Trodimenzionalna fraktalna analiza pojava u kršu

Mikuš, Andreja

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Science / Sveučilište u Zagrebu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:217:373230

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of the Faculty of Science - University of Zagreb

SVEUČILIŠTE U ZAGREBU

PRIRODOSLOVNO–MATEMATIČKI FAKULTET

MATEMATIČKI ODSJEK

Andreja Mikuš

TRODIMENZIONALNA FRAKTALNA ANALIZA POJAVA U KRŠU

Diplomski rad

Voditelj rada:
doc. dr. sc. Dalibor Paar

Zagreb, studeni, 2016.

Ovaj diplomski rad obranjen je dana _____ pred ispitnim povjerenstvom u sastavu:

1. _____, predsjednik
2. _____, član
3. _____, član

Povjerenstvo je rad ocijenilo ocjenom _____.

Potpisi članova povjerenstva:

1. _____
2. _____
3. _____

Hvala mojem mentoru, doc. dr. sc. Daliboru Paaru, na pomoći i uloženom trudu pri izradi ovog diplomskog rada.

Hvala mojoj obitelji, posebno mojim roditeljima i sestrama.
Bez vaše podrške, ljubavi i odricanja tijekom cijelog mog obrazovanja, bilo bi nemoguće sve ovo postići.

Sadržaj

Sadržaj **1**

1 Projektna nastava **4**
1.1 Računalo u nastavi matematike i fizike 4
1.2 Python u nastavi 6
1.3 Projektna nastava 6

2 Fraktalna analiza **10**
2.1 Fraktali 10
2.2 Svojstva fraktala 12
2.3 Podjela fraktala 13
2.4 Cantorov skup 16
2.5 Kochova krivulja i Kochova pahuljica 17
2.6 Trokut Sierpińskog 18
2.7 Mandelbrotov skup 19
2.8 Fraktalna dimenzija 21
2.9 Fraktali i fraktalna dimenzija u nastavi 23

3 Projekt: Trodimenzionalna fraktalna analiza reljefa u kršu **27**
3.1 Instaliranje programske podrške 27
3.2 Program za proračun 3D fraktalne dimenzije 28
3.3 Primjer proračuna 3D fraktalne dimenzije 32

Zaključak **38**

Literatura **39**

Prilog: Program za proračun fraktalne dimenzije trodimenzionalnog skupa podataka **41**

Uvod

Kroz povijest, nastava matematike i fizike prolazila je različite faze svojeg razvoja, od klasične nastave gdje je u fokusu bila prvenstveno pasivnost učenika i predavačka nastava do suvremene nastave gdje učenik postaje aktivan sudionik nastavnog procesa, a nastavnik organizator procesa učenja.

Za razliku od razvijenih zemalja gdje je suvremena nastava pronašla svoje mjesto u nastavnom procesu, kod zemalja u razvoju, kao što je Hrvatska, današnja nastava nedovoljno prati smjernice heurističke i interaktivne istraživački usmjerene nastave. Kako bi se olakšale i ubrzale promjene u nastavnom procesu koje dolaze s napuštanjem tradicionalnih metoda podučavanja, u školama se sve češće provodi projektna nastava.

Da bi projektna nastava polučila što bolji rezultat vezan uz stjecanje novih i trajnih znanja uz odgovarajuću motivaciju učenika, potrebno je osmišljavati problematike koje izlaze iz standardnih primjera kakvi su se do sada obrađivali u okviru nastave. Kao primjer projektnih zadataka i tema mogu se koristiti računalne analize ili simulacije kojima učenik istražuje prirodne pojave. Za očekivati je da će se u budućnosti i u Hrvatskoj povećavati udio ovog oblika nastave (u skandinavskim zemljama projektna i terenska nastava zauzimaju i do 50% nastave) te će biti potreban širok spektar ideja što ponuditi u okviru te nastave.

