tai, kokiu būdu fraktalai generuojami, jie skirstomi į tris grupes:

- Iteruotųjų funkcijų sistemos. Jomis aprašoma pastovi geometrinė keitimo taisyklė. Šiuo būdu sudaromi tokie fraktalai, kaip Kantoro aibė, Sierpinskio kilimėlis, Sierpinskio nėrinys, Peano kreivė, Kocho snaigė, Harter-Heighway drakono kreivė, T kvadratas, Mengerio kempinė ir pan. (7 pav.)



7 pav. Sierpinskio kilimėlis, Sierpinskio nėrinys, Peano kreivė, drakono kreivė, T kvadratas [10]

- Neaprėžtu sekos didėjimu aprašomi fraktalai. Šiuo būdu generuojami fraktalai aprašomi rekurentiniais sąryšiais kiekviename erdvės (pvz., kompleksinių skaičių aibės) taške. Tokių fraktalų pavyzdžiai: Mandelbroto aibė, degančio laivo fraktalas ir Liapunovo fraktalas (8 pav.).



8 pav. Mandelbroto aibė, degančio laivo fraktalas, Liapunovo fraktalas [10]

- Atsitiktiniai fraktalai. Tai fraktalai, generuojami ne determinuotaisiais, o atsitiktiniais procesais. Tokių fraktalų pavyzdžiai yra fraktaliniai kraštovaizdžiai, Brauno[4] medis (9 pav.).

9 pav. Brauno medis [10]

Fraktalai gali būti klasifikuojami ir pagal mastelio simetrijos tipą. Yra trys fraktalų mastelio simetrijos tipai:

- Tikslioji mastelio simetrija. Fraktalo, turinčio tokią mastelio simetriją, vaizdas atrodo identiškas priartinus jį neribotą skaičių kartų. Tikslią mastelio simetriją dažniausiai turi iteruotųjų funkcijų sistemomis generuojami fraktalai.
- Apytikslė mastelio simetrija. Šios mastelio simetrijos fraktalų vaizdai, juos artinant, atrodo labai panašūs į pradinį vaizdą, tačiau nėra tikslios jo kopijos. Artinant fraktalo vaizdą, matomos deformuotos viso fraktalo kopijos. Rekurentiniais sąryšiais aprašomi fraktalai paprastai turi apytikslę mastelio simetriją.
- Statistinė mastelio simetrija. Tai silpniausias mastelio simetrijos tipas. Tokiuose fraktaluose, artinant vaizdą, tik išlaikomi esami skaitiniai ar statistiniai įverčiai. Statistinę mastelio simetriją turi atsitiktiniai fraktalai.

Ne visi mastelio simetriją turintys objektai yra fraktalai. Pavyzdžiui, realiųjų skaičių tiesė turi tikslią mastelio simetriją, tačiau jos Hausdorfo ir topologinė dimensijos abi yra lygios vienetui, todėl realiųjų skaičių tiesė nėra fraktalas.

---

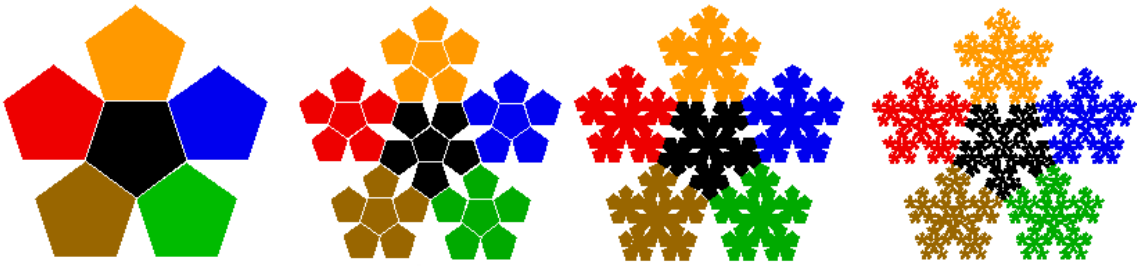
4 Robert Brown (1773–1858) – britų botanikas.

1.2. Fraktalų istorija

Objektai, kuriuos dabar vadiname fraktalais, buvo tyrinėjami daug anksčiau, nei įvestas šis terminas.

Jau graikų geometrai nagrinėjo kūgių pjūvius. Tai buvo 1000 metų prieš tai, kai N. Kopernikas[5], J. Kepleris[6] ir I. Niutonas[7] paneigė susidariusią nuomonę, kad visi dangaus kūnai juda apskritimais, ir atrado elipses, paraboles ir hiperboles (o kūgių pjūviai ir yra elipsės, parabolės arba hiperbolės). XVII amžiuje I. Niutonas ir G. Leibnicas[8] pradėjo skaičiuoti funkcijų diferencialus ir išvestines. Tuomet buvo pastebėta, kad kai kurios iš gautų funkcijų gerai tinka tikrovės modeliavimui. Tačiau apie 1870 metus matematikoje įvyko perversmas: buvo atrasti geometriniai dariniai, kurių negalima buvo pavadinti nei vienmačiais, nei dvimačiais, nei trimačiais objektais. Dauguma į juos žiūrėjo kaip į patologiją. Tyrinėtojai tuo metu stengėsi suprasti svyravimus: Nilo potvynius, kainų kaitą ekonomikoje, molekulių sukimąsi judančiuose skysčiuose, kur tradicinių modelių duomenys netiko. Daugelį metų ši matematikos raida atrodė nesusijusi su fraktalais, tačiau tokie objektai, kaip ir kai kurios matematinės kreivės, chaotiški orbitų judėjimai, netvarkingi laiko tarpų grafikai, turi mastelio simetriją: padidinus mažą fragmentą, jis atrodo labai panašus į pradinį viso objekto vaizdą. Šie faktai įnešė indėlį į fraktalų teoriją, jie yra fraktalų teorijos šaknys.

Dar 1525 metais vokiečių menininkas Albrechtas Dureris[9] parašė „Dailininko vadovą“, kurio viename skyriuje buvo nagrinėjami raštai, sudaryti iš penkiakampių. Durerio penkiakampis (10 pav.) labai panašus į Sierpinskio kilimėlį, tik sudarytas ne iš kvadratų, o iš penkiakampių.



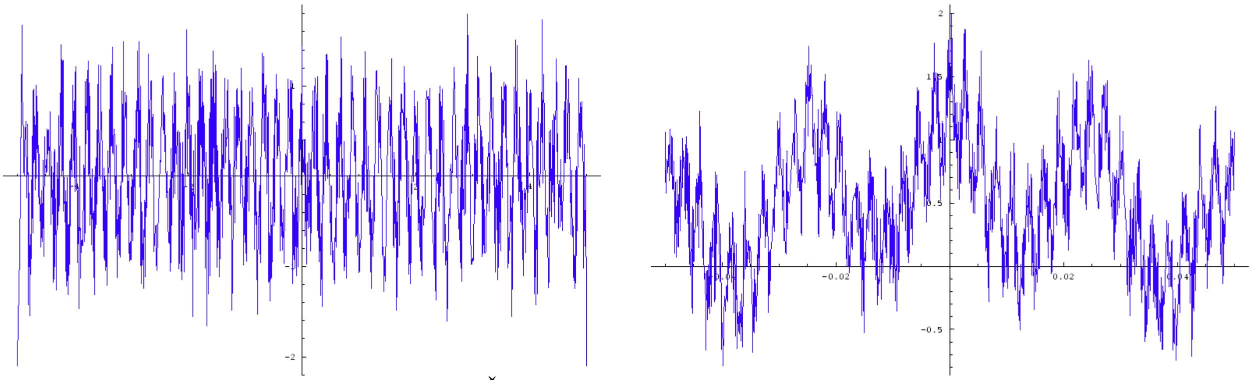
10 pav. Keturi pirmieji Durerio penkiakampio iteracijos žingsniai

Pirmą kartą rekursinį panašumą nagrinėjo ir aprašė filosofas G. Leibnicas. 1872 metais Karlas Vejerštrasas[10] atrado funkciją, kuri yra tolydi visoje skaičių aibėje, tačiau nediferencijuojama jokiame taške:

5 Nicolaus Copernicus (1473–1543) – astronomas, pirmasis pateikęs heliocentrinį saulės sistemos aiškinimą.
6 Johannes Kepler (1571–1630) – vokiečių matematikas, astronomas, labiausiai žinomas dėl planetų judėjimo dėsnių išvedimo.
7 Isaac Newton (1643–1727) – anglų matematikas, fizikas, astronomas, išradėjas ir filosofas, laikomas vienu iš labiausiai nusipelniusių mokslininkų istorijoje.
8 Gottfried Wilhelm Leibniz (1646–1716) – vokiečių filosofas, eruditas, daug nusipelnęs ir filosofijos, ir matematikos istorijoje.
9 Albrecht Dürer (1471–1528) – vengrų kilmės vokiečių dailininkas, medžio drožėjas, graveris, matematikas, pirmasis Europoje mene pavaizdavęs magiškąjį kvadratą.
10 Karl Theodor Wilhelm Weierstrass (1815–1897) – vokiečių matematikas, laikomas šiuolaikinės analizės pradininku.

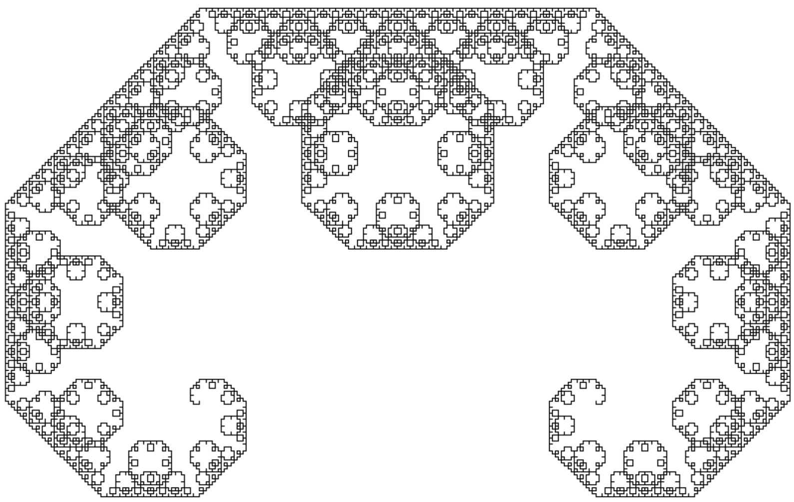
$$f(x)=\sum_{n=0}^{\infty} a^n \cos(b^n \pi x),\ \text{čia}\ 0<a<1\ \text{ir}\ ab>1+\frac{3}{2}\pi .$$

Šios funkcijos grafikas dabar vadinamas fraktalu, nes, kaip ir fraktalai, yra be galo sudėtinis, t. y. turi mastelio simetriją.



11 pav. Vejerštraso funkcija, kai a=9, b=0,7. Šimtą kartų priartinus grafiko vaizdą, matoma aiški mastelio simetrija

1904 metais Helge von Kochas, norėdamas patikslinti ir sukonkretinti Vejerštraso rezultatą, pateikė labiau geometrinį panašios funkcijos aprašą. Šios funkcijos grafikas vadinamas Kocho snaige, ją nagrinėjome 1.1 skyriuje. Tai viena iš pirmųjų aprašytų fraktalinių kreivių. Kreives, turinčias mastelio simetriją, toliau nagrinėjo Paul Pierre Levy[11], 1938 metais aprašęs naują fraktalinę kreivę – Levy C kreivę.

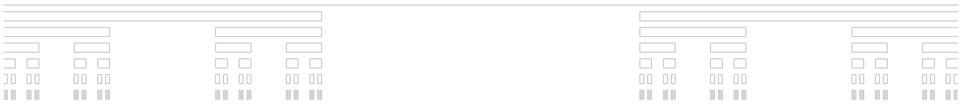


12 pav. Levy C kreivė

Džordžas Kantoras[12] aprašė realiųjų skaičių aibės poaibius, turinčius neįprastų savybių: tokius poaibius sudarė tik realieji skaičiai tarp 0 ir 1. Kantoro aibė sudaroma pakartotinai dalijant atkarpas į tris lygias dalis ir išmetant viduriniąją. Pradedama pašalinant vidurinįjį trečdalį iš vienetinio intervalo $[0;1]$. Po pirmojo žingsnio lieka aibė, sudaryta iš intervalų

11 Paul Pierre Lévy (1886–1971) – prancūzų matematikas
12 Georg Ferdinand Ludwig Philipp Cantor (1845–1918) – vokiečių matematikas, labiausiai žinomas kaip aibių teorijos kūrėjas.

$\left[0;\frac{1}{3}\right]\cup\left[\frac{2}{3};1\right]$. Kiekvienas iš šių intervalų taip pat dalijamas į tris dalis ir išmetamos viduriniosios. Šis procesas kartojamas be galo. Kantoro aibę sudaro tie intervalo $[0;1]$ taškai, kurie nepašalinami šio begalinio proceso metu. Šeši pirmieji Kantoro aibės sudarymo žingsniai pavaizduoti grafiškai:



13 pav. Kantoro aibės sudarymas

Tokia Kantoro aibė taip pat laikoma fraktalu. Kompleksinių skaičių aibėje iteruotas funkcijas XIX amžiaus pradžioje – XX amžiaus pabaigoje tyrinėjo Henri Poincare, Felix Klein, Pierre Fatou, Gaston Julia. Deja, tuomet, be kompiuterinės grafikos galimybių, nebuvo priemonių vaizdžiai atskleisti visą jų išnagrinėtų objektų grožį.

7-ajame XX amžiaus dešimtmetyje B. Mandelbrotas savo darbuose[13] ėmė nagrinėti mastelio simetriją. 1975 metais B. Mandelbrotas pasiūlė teminą „fraktalas“, pateikė įspūdingus kompiuteriu sukurtus fraktalų vaizdus.

Šiandieninės fraktalų geometrijos pradžia ir yra siejama su B. Mandelbroto darbais. 1977 metais išleistoje knygoje „Gamtos fraktalų geometrija“ mokslininkas rašė: „Kodėl (elementarioji) geometrija dažnai vadinama šalta ir sausa? Viena iš priežasčių – tai negalėjimas atvaizduoti debesų, kalnų, krantų ar medžio formų. Debesys – ne rutuliai, kalnai – ne kūgiai, krantai – ne apskritimai, ir nei medžio žievė glodi, nei žaibas tiesėm sklinda. <...> Daugybė gamtos formų yra tokios netaisyklingos ir padrikos, kad palyginus su elementariąja geometrija, gamta demonstruoja ne tik aukštesnį, bet ir visiškai kitokį sudėtingumo laipsnį.“[6]

Dabar fraktalų geometrija taikoma daugybėje sričių: interneto technologijose, signalų ir paveikslų glaudinime, kompiuterinėje grafikoje, medžiagotyroje, populiacijų biologijoje, žmogaus fiziologijoje, įvairiose meno šakose ir net psichologijoje.

1.3. Mandelbroto ir Julijaus aibės

Mandelbroto ir Julijaus aibės yra ko gero labiausiai žinomi ir gražiausi fraktalai. Julijaus aibės (kartais dar vadinamos Julijaus-Fatu aibėmis) taip pavadintos prancūzų matematikų Gastono Julijaus[14] ir Pjero Fatu[15] garbei.

13 *How Long Is the Coast of Britain? Statistical Self-Similarity and Fractional Dimension* (*Science*, 1967). Šiame straipsnyje B. Mandelbrotas aprašo kreives, turinčias mastelio simetriją. Aprašomos kreivės – tai fraktalų pavyzdžiai, tačiau šiame straipsnyje šio termino autorius dar nevartoja.

14 Gaston Maurice Julia (1893–1978) – prancūzų matematikas, išvedęs Julijaus aibių formulę. Jo darbus populiarino B. Mandelbrotas.

15 Pierre Joseph Louis Fatou (1878–1929) – prancūzų matematikas, daugiausia nagrinėjęs kompleksinę analizinę dinamiką.

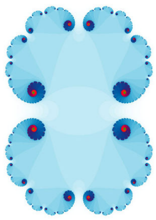
Nepaprasto grožio ir, atrodytų, sudėtingumo geometriniai dariniai aprašomi itin paprastomis kompleksinių skaičių sekomis:

$$z_{n+1} = z_n^2 + c,$$

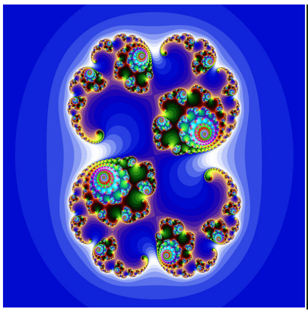
čia c yra kompleksinė konstanta.

Ši seka priklauso nuo konstantos c ir pradinio taško z_0. Julijaus aibės gaunamos fiksuojant konstantą c ir leidžiant z_0 kisti kompleksinių skaičių aibėje. Mandelbroto aibę gauname fiksuodami tašką $z_0 = 0$ ir keisdami parametrą c. Jei tašką z_0 paimsime toli nuo 0, tai seka labai greitai ir neaprėžtai didės. Tai galioja ir tam tikriems n, jei taškas z_n yra toli nuo 0. Tačiau yra tokių z_0, kuriems seka (z_n) yra aprėžta. Tokie taškai, su duotąja konstanta c, sudaro daugianario $f_c(z) = z^2 + c$ užpildytą Julijaus aibę K_c. Tikroji Julijaus aibė sudaryta tik iš kraštinių aibės K_c taškų, tačiau paprastumo dėlei aibę K_c vadinsime Julijaus aibe.

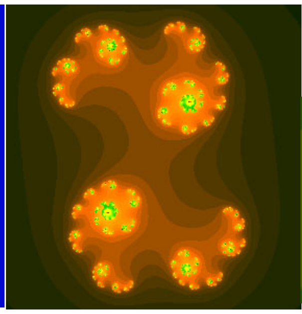
Julijaus aibė ypač priklauso nuo konstantos c, todėl parinkdami skirtingas c reikšmes, gausime daugybę visiškai skirtingų Julijaus aibių (14–17 pav.).