Cilj ovog rada je prezentirati jednu od takvih ideja koje pokrivaju niz područja (fizika, matematika, programiranje, ali i priroda i društvo te geografija). Radi se o tehnici analize fraktalne geometrije, odnosno određivanju fraktalne dimenzije kojom se utvrđuje fraktalnost određene prirodne pojave (u ovom slučaju špilje kao prirodne pojave u kršu). Fraktalna dimenzija kao rezultat analize prirodne pojave je kvantitativni podatak koji može imati veliku važnost, ne samo radi utvrđivanja fraktalnosti (a time i kvantitativna mjera razine kompleksnosti) dane pojave već ona može biti indikator različitih geoloških, hidroloških i fizikalnih parametara koji su utjecali na razvoj špilja. Metoda se može primijeniti i na druge pojave u kršu poput mreže vodenih tokova, distribuciju ponikvi itd. Nešto od toga je u svijetu publicirano u znanstvenoj literaturi pa uvođenjem učenika u tu tematiku zapravo ih uvodimo u aktualnu znanstvenu metodu, odnosno učimo ih neke od temeljnih koncepata znanstvenog istraživanja: izbor metode, izbor pojave koja se proučava, analiza i interpretacija. U tom kontekstu ovakav projektni zadatak može imati i širi okvir.

Metoda fraktalne analize, odnosno određivanja fraktalne dimenzije u osnovi nije složena. Bazirana je na tehnici brojanja jediničnih kockica (box counting method) te na log-log ovisnosti između broja kockica koje prekrivaju objekt i dimenzija kockica. Ta ovisnost je linearna unutar određenog područja varijabli i u tom području fraktalna dimenzija predstavlja nagib te linearne ovisnosti. Fraktalna dimenzija daje bitnu informaciju o fraktalnosti dane prirodne pojave (odnosno kvantitativno definira njenu kompleksnost).

U diplomskom radu, osim projektne nastave, fraktala i fraktalne dimenzije u nastavi matematike i fizike, pojašnjena je i važnost informatike. U svrhu što boljeg prikaza mogućnosti upotrebe računalnih analiza u proučavanju prirodnih pojava i aktivnog učenja zasnovanog na samostalnom istraživanju učenici mogu koristiti jednostavan i lako dostupan programski jezik Python koji je postao standardni alat, kako u nastavi tako i u znanosti.

Poglavlje 1

Projektna nastava

1.1 Računalo u nastavi matematike i fizike

Kako se sve više odmičemo od tradicionalne nastave, u kojoj je učenik bio tek pasivni slušatelj i promatrač, prema suvremenoj nastavi, u kojoj je učenik aktivni sudionik odgojno – obrazovnog procesa, uočavamo važnost suvremenih pomagala i sredstava koji su uključeni u nastavu matematike i fizike.

Općenito, sama nastava matematike i fizike je učenicima vrlo često kompleksna i nerazumljiva zbog složenosti problema i pojava koja se javljaju. Zbog njihovog lakšeg shvaćanja i proučavanja, a time i obrade dobivenih podataka u nastavi se sve više koristi računalo.



Slika 1.1: Prikaz računala kao poveznice između različitih predmeta

Pojava računala u nastavi matematike i fizike, osim boljeg shvaćanja nastavnog sadržaja, pridonijela je povećanju dinamičnosti u razredu i zainteresiranosti učenika za rad te njihovom lakšem učenju i bržem napretku u savladavanju gradiva. Također, uvođenjem računala u nastavu, učenici bolje i lakše uočavaju poveznicu između određenih predmeta (Slika 1.1).

Takva suvremena nastava u kojoj se koriste računala omogućava učenicima da većim dijelom sami ili u grupi (uloga nastavnika je da usmjerava) prolaze kroz odgojno – obrazovni proces, međusobno diskutiraju o danim projektima i materijalima, sami pripremaju izlaganje, sami odvajaju bitno od nebitnog, pronalaze nove načine kako nešto učiti...

Upotreba računala u nastavi daje značajan doprinos na način da pomaže učenicima u:

- eksperimentiranju, stvaranju hipoteza i provjeravanju njihove valjanosti,
- radu s realističnim podacima,
- vizualizaciji složenih skupova podataka,
- pojednostavljivanju složenih problema i zadataka,
- usporedbi teorijske podloge s eksperimentalnim izračunima,
- razvijanju logičkog mišljenja,
- razvijanju sposobnosti matematičkog modeliranja na temelju danih podataka,
- razvijanju vještina učenja i komunikacije s drugim učenicima i s nastavnikom.

Koristeći se računalom u nastavi matematike i fizike učenici se koncentriraju na osmišljavanje matematičkih ideja kojima bi objasnili pojedine fizikalne probleme i pojave, na jednostavniji i brži način rješavaju probleme koji su stavljeni pred njih na način da istražuju i otkrivaju te proširuju dane probleme.

Različiti računalni programi i simulacije omogućuju učenicima da si vizualno predoče problem koji je postavljen pred njih te da jasnije shvate što se traži od njih kod njegovog rješavanja. Zorni prikaz, bilo matematičkog bilo fizikalnog problema, učenike prvenstveno usmjerava na pravilno razmišljanje o problemu. Također, vizualizacijom apstraktnih problema pomoću računala učenici mogu dobiti više ideja i načina za njegovo rješavanje. U mnogim slučajevima kada školski laboratoriji iz fizike nisu dovoljno opremljeni aparaturom potrebnom za provođenje različitih eksperimenata, računalne simulacije olakšavaju učenicima upoznavanje sa problemom.

Hrvatske škole se sve više opremaju računalima, laptopima, printerima, projektorima, skenerima i ostalim nastavnim sredstvima i pomagalima kako bi se poboljšao standard učenja i poučavanja. Uvođenjem računala u nastavu nastava postaje učenicima zanimljivija nego prije, privlači ih mogućnost samostalnog istraživanja i iznošenja vlastitih otkrića i saznanja svojim kolegama.

1.2 Python u nastavi

Razvojem tehnologije i sve većom uporabom računala u nastavi, nastavnicima i učenicima omogućeno je korištenje velikog broja različitih informatičkih programa i alata u svakodnevnom učenju i poučavanju. Zbog svoje jednostavnosti, Python i njegovi pripadajući alati koriste se kao temelj i početak svakog programiranja u osnovnim i srednjim školama. Ipak, jedna od najvećih prednosti, zašto je ovaj programski jezik toliko rasprostranjen jest ta da je besplatan.

Python je interpreterski, interaktivni i objektno orijentirani programski jezik stvoren početkom 90-tih godina 20. stoljeća. Naredbe koje se koriste su vrlo jednostavne te učenik, bez da zna išta o programu, može izvoditi jednostavne naredbe a time ne mora ovisiti o mogućnosti nastavnika da mu diktira naredbe koje će upisivati. Python omogućava prikupljanje, analiziranje, vizualizaciju i obradu dobivenih podataka bez konstantnog prebacivanja iz programa u program što uvelike olakšava provedbu učeničkih projekata u školi i izvan nje. Osim što se u nastavi počeo sve više koristiti zbog prikaza podataka i mjerenja dobivenih u raznim pokusima, Python je idealan za motivaciju učenika koji brzo gube zanimaciju za matematiku i fiziku zbog teškoće problemskih zadataka na koje nailaze. Omogućava podizanje dinamičnosti u razredu i lakše objašnjavanje matematičkih i fizičkih cjelina koje su teške i nerazumljive učenicima.

Python je idealan jezik za učenje programiranja i rješavanje fizikalnih problema (na način da se u Pythonu rade različite simulacije zbog nemogućnosti izvođenja eksperimenata na satu fizike) kako za učenika tako i za nastavnika.