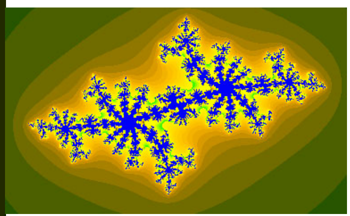
14 pav. Julijaus aibė, kai c=0,285



15 pav. Julijaus aibė, kai c=0,285+0,013i



16 pav. Julijaus aibė, kai c=0,45-0,1428i

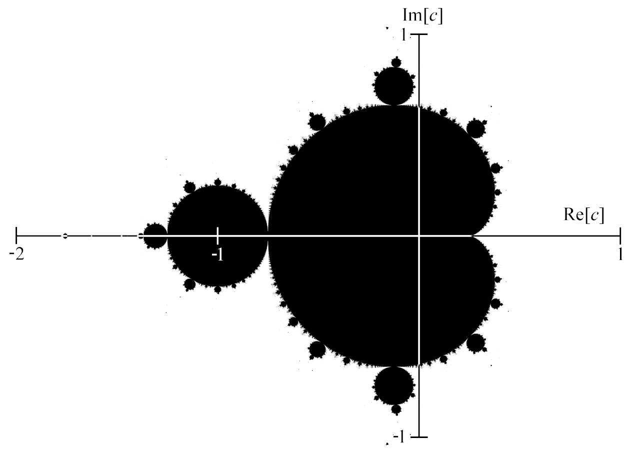


17 pav. Julijaus aibė, kai c=-0,70176-0,3842i

Egzistuoja dviejų tipų Julijaus aibės: vienos iš jų yra vientisos, jos vadinamos jungiosiomis aibėmis (Fatu aibėmis); kitos sudarytos iš daugybės atskirų taškų ir vadinamos Kantoro aibėmis (Fatu dulkėmis).

Dabar galima apibrėžti naują aibę. Aibė konstantos c reikšmių, su kuriomis aibė K_c yra jungi, vadinama Mandelbroto aibe M. B. Mandelbrotas buvo pirmasis aprašęs šią aibę ir kompiuteriu sugeneravęs jos vaizdą.

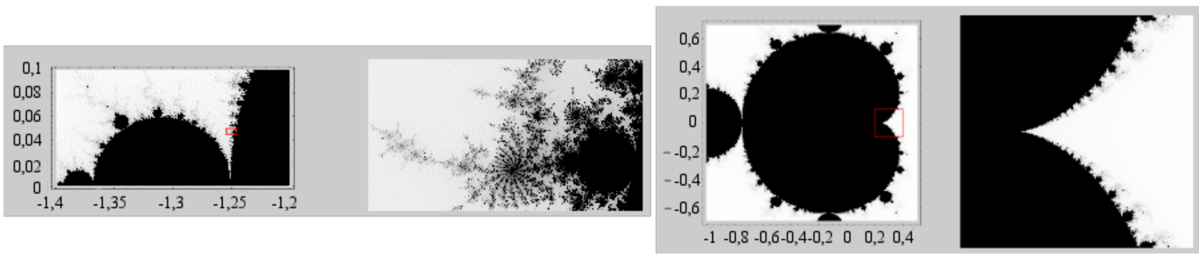
Mandelbroto aibę galima apibrėžti ir kitaip: tai aibė c reikšmių, su kuriomis, pradedant $z_0 = 0$, seka (z_n) yra aprėžta.



18 pav. Mandelbroto aibė

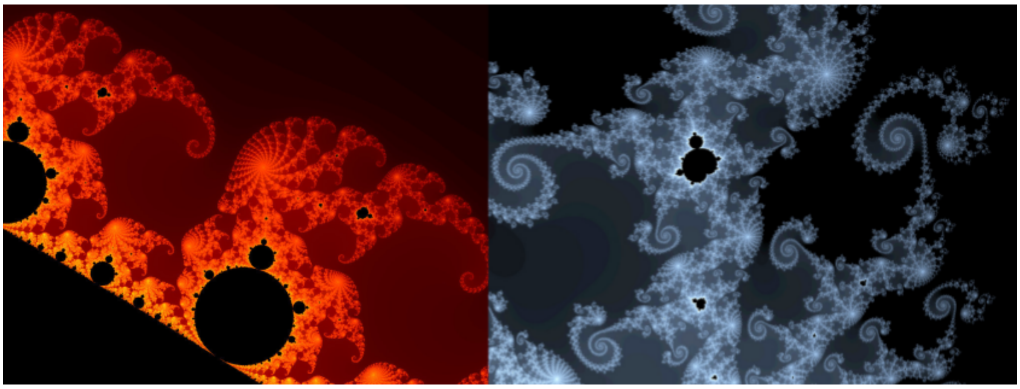
Pažvelgę į Mandelbroto aibės vaizdą (18 pav.), matome kardioidės apribotą sritį. Kardioidės susikirtimo taškas yra taške 0,25, o priešingas kraštas kerta realiąją ašį taške -0,75. Kardioidę liečia skritulys su centru taške -1 ir spinduliu 0,25. Kardioidę taip pat liečia be galo daug mažesnių skritulių, išsidėsčiusių aplink visą kardioidę, šiuos skritulius liečia be galo daug dar mažesnių skritulių ir t. t. Į kairę nuo didžiojo skritulio, ant realiosios ašies yra taškas, vadinamas Myrbergo-Feigenbamo tašku, jo koordinatė yra -1,401... Atkarpa tarp šio taško ir taško -2 yra aibėje M. Ant šios atkarpos yra dar viena kardioidės formos sritis, kurios susikirtimo taškas yra -1,75, ir ją liečia be galo daug skritulių, kaip ir didžiąją kardioidę. Tokių kardioidžių apribotų sričių yra daugiau, ir ne tik ant realiosios ašies. B. Mandelbrotas rado vieną tokią sritį su centru taške 0,1565201668+1,032247109i ir įrodė, kad jų yra be galo daug. Visos šios sritys yra sujungtos su didžiąja kardioide gijomis, ant kurių taip pat yra tokių pat kardioidės formos sričių su jas liečiančiais skrituliais, todėl aibė M yra jungi.

Kai kurios įdomesnės Mandelbroto aibės sritys turi net atskirus pavadinimus. Pavyzdžiui, sritis, kurios centras yra taške -1,25+0,047i, o skersmuo – apie 0,009+0,005i, vadinama jūros arkliukų slėniu, o sritis su centru 0,3 ir skersmeniu 0,1+0,1i – dramblių slėniu (19 pav.).



19 pav. Jūros arkliukų slėnis (kairėje) ir dramblių slėnis

Dauguma Julijaus aibių turi mastelio simetriją. Jei daug kartų padidintume aibės K_c kraštą, išvystume vienodą vaizdą, nepriklausomai nuo to, kiek kartų didiname ir kurią aibės krašto vietą didiname. Tuo tarpu Mandelbroto aibė turi tik apytikslę mastelio simetriją. Aibėje M yra be galo daug mažesnių jos kopijų, tačiau didinant gaunamas vaizdas priklauso nuo to, kiek kartų ir kurią aibės M vietą didiname (20 pav.).

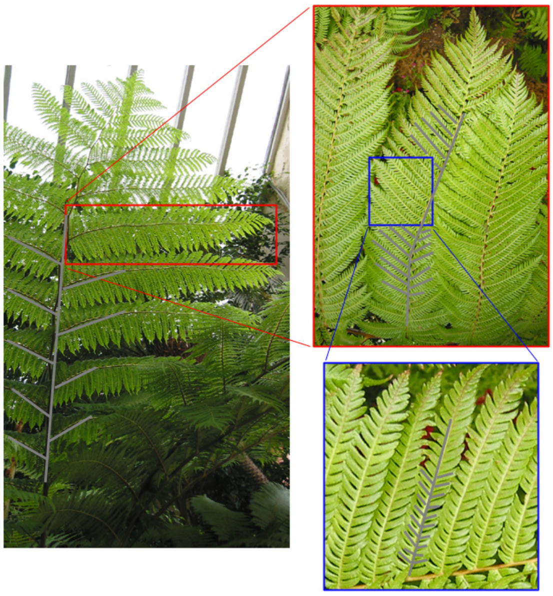


20 pav. Padidintos skirtingos Mandelbroto aibės dalys

1.4. Fraktalai realiame pasaulyje

Gamtoje gausu apytikslių fraktalų. Tokie natūralūs fraktalai taip pat turi mastelio simetriją, tačiau artinimų skaičius yra baigtinis. Debesys, snaigės, kalnai, upių raizginiai, žmogaus kraujagyslių sistema yra puikūs fraktalų gamtoje pavyzdžiai.

Gerai žinomas fraktalas gamtoje – paparčio lapas (21 pav.). Jame akivaizdi mastelio simetrija – kiekviena paparčio lapo šakelė panaši į visą lapą, o taip pat sudaryta iš mažesnių šakelių, panašių į ją. Žinoma, toks mastelio simetrijos žingsnių skaičius paparčio lape yra baigtinis.



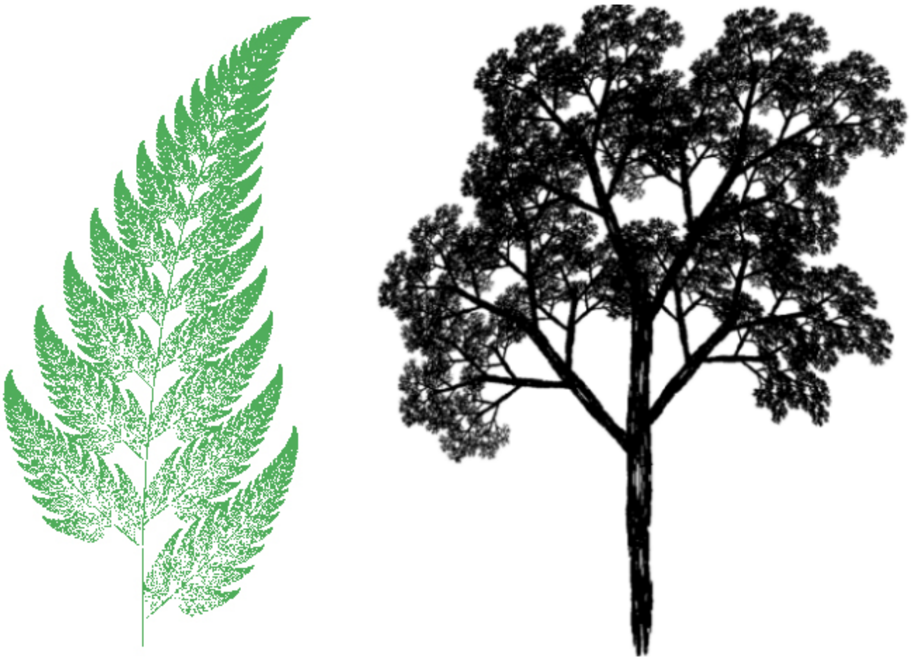
21 pav. Paparčio lapas turi mastelio simetriją

Fraktalo savybių turi net ir 22 paveiksle matomas vaizdas. Tai gali būti uolos paviršius, molio lopinėlis. Pasirinkus tam tikrą tokio vaizdo dalį ir ją priartinus, gautas vaizdas bus panašus į pradinį.

22 pav. Ir šiame vaizde įžvelgiama mastelio simetrija

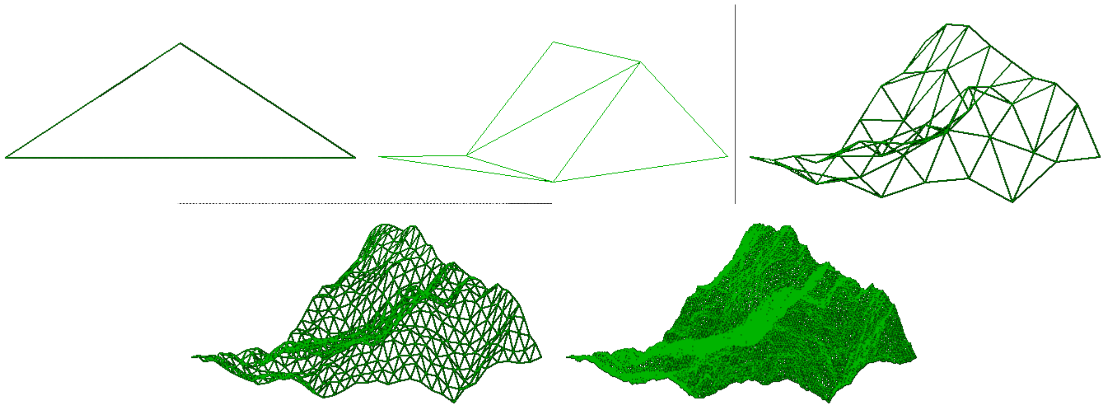
Atsiradus kompiuteriams atsirado galimybė išsamiau nagrinėti fraktalus. Kompiuteriais buvo galima ne tik daug sparčiau generuoti ir nagrinėti įvairius fraktalus, bet ir kurti tikroviškus vaizdus panaudojant fraktalų sudarymo algoritmus.

Medžiai ir paparčių lapai turi fraktalų savybių ir juos pagal rekursinius algoritmus galima modeliuoti kompiuteriu (23 pav.).



23 pav. Iteruotųjų funkcijų sistemomis sumodeliuotas paparčio lapas ir medis

Kalno paviršių taip pat galima modeliuoti kompiuteriu pagal fraktalų generavimo algoritmus. Pavyzdžiui, imkime trikampį trimatėje erdvėje (24 pav.). Atkarpomis sujunkime trikampio kraštinių vidurio taškus. Gauname keturis trikampius. Tada pradinio trikampio kraštinių vidurio taškus atsitiktinai paslenkame nustatytoje srityje. Ta sritis – apskritimas aplink pradinį tašką su tam tikru spinduliu. Antrajame žingsnyje tą patį veiksmą kartojame su keturiais mažesniaisiais trikampiais, tik šįkart apskritimo, kuriame galime slinkti taškus, spindulys bus perpus mažesnis. Tokiu rekursiniu algoritmu sukurto fraktalo bet kuri priartinta dalis bus panaši į pradinį fraktalą.



24 pav. Kalno paviršiaus modeliavimas kompiuteriu

1.5. Fraktalai mokykloje

Fraktalai į mokyklinę programą neįtraukiami, tačiau jie puikiai tiktų užklasinėse pamokėlėse, naudingi būtų ir matematikos pamokose. Ne dėl to, kad fraktalai svarbūs matematikos kurse, tačiau dėl to, kad fraktalai, kaip jokia kita tema, gali ypač sukelti mokinių domėjimąsi matematika. Kokios to priežastys? [4]

- Fraktalus nagrinėti įdomu. Tai kelia ir mokinių, ir mokytojų entuziazmą mokytis ir ieškoti naujų atradimų.
- Fraktalai yra gražūs. Tai sužadina vaizduotę ir skatina tolesnį domėjimąsi dėstomu dalyku.
- Fraktalai suprantami visiems. Skirtingo sudėtingumo fraktalus gali nagrinėti ir jaunesni, ir vyresni mokiniai, studentai.
- Fraktalai žadina norą pažinti. Paprastos, lengvai keičiamos fraktalų sudarymo taisyklės leidžia nagrinėti ir gauti netikėčiausius rezultatus keičiant tam tikrus parametrus.
- Kompiuteriai pagerina mokymąsi. Kompiuteriu sukurti fraktalų vaizdai dar labiau sužadina norą gilinti matematikos žinias.

Susipažinus su fraktalais, galima juos nagrinėti, atvaizduoti kompiuterinėmis priemonėmis ir informacinių technologijų pamokose.

Užduotys mokiniams galėtų būti pagal šio darbo 1.1 skyriuje pateiktus algoritmus sudaryti paprastus fraktalus – Kocho snaigę, Sierpinskio nėrinį, pažaisti chaoso žaidimą (jis paaiškinamas antroje darbo dalyje), vyresniesiems galima pasiūlyti nagrinėti sudėtingesnius fraktalus, atvaizduoti juos kompiuterinėmis priemonėmis, patiems sukurti fraktalą.

2. Fraktalai su Logo

2.1. Logo programavimo kalba

Logo programavimo kalba labiausiai tinka norint perprasti darbo kompiuteriu principus. Šia programa galima atlikti įvairius uždavinius: piešti paveikslus, braižyti geometrines figūras, atlikti matematinius skaičiavimus, modeliuoti įvairius fizikos, chemijos procesus.

Logo programa labai patogu modeliuoti ir projektuoti įvairių sričių uždavinius naudojant šiuolaikinio programavimo idėjas. Šį darbą gali atlikti netgi jaunesniojo amžiaus vaikai. Programa nesudėtinga, greitai perprantama, tačiau turinti daug galimybių įvairiems darbams projektuoti bei programavimo veiksmams užrašyti.

Logo programa leidžia mokiniui kiekvieną žingsnį aprašyti pačiam, taigi pati programa jokių „paslėptų“ procedūrų neatlieka. Tai svarbu nagrinėjant fraktalus ir norint atskleisti tikrąjį jų grožį.

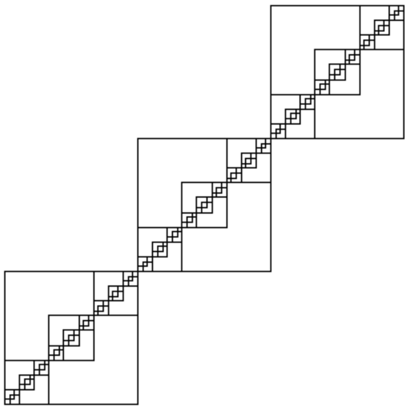
Lietuvoje Logo pradėtas naudoti prieš penkiolika metų. Dabar lokalizuota Komenskio Logo versija naudojama daugelyje Lietuvos mokyklų. Netrukus Lietuvos mokyklas turėtų pasiekti naujausia ir daugiausia galimybių turinti lokalizuota Imagine Logo versija. Šiame darbe pateikiami programų fraktalams kurti pavyzdžiai skirti dabar Lietuvoje populiariai sulietuvintai Komenskio Logo programai.

2.2. Paprasčiausių fraktalų kūrimas su Logo

Pradedant pažintį su fraktalais, galima pademonstruoti fraktalų savybes naudojant paprasčiausias geometrines figūras, susipažinti su fraktalų sudarymo procesu – rekursija. Žinoma, rekursinis procesas nėra lengvai perprantamas, tačiau Logo suteikia priemones, žymiai palengvinančias šį uždavinį.

Šia paprasta procedūra, panaudojant ciklą ir rekursiją, sukuriama figūra, turinti mastelio simetriją, ir sudaryta iš kvadratų:

```
tai kvadratai :kr_ilgis :kv_skaičius
  jeigu :kr_ilgis < 2 [baik]
  kartok 2 [kartok :kv_skaičius [pr :kr_ilgis dš 90 ►
  pr :kr_ilgis kr 90] kr 180]
  kvadratai :kr_ilgis / 3 :kv_skaičius * 3
taškas
```

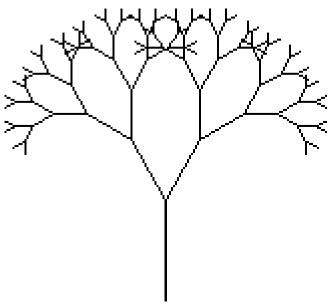


25 pav. Procedūra `kvadratai` gaunama figūra su mastelio simetrija

Susipažinus su rekursija ir pritaikius ją elementarioms figūroms, nesunku pereiti prie tikrų, tegu pradžioje ir paprasčiausių, fraktalų.