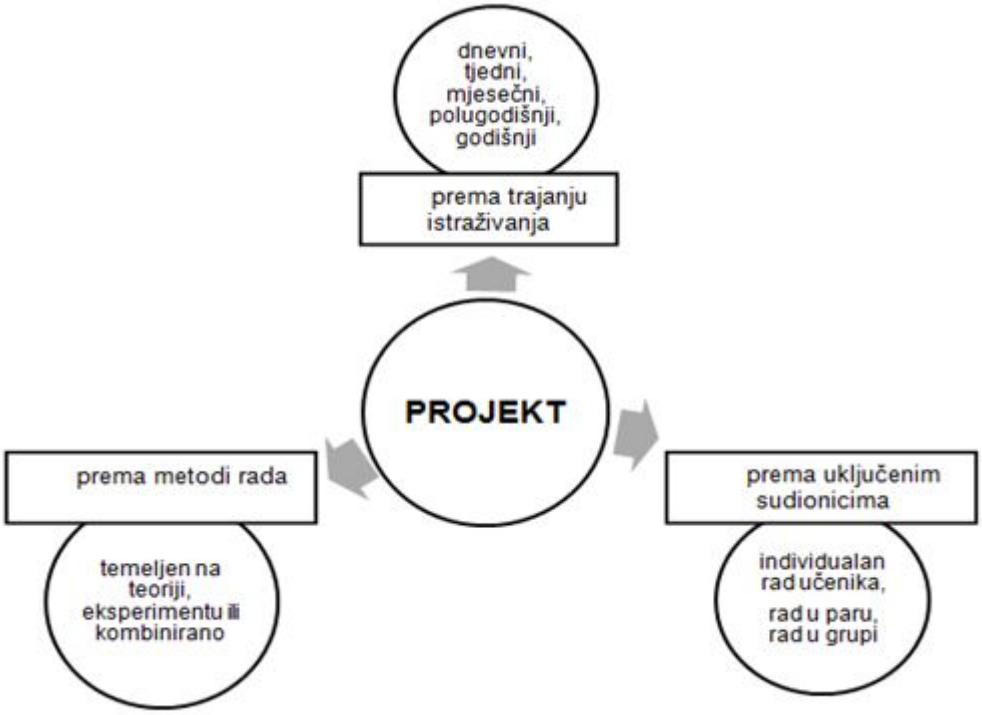
1.3 Projektna nastava

Klasična nastava matematike i fizike najčešće je bila ograničena nastavnim planom i programom i koncentrirana na realizaciju materijalnih zadaća nastave. Zadaci zatvorenog tipa, nedostatak timskog rada učenika kao i projektnih zadataka kod učenika je rezultiralo slabijim povezivanjem usvojenog znanja, vještina i sposobnosti.

Za razliku od klasične nastave, u suvremenu nastavu matematike i fizike sve češće uvodi se metoda suradničko – timskog rada. Kao glavna odrednica takve metode ističe se partnerski odnos učenika i nastavnika. Nastavnici nastoje izvoditi nastavu na način da ona izlazi iz okvira uskih matematičkih i fizičkih sadržaja, a samim time i iz okvira osnovnoškolske i srednjoškolske matematike i fizike. Kao važan oblik takvo suvremenog učenja i poučavanja ističu se projektni zadaci, odnosno projektna nastava.

Postoji nekoliko različitih definicija projektne nastave. Projekt je kompleksan zadatak baziran na izazovnom i interesantnom pitanju ili problemu, koji traži od učenika da postavi istraživanje, istraži, riješi problem kroz duži vremenski period i donese konačni produkt koji se javno prezentira. Projektna nastava je model organiziran oko projekata.

Projektna nastava je oblik nastave koji je sve više prisutan u razredu. Cilj takve nastave je da u prvi plan procesa učenja i poučavanja stavi učenika. Uvođenjem projekata u nastavni plan i program učenike se potiče na aktivno učenje istraživanjem, razmišljanjem i zaključivanjem čime se razvijaju sposobnosti opažanja i opisivanja, bilježenja podataka, logičkog zaključivanja, razlikovanja bitnih od nebitnih činjenica, usmenog i pisanog izražavanja i dr.



Slika 1.2: Podjela projekta prema trajanju istraživanja, uključenim sudionicima i metodi rada

Glavne značajke projektne nastave su te da projekti nisu ograničeni vremenom (mogu biti dnevni, tjedni, mjesečni, polugodišnji, godišnji), brojem sudionika (mogu biti individualni, u paru, u grupi) kao ni metodom rada (mogu biti temeljeni na teoriji, eksperimentu ili kombinirano) (Slika 1.2).

Realizacija projekata podijeljena je u nekoliko faza (etapa):

1. Priprema projekta
2. Upoznavanje s problemom i postavljanje hipoteze
3. Razmišljanje o mogućim načinima rješavanja danog problema
4. Provedba projekta
5. Donošenje zaključaka o projektu
6. Prezentiranje dobivenih rezultata
7. Evaluacija rada

U prvoj fazi, *Priprema projekta*, glavni dio organizacije preuzima nastavnik. On mora detaljnije promisliti o problemu (o vremenskoj i prostornoj organizaciji te o sredstvima koji su učenicima potrebni) i dodatno ga istražiti kako bi projekt koji zadaje svojim učenicima bio što zanimljiviji i motivirajući za njihovo daljnje učenje, mora pripremiti svoje učenike za određenu temu (učenicima je potrebno neko predznanje za provedbu projekta), treba odrediti obrazovna postignuća koja se tim projektom mogu steći,...