Šia Logo procedūra nupiešiamas fraktalinis medis. Procedūros pradinis duomuo – medžio kamieno (pirmojo rekursijos žingsnio) ilgis:

```
tai medis :dydis
  jeigu :dydis < 5 [baik] pr :dydis
  kr 30 medis :dydis * 0,7 dš 60 medis :dydis * 0,7
  kr 30 at :dydis
taškas
```

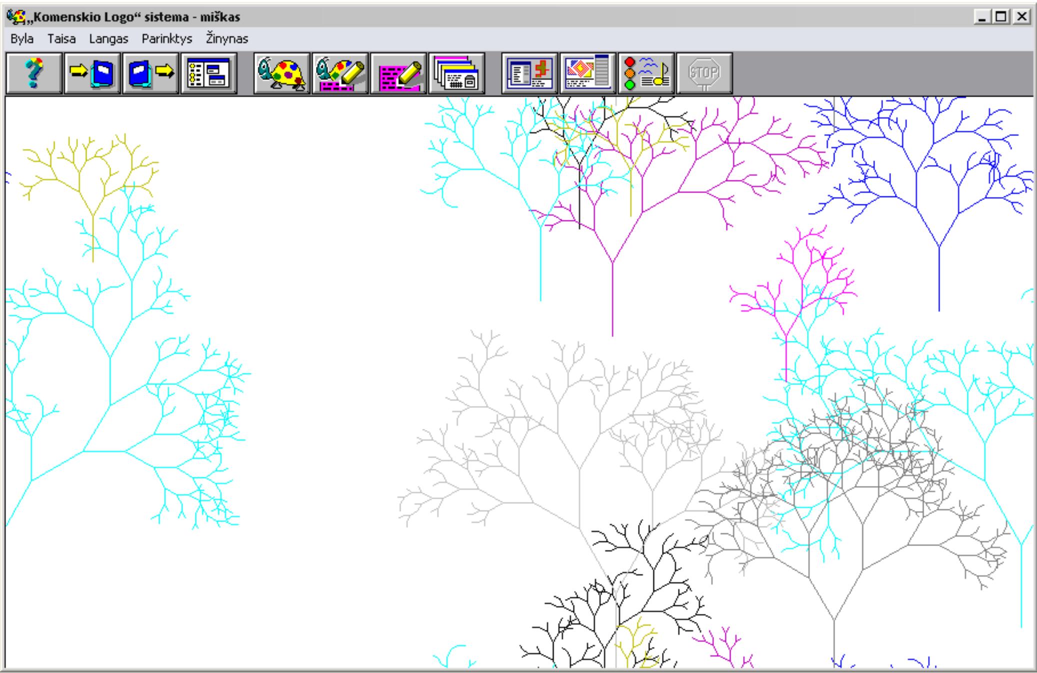


26 pav. Procedūros `medis 50` rezultatas

Šioje procedūroje panaudoję atsitiktinių skaičių parinkimo funkciją `bet.koks` galime sukurti tikroviškesnius, ne tokius simetriškus medžius. Aprašę dar vieną procedūrą, keletą kartų įvykdančią pirmąją, pradedant iš skirtingų vietų ekrane, gausime programą, kuri nupieš daug skirtingų medžių – mišką. Tokią programą galima panaudoti įvairiuose Logo projektuose. Vykdant programą nurodoma, kiek medžių piešti ir didžiausias galimas medžio kamieno ilgis:

```
tai medis2 :dydis
  jeigu :dydis < 5 [baik] pr :dydis
  kr 30 medis2 :dydis * ( ( ( bet.koks 5 ) + 5 ) / 10 )
  dš 60 medis2 :dydis * ( ( ( bet.koks 5 ) + 5 ) / 10 )
  kr 30 at :dydis
taškas
```

```
tai miškas :m :d
  kartok :m [ppsp bet.koks 14 tebus "x bet.koks 600 ►
  tebus "y bet.koks 300 es eik.x.y :x :y pš ►
  medis2 :d * ( ( ( bet.koks 5 ) + 5 ) / 10 )]
taškas
```

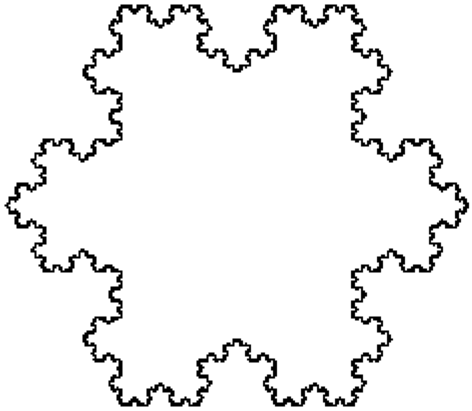


27 pav. Procedūros `miškas 10 80` rezultatas

Pereikime prie pirmoje darbo dalyje minėtų fraktalų. Nesudėtingomis procedūromis galima nubraižyti Kocho snaigę ir Sierpinskio nėrinį. Žinoma, žingsnių skaičius, kurį galima atvaizduoti kompiuterio ekrane, bus ribotas.

Sudarydami Kocho snaigę, procedūrai nurodome pradinio trikampio kraštinės ilgį ir rekursijos žingsnių skaičių:

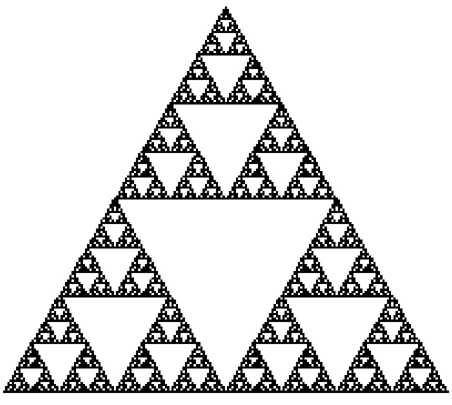
```
tai koch :ilg :žgs
 kartok 3 [kršt :ilg :žgs rt 120]
taškas

tai kršt :ilg :žgs
 jeigu :žgs = 0 [pr :ilg baik]
 kršt :ilg / 3 :žgs - 1
 kr 60 kršt :ilg / 3 :žgs - 1
 dš 120 kršt :ilg / 3 :žgs - 1
 kr 60 kršt :ilg / 3 :žgs - 1
taškas
```



28 pav. Procedūros `koch 200 6` rezultatas

Sierpinskio trikampio programoje pradinių duomenų daugiau – apatinės kairiosios pradinio trikampio viršūnės koordinatės, pradinio trikampio kraštinės ilgis ir rekursijos žingsnių skaičius:

```
tai sierpinski :x :y :ilg :žgs
  jeigu :žgs = 0 [trikampis :x :y :ilg baik]
  sierpinski :x :y :ilg / 2 :žgs - 1
  sierpinski :x + :ilg / 2 :y :ilg / 2 :žgs - 1
  sierpinski :x + :ilg / 4 :y + :ilg * 0.433 :ilg / 2 :žgs-1
taškas

tai trikampis :x :y :ilg
  es eik.x :x eik.y :y žvelk 30 pš
  kartok 3 [pr :ilg dš 120]
  es eik.x :x + :ilg / 2 eik.y :y + 2 pš
  spalvink
taškas
```



29 pav. Procedūros `sierpinski -100 -100 200 6` rezultatas

2.3. Chaoso žaidimas

Imkime bet kokį trikampį *ABC* ir lošimo kauliuką. Kiekvienai trikampio viršūnei priskirkime dvi iš šešių galimų kauliuko metimo baigčių. Tarkime, viršūnei *A* priskiriame skaičius 1 ir 2, viršūnei *B* – skaičius 3 ir 4, o viršūnei *C* – skaičius 5 ir 6. Tokiu būdu kiekviena viršūnė turi vienodą tikimybę būti pasirinkta. Dabar jau galima pradėti žaidimą.

Ridename kauliuką. Pažymime viršūnę, atitinkančią atvirtusį skaičių. Tarkime, iškrito 5. tada pažymime viršūnę *C*. Tai bus mūsų išeities taškas.

1 žingsnis. Vėl ridename kauliuką. Tarkime, atvirto 2 (viršūnė *A*). Pažymime tašką M_1, esantį pusiaukelėje tarp *C* ir *A*.

2 žingsnis. Vėl ridename kauliuką. Pažymime tašką M_2, esantį pusiaukelėje tarp buvusio taško M_1 ir naujai išrinktos viršūnės.

3 žingsnis. Tęsiame šį chaoso žaidimą iki begalybės, kiekvieną kartą pažymėdami pusiaukelės tašką tarp buvusio pusiaukelės taško ir naujai išrinktos viršūnės.

Pamėginkime šį žaidimą realizuoti Logo kalba. Tam tereikia aprašyti dvi paprastas procedūras. Pirmąja procedūra imituojame pirmąjį kauliuko metimą ir pažymime išeities tašką. Ją vykdydami tai pat nurodome, kiek kartų bus metamas kauliukas:

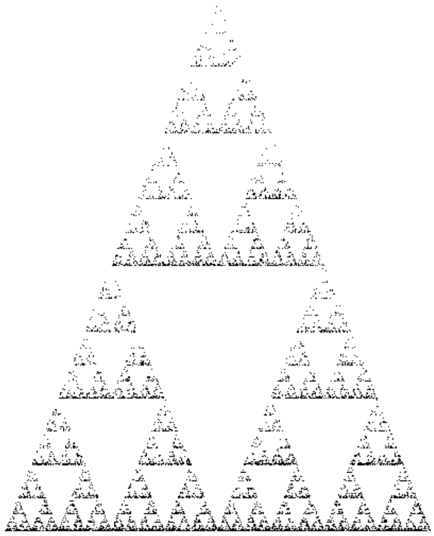
```
tai chaosas :n
  tebus "t bet.koks 6
  jeigu :t = 1 [es eik.į [-200 -200] pš pr 0,1 at 0,1]
  jeigu :t = 2 [es eik.į [-200 -200] pš pr 0,1 at 0,1]
  jeigu :t = 3 [es eik.į [200 -200] pš pr 0,1 at 0,1]
  jeigu :t = 4 [es eik.į [200 -200] pš pr 0,1 at 0,1]
  jeigu :t = 5 [es eik.į [0 300] pš pr 0,1 at 0,1]
  jeigu :t = 6 [es eik.į [0 300] pš pr 0,1 at 0,1]
  kartok :n - 1 [tebus "k bet.koks 6 trik :k]
taškas
```

Antrąja procedūra aprašome vėžliuko veiksmus kiekvienai iš likusių kauliuko metimų baigmių:

```
tai trik :k
  jeigu :k = 1 [tebus "ž ( šaknis ( ( - 200 - koks.x ) * ►
  ( - 200 - koks.x ) + ( - 200 - koks.y ) * ►
  ( - 200 - koks.y ) ) ) / 2 žvelk kiek.link [-200 -200] ►
  es pr :ž pš pr 0,1 at 0,1]
  jeigu :k = 2 [tebus "ž ( šaknis ( ( - 200 - koks.x ) * ►
  ( - 200 - koks.x ) + ( - 200 - koks.y ) * ►
  ( - 200 - koks.y ) ) ) / 2 žvelk kiek.link [-200 -200] ►
  es pr :ž pš pr 0,1 at 0,1]
  jeigu :k = 3 [tebus "ž ( šaknis ( ( 200 - koks.x ) * ►
```

```
( 200 - koks.x ) + ( - 200 - koks.y ) * ►
( - 200 - koks.y ) ) ) / 2 žvelk kiek.link [200 -200] ►
es pr :ž pš pr 0,1 at 0,1]
jeigu :k = 4 [tebus "ž ( šaknis ( ( 200 - koks.x ) * ►
( 200 - koks.x ) + ( - 200 - koks.y ) * ►
( - 200 - koks.y ) ) ) / 2 žvelk kiek.link [200 -200] ►
es pr :ž pš pr 0,1 at 0,1]
jeigu :k = 5 [tebus "ž ( šaknis ( ( 0 - koks.x ) * ►
( 0 - koks.x ) + ( 300 - koks.y ) * ►
( 300 - koks.y ) ) ) / 2 žvelk kiek.link [0 300] ►
es pr :ž pš pr 0,1 at 0,1]
jeigu :k = 6 [tebus "ž ( šaknis ( ( 0 - koks.x ) * ►
( 0 - koks.x ) + ( 300 - koks.y ) * ►
( 300 - koks.y ) ) ) / 2 žvelk kiek.link [0 300] ►
es pr :ž pš pr 0,1 at 0,1]
taškas
```

Chaoso žaidime metus kauliuką pakankamai daug kartų (naudojantis kompiuterio programa nesunku „mesti“ kauliuką kad ir 10000 kartų), pradeda ryškėti iš atidedamų taškų sudaryta figūra – tai Sierpinskio nėrinys.

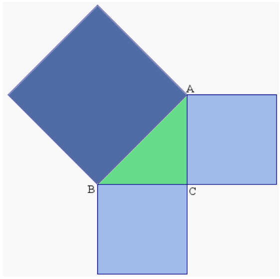


30 pav. Procedūros `chaosas 10000` rezultatas

2.4. Natūralių fraktalų vaizdavimas su Logo

Pitagoro teorema yra tikriausiai labiausiai žinoma senovės graikų teorema. Ypač įdomu, kad ir ja galima pasinaudoti sudarant fraktalus.

Pitagoras savo teoremoje teigė: jei trikampis *ABC* yra status, tuomet ant trikampio statinių sukonstruotų kvadratų plotų suma yra lygi ant trikampio įžambinės sukonstruoto kvadrato plotui (31 pav.).

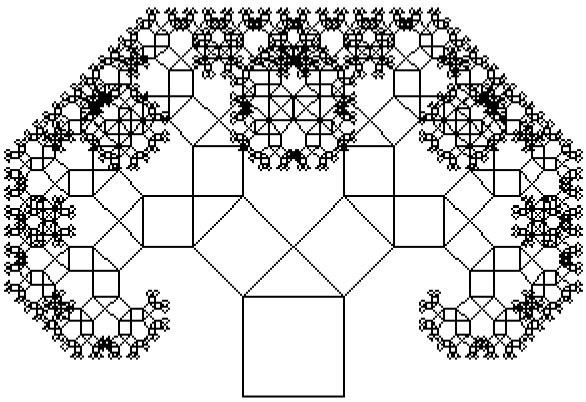


31 pav. Pitagoro teoremos brėžinys

31 paveikslo brėžiniu paremtas fraktalas – Pitagoro medis. Jis, nors ir pavaizduotas elementariomis figūromis, labai panašus į žiedinio kopūsto galvą. Konstrukcija labai panaši į jau nagrinėtus fraktalinius medžius, tik čia atkarpos pakeičiamos kvadratais.

Štai Logo programa Pitagoro medžiui braižyti (vykdant programą nurodomi didžiausio pradinio kvadrato ir mažiausio kvadrato kraštinių ilgiai):

```
tai pitagoras :x :y
  jeigu :x < :y [baik]
  kvadratas :x
  pr :x kr 45
  pitagoras :x / šaknis ( 2 ) :y
  es dš 90 pr :x / šaknis ( 2 ) pš
  pitagoras :x / šaknis ( 2 ) :y
  es at :x / šaknis ( 2 ) kr 45 pš at :x
taškas

tai kvadratas :x
  kartok 4 [pr :x dš 90]
taškas
```


32 pav. Procedūros `pitagoras 50 2` rezultatas

Kita programa realaus pasaulio fraktalui atvaizduoti šiek tiek sudėtingesnė. Prieš ją pateikiant susipažinsime su vienu iš fraktalų sudarymo algoritmų – iteruotųjų funkcijų sistemomis.

Iteruotųjų funkcijų sistemas (IFS) dabartiniu jų pavidalu pirmasis aprašė J. Hutchinson[16] 1981 metais. IFS fraktalai dažniausiai vaizduojami dvimatėje erdvėje, nors juos galima atvaizduoti ir kitoje n-matėje erdvėje. IFS fraktalas – tai rekursinių lygčių sistemos sprendinys. Toks fraktalas sudarytas iš kelių savo paties atvaizdžių sąjungos, kai kiekvienas atvaizdis aprašomas tam tikra funkcija. Jau ne kartą minėtas Sierpinskio nėrinys yra IFS fraktalo pavyzdys. Atvaizduojant fraktalo elementus, jie dažniausiai sumažinami, todėl taškai taip pat mažėja ir perkeliami arčiau vienas kito. Tokiu būdu IFS fraktalas yra sudaromas iš kelių galbūt persidengiančių mažesnių savo paties atvaizdžių, o kiekvieną iš šių atvaizdžių sudaro dar mažesni jo atvaizdžiai ir t. t. Iš čia fraktalai įgyją ir pagrindinę savo savybę – mastelio simetriją.

Iteruotųjų funkcijų sistemą galima užrašyti taip:

$$S=\bigcup_i f_i(S), \text{ čia } S\subseteq\mathbb{R}^n \text{ ir } f_i\colon\mathbb{R}^n\to\mathbb{R}^n.$$

Paprasčiau tariant, IFS yra aibė atvaizdžių, kuriuos taikant geometrinėms figūroms jų seka artėja prie tam tikros ribinės figūros, fraktalų teorijoje vadinamos atraktoriumi (pavyzdžiui, taisyklingųjų n-kampių sekos riba, kai $n\to\infty$, yra apskritimas).

Dažniausiai naudojamas algoritmas IFS fraktalams sudaryti yra chaoso žaidimas, kurio konkretų atvejį Sierpinskio nėriniui sudaryti jau nagrinėjome: parenkamas atsitiktinis pirmasis taškas, tada atsitiktinai parenkama viena iš sistemos funkcijų, ir ją iteruojant randamas kitas fraktalo taškas.

Yra ir kitas būdas IFS fraktalams sudaryti. Sudaromos visos galimos nustatyto ilgio funkcijų sekos ir taškai atidedami taikant kiekvieną iš gautų funkcijų sekų pradiniam taškui arba figūrai.

Užrašykime iteruotųjų funkcijų sistemą natūraliam fraktalui – paparčio lapui – sudaryti.

Pirmiausia atidedamas pradinis taškas koordinačių pradžioje $(x_0=0;\ y_0=0)$. Tada atidedami kiti taškai, atsitiktinai parenkant vieną iš keturių koordinačių transformacijų:

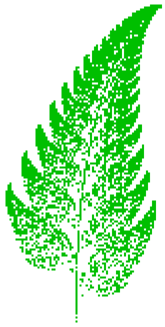
1. $x_{n+1}=0;\ y_{n+1}=0{,}16\,y_n$ – tikimybė, kad bus parinkta ši koordinačių transformacija, yra 0,01.
2. $x_{n+1}=0{,}2\,x_n-0{,}26\,y_n;\ y_{n+1}=0{,}23\,x_n+0{,}22\,y_n+1{,}6$ – ši koordinačių transformacija parenkama su tikimybe 0,07.
3. $x_{n+1}=-0{,}15\,x_n+0{,}28\,y_n;\ y_{n+1}=0{,}26\,x_n+0{,}24\,y_n+0{,}08$ – tikimybė pasirinkti šią transformaciją yra 0,07.
4. $x_{n+1}=0{,}85\,x_n+0{,}04\,y_n;\ y_{n+1}=-0{,}04\,x_n+0{,}85\,y_n+1{,}6$ – tikimybė, kad bus pasirinkta ši transformacija, yra 0,85.

Pirmąja koordinačių transformacija piešiamas stiebas, antrąja – apatinis kairysis lapas, trečiąja – apatinis dešinysis lapas. Ketvirtąja transformacija generuojamos sumažintos stiebo ir dviejų apatinių lapų kopijos – tai kiti paparčio lapai ir jų stiebai.

16 John Irwin Hutchinson (1867–1935) – amerikiečių matematikas.

Dabar galime užrašyti šias IFS taisykles Logo kalba. Tam aprašysime keletą procedūrų: keturias procedūras visoms koordinačių transformacijoms aprašyti, vieną procedūrą transformacijai atsitiktinai parinkti, vieną procedūrą taškui atidėti ir vieną procedūrą, vykdančią programą.

```
tai papartis :n
  ppsp 2 es tebus "x 0 tebus "y 0 kartok :n [parink piešk]
taškas

tai parink
  tebus "r bet.koks 100
  jeigu :r = 0 [transf1 baik]
  jeigu :r < 8 [transf2 baik]
  jeigu :r < 15 [transf3 baik]
  transf4
taškas

tai piešk
  eik.x :x * 160 eik.y :y * 160 - 80
  pš pr 0,1 at 0,1 es
taškas

tai transf1
  tebus "x 0 tebus "y 0,16 * :y
taškas

tai transf2
  tebus "t 0,2 * :x - 0,26 * :y
  tebus "y 0,23 * :x + 0,22 * :y + 0,16
  tebus "x :t
taškas

tai transf3
  tebus "t -0,15 * :x + 0,28 * :y
  tebus "y 0,26 * :x + 0,24 * :y + 0,08
  tebus "x :t
taškas

tai transf4
  tebus "t 0,85 * :x + 0,04 * :y
  tebus "y -0,04 * :x + 0,85 * :y + 0,16
  tebus "x :t
taškas
```



33 pav. Procedūros papartis 10000 rezultatas

2.5. Logo programos Mandelbroto ir Julijaus aibėms sudaryti

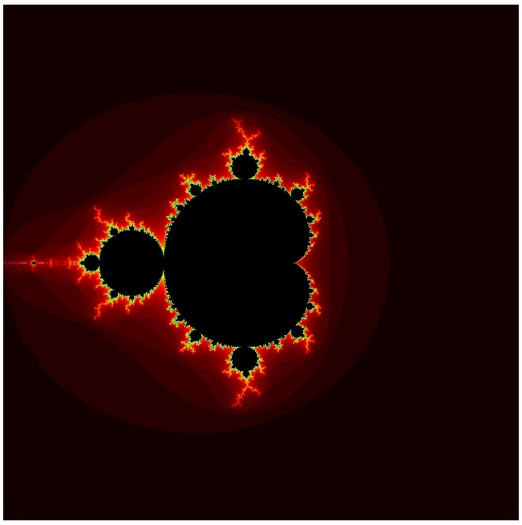
Mandelbroto bei Julijaus aibes nagrinėjome pirmojoje darbo dalyje, tad čia pateikiamos tik Logo programos šioms aibėms sudaryti.

Algoritmas Mandelbroto aibei sudaryti yra gana paprastas. Kompleksinių skaičių aibės sritį, kurioje yra Mandelbroto aibė padalijame į tam tikrą skaičių pikselių. Tarkime, kad *c* yra kiekvieno tokio pikselio vidurio taškas. Keičiant *c* reikšmes sekoje $z_{n+1}=z_n^2+c$, tikrinama, ar pikselio, kurio centras yra taške *c*, skersmuo yra didesnis už 4. Jei taip, pikselis nepriklauso Mandelbroto aibei, ir atskira procedūra, priklausomai nuo to, kaip sparčiai seka didėja tame taške, taškas nuspalvinamas pagal RGB modelį parinkta spalva. Jei pikselio skersmuo nėra didesnis už 4, fiksuotą skaičių kartų pakeitę *c* reikšmę darome prielaidą, kad pikselis yra Mandelbroto aibėje arba labai arti jos. Tokiu atveju pikselis nuspalvinamas juodai.

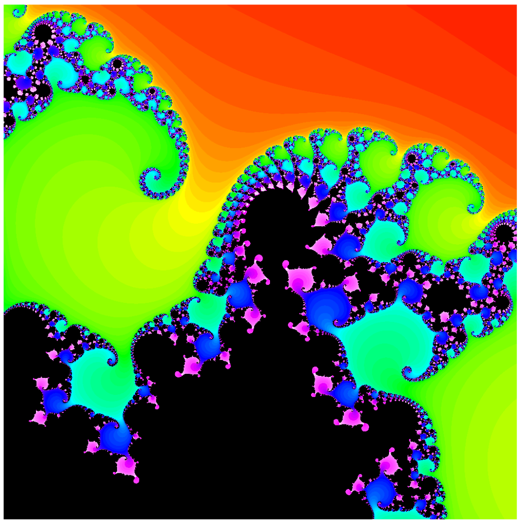
Programos rezultatas – Mandelbroto aibės sritis, apribota kvadratu, kurio apatinio kairiojo kampo koordinates ir kraštinės ilgį nurodome vykdant programą. Pavyzdžiui, norint atvaizduoti visą Mandelbroto aibę, įvedame komandą `mandelbrot -2 -2 4`.

```
tai mandelbrot :xstart :ystart :side
  langas vvl
  slėpkis es
  kartok 1000 [tebus "x :xstart + ( kartojimai / ►
  ( 1000 / :side ) ) tikrink :x :xstart :ystart :side]
  parodyk "baigta
taškas

tai tikrink :x :xstart :ystart :side
  kartok 1000 [iteracija :x :ystart + ( :side / 1000 ) * ►
  kartojimai 0 0 0 :xstart :ystart :side]
taškas
```

```
tai iteracija :cx :cy :zx :zy :it :xstart :ystart :side
  jeigu :it > 100 [juodas :cx :cy :xstart :ystart :side baik]
  tebus "zxn :zx * :zx - :zy * :zy + :cx
  tebus "zyn :cy + 2 * :zx * :zy
  jeigu ( ilgis ( :zxn * :zxn + :zyn * :zyn ) ) > 4 ►
  [spalva :cx :cy :it :xstart :ystart :side baik]
  iteracija :cx :cy :zxn :zyn :it + 1 :xstart :ystart :side
taškas

tai juodas :cx :cy :xstart :ystart :side
  eik.x.y ( :cx - :xstart ) * ( 1000 / :side ) - 500 ►
  ( :cy - :ystart ) * ( 1000 / :side ) - 500
  ppsp 0
  pš
  pr 1
  es
taškas

tai spalva :cx :cy :it :xstart :ystart :side
  eik.x.y ( :cx - :xstart ) * ( 1000 / :side ) - 500 ►
  ( :cy - :ystart ) * ( 1000 / :side ) - 500
  jeigu :it > 0 [ppsp ( gksr ( liekana :it 14 ) * 18 0 0 )]
  jeigu :it > 13 [ppsp ( gksr 255 ( liekana :it 14 ) * ►
  18 0 )]
  jeigu :it > 27 [ppsp ( gksr 255 - ( liekana :it 14 ) * ►
  18 255 0 )]
  jeigu :it > 41 [ppsp ( gksr 0 255 ( liekana :it 14 ) * ►
  18 )]
  jeigu :it > 55 [ppsp ( gksr 0 255 - ( liekana :it 14 ) * ►
  18 255 )]
  jeigu :it > 69 [ppsp ( gksr ( liekana :it 14 ) * 18 ►
  0 255 )]
  jeigu :it > 83 [ppsp ( gksr 255 ( liekana :it 16 ) * 16 ►
  255 )]
  pš
  pr 1
  es
taškas
```



34 pav. Procedūros `mandelbrot -2 -2 4` rezultatas

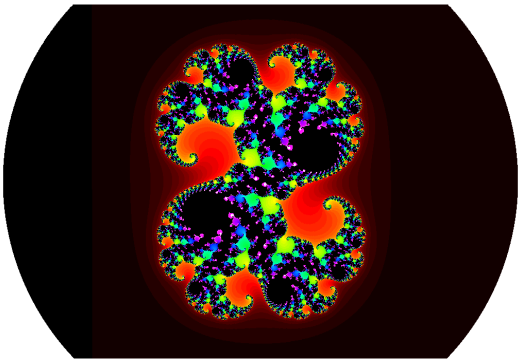


35 pav. Procedūros `mandelbrot 0,28 -0,012 0,004` rezultatas

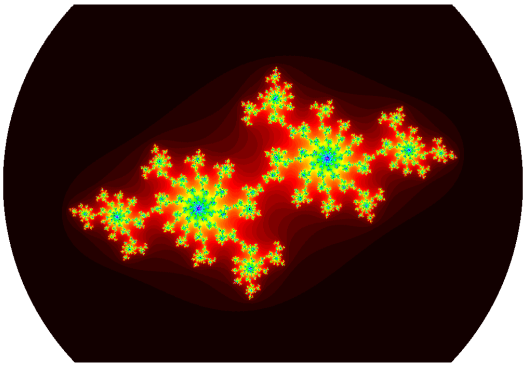
Programa Julijaus aibėms sudaryti labai panaši į pateiktą Mandelbroto aibės sudarymo programą. Vykdydami šią programą, nurodome konstantos *c* reikšmę – realiąją ir menamąją kompleksinio skaičiaus dalis. Taip gaunamas visas Julijaus aibės vaizdas, skirtingas su įvairiomis *c* reikšmėmis.

```
tai julia :cx :cy
  langas
  vvl
  slėpkis
  es
  kartok 1000 [tebus "zx -2 + ( kartojimai / 250 ) ►
  tikrink :cx :cy :zx]
  parodyk "baigta
taškas

tai tikrink :cx :cy :zx
  tebus "zyamp šaknis 4 - :zx * :zx
  kartok apvalink :zyamp * 500 [tebus "zy 0 - :zyamp + ►
  ( kartojimai / 250 ) tebus "zxa :zx ►
  iteracija :cx :cy :zxa 0 - :zyamp + ►
  ( kartojimai / 250 ) 0 :zx :zy]
taškas

tai iteracija :cx :cy :zxa :zya :it :zx :zy
  jeigu :it > 100 [juodas :zx :zy baik]
  tebus "zxn :zxa * :zxa - :zya * :zya + :cx
  tebus "zyn :cy + 2 * :zxa * :zya
```

```
  jeigu ( ilgis ( :zxn * :zxn + :zyn * :zyn ) ) > 4 ►
  [spalva :zx :zy :it baik]
  iteracija :cx :cy :zxn :zyn :it + 1 :zx :zy
taškas

tai juodas :zx :zy
  eik.x.y :zx * 250 :zy * 250
  ppsp 0
  pš
  pr 1
  es
taškas

tai spalva :zx :zy :it
  eik.x.y :zx * 250 :zy * 250
  jeigu :it > 0 [ppsp ( gksr ( liekana :it 14 ) * 18 0 0 )]
  jeigu :it > 13 [ppsp ( gksr 255 ( liekana :it 14 ) * ►
  18 0 )]
  jeigu :it > 27 [ppsp ( gksr 255 - ( liekana :it 14 ) * ►
  18 255 0 )]
  jeigu :it > 41 [ppsp ( gksr 0 255 ( liekana :it 14 ) * ►
  18 )]
  jeigu :it > 55 [ppsp ( gksr 0 255 - ( liekana :it 14 ) * ►
  18 255 )]
  jeigu :it > 69 [ppsp ( gksr ( liekana :it 14 ) * 18 ►
  0 255 )]
  jeigu :it > 83 [ppsp ( gksr 255 ( liekana :it 16 ) * 16 ►
  255 )]
  pš
  pr 1
  es
taškas
```



36 pav. Procedūros `julia 0,285 0,013` rezultatas



37 pav. Procedūros `julia -0,70176 -0,3842` rezultatas

Vykdant abi šias programas, kiekvienam taškui atidėti atliekama daug sudėtingų skaičiavimų, todėl programos veikia labai lėtai. Kompiuteris gali atlikti aritmetines operacijas tik su sveikaisiais skaičiais, todėl kompleksiniai skaičiai aproksimuojami racionaliaisiais, o šie išreiškiami sveikaisiais skaičiais ir operacijomis su jais. Tokia operacijų virtinė ypač ir sulėtina programos darbą.

Išvados

Šiame darbe ištyriau, kaip modeliuojami fraktalai bei pateikiau keletą programų Logo kalba fraktalams sudaryti. Programos antrojoje darbo dalyje pateikiamos nuosekliai, nuo paprasčiausių programėlių supažindinti su rekursija iki sudėtingesnių algoritmų įvairiems fraktalams sudaryti, taip pasiūlant glaustą fraktalų mokymo naudojant Logo metodiką.

Fraktalus reikėtų laikyti vaizdžia matematinių objektų vizualizavimo priemone, naudotina matematikos ir informatikos dalykų užklasinėje veikloje. Pavyzdžiui, iliustruojant rekursijos, kompleksinio kintamojo sąvokas, kompiuterinę grafiką.

Įspūdingas fraktalų grožis ir natūralūs fraktalai gamtoje ypač sudomina mokinius. Nagrinėjant fraktalus susipažįstama su koordinačių sistemomis, sveikųjų skaičių aritmetika, begalybės sąvoka, lavinami skaičiavimo ir braižymo įgūdžiai. Fraktalus gali nagrinėti įvairaus amžiaus mokiniai. Perteikiant fraktalus paprasta ir visiems lengvai perprantama Logo kalba, kartu yra ugdomi algoritmavimo, programavimo ir darbo su kompiuteriu gebėjimai. Sudėtingiems fraktalams sudaryti Logo priemonės nėra racionalios dėl lėto programų darbo atliekant ypač daug skaičiavimų. Žinoma, galima sudaryti fraktalų generavimo algoritmus Logo kalba, tačiau sudėtingiems fraktalams išsamiau nagrinėti labiau tinka tam skirtos programos, galinčios generuoti ypač sudėtingus fraktalus ar jų dalis per žymiai trumpesnį laiką.

Summary

Generation of fractals with Logo

Master's degree thesis

Fractals are all around us, in the shape of a mountain range, cloud formation, flickering fires or our own vascular systems. Such familiar things have become the focus of intense research over the last three decades. Complex mathematical formulas were discovered behind these simple at a first glance structures. It may seem odd that this complexity encourages to go further and further into the fractal theory. Especially when the beauty of fractals is demonstrated using simple tools, such as Logo.

Logo is the name for a philosophy of education and a continually evolving family of programming languages that aid in its realization. Logo programming language uses a turtle, a concrete object that may be seen and handled. In this way geometry is much easier to understand even for the youngest children than the traditional Euclidean geometry, which is constructed on abstractions.

The goal of this master's degree thesis is to analyze properties and generation algorithms of simple fractals and to implement these algorithms into Logo programming language suggesting methodology of teaching fractals using Logo in schools and providing some examples of Logo programs for fractal generation.

The thesis consists of two parts. The first part provides basic theory on fractals. It begins with a simple explanation of what a fractal is using examples of self-similarity and recursive process and going into a more mathematical definition of fractals, introduced by B. Mandelbrot. After a brief history of fractals, a more in-depth analysis of Mandelbrot and Julia sets, the two well-known fractals arising from very simple sequences of complex numbers defined by the relation $z_{n+1}=z_n^2+c$, is given. The last chapter of the first part points out the reasons how fractals are useful and why they should be taught at school – fractals are fun; fractals are beautiful; anyone can play with them; fractals promote curiosity; computers, when used to explain fractal theory, enhance learning.

The second part focuses on using Logo to generate fractals. It provides a few Logo programs of various complexity ranging from simple recursive functions to handling operations with complex numbers. Examples of Logo programs include generation of fractal trees, Koch snowflake and Sierpinski gasket, implementation of chaos game and iterated function systems, and manipulating complex numbers to draw Mandelbrot and Julia sets. The programs are arranged in such a way that they suggest a brief methodology of integrating fractal theory and its Logo implementation into the mathematics curriculum or after-school activities.

Literatūra

1. A. Blaho, I. Kalaš. Komenskio logo: kūrybiškoji informatika. Vilnius, 2002.
2. V. Dagienė. Logo pradžiamokslis. Vilnius, 2001.
3. A. Douady. Julia Sets and the Mandelbrot Set. *The Beauty of Fractals. Images of Complex Dynamical Systems (Ed. H.-O. Peitgen, P. H. Richter)*. Springer-Verlag Berlin, 1986, 161–175.
4. M. L. Frame, B. B. Mandelbrot. Fractals, Graphics & Mathematics Education. Washington D.C. Mathematical Association of America, 2002.
5. J. Harrison. An Introduction to Fractals. *Chaos and Fractals: The Mathematics Behind The Computer Graphics. Proceedings in Applied Mathematics, Volume 39,* Providence, Rhode Island, 1988, 107–126.
6. B. Mandelbrot. The Fractal Geometry of Nature. New York, 1986.
7. P. Tanenbaumas, R. Arnoldas. Kelionės į šiuolaikinę matematiką. Vilnius, 1995.
8. MathWorld, http://mathworld.wolfram.com.
9. PlanetMath, http://planetmath.org.
10. Wikipedia, the free encyclopedia, http://en.wikipedia.org.