U drugoj fazi, *Upoznavanje s problemom i postavljanje hipoteze*, nastavnik učenicima iznosi složen i otvoren problem koji izlazi iz okvira nastavnih sadržaja. Također, taj problem treba biti jer je cilj projekta da učenici nadograde postojeće znanje. U ovom dijelu projekta, biraju se sudionici koji bi radili na projektu (učenici se sami mogu podijeliti i organizirati ili ih grupira nastavnik). Učenici postavljaju hipotezu čiju točnost moraju provjeriti.

Prije nego li krenu na provedbu projekta, u trećoj fazi, učenici moraju promisliti i raspraviti (međusobno, ali i s nastavnikom koji vodi projekt) o mogućim pravcima rješavanja danog problema.

U četvrtoj fazi, *Provedba projekta*, učenici počinju s prikupljanjem informacija i radnog materijala. Važno je napomenuti da zapisuju i dokumentiraju svaki korak istraživanja. Učenicima se treba sugerirati da koriste dodatna pomoćna sredstva kao što su računalo, dodatna literatura,... Istodobno, učenici trebaju komunicirati s nastavnikom i izvještavati ga o provedenim istraživanjima.

U petoj fazi, učenici na temelju dobivenih rezultata donose zaključke o danom problemu. Komentiraju rezultate koje su dobili i uspoređuju teorijsku podlogu s eksperimentom koji su proveli.

Nakon donošenja vlastitih zaključaka o problemu koji su istraživali, učenici moraju dobivene rezultate prezentirati nastavniku i svojim kolegama u razredu. Također, potrebno je generalizirati dobivene podatke.

Posljednju fazu realizacije projekta provode sami učenici (a ne nastavnik) ponovnim promišljanjima o projektu, obrazlažu ispravne i neispravne pretpostavke koje su donijeli tijekom rada na projektu, diskutiraju o projektu,... Od nastavnika učenici dobivaju povratnu informaciju ako treba nešto promijeniti.

Treba naglasiti da nastavnici u projektnoj nastavi imaju ulogu moderatora nastave (ne daju upute već postavljaju otvorena pitanja koja učenicima služe kao smjernice); oni potiču, motiviraju, savjetuju i usmjeravaju učenike te ih uče vrednovanju i kritici vlastitog rada. Važno je također naglasiti da se pred učenike uvijek stavljaju realni ciljevi i zadaci prilagođeni njihovim mogućnostima.

Projektna nastava uvelike se odvija izvan učionice. Učenicima je dopuštena maksimalna sloboda u izražavanju i istraživanju, sudjelovanje u planiranju, programiranju i ostvarivanju rada. Možda i najbitnija stavka projektne nastave je ta da učenici mogu sami vrednovati vlastiti rad. Radom na projektu učenik stvara pozitivne stavove ne samo prema sebi i svojim mogućnostima već i prema svojim kolegama i njihovom načinu razmišljanja.

Uvođenjem projekata u nastavu, odstupa se od klasičnih uskih okvira nastavnog plana. Nastava postaje zahtjevnija, kako za učenika koji treba samostalno istraživati tako i za nastavnika koji se treba dodatno angažirati, ali istodobno zanimljiva i motivirajuća.

Poglavlje 2

Fraktalna analiza

„ Naš osjećaj lijepog potaknut je harmoničnim skladom reda i nereda koji su svojstveni prirodnim predmetima – oblacima, drveću, planinskim lancima ili kristalima snijega. Svi ti oblici su dinamički procesi pretopljeni u fizičke oblike, a posebne kombinacije reda i nereda tipične su za njih.“

James Gleick

2.1 Fraktali

Nalazimo li se na brodu daleko od kopna, ali ne dovoljno daleko da ga ne vidimo, činit će nam se da je obala ravna. No, ako se približimo dovoljno blizu obali, možemo vidjeti da to i nije baš tako: obala ipak nije ravna. Do istog zaključka možemo doći ako obalu gledamo iz ptičje perspektive (Slika 2.1). S druge strane, ako promatramo veoma mali dio obale možemo uočiti da je on sličan samoj obali.



Slika 2.1: Otok Vela Palagruža (gledano iz ptičje perspektive)

Klasično, euklidsko, gledanje na svijet oko nas počelo se mijenjati početkom 20. stoljeća uvođenjem teorije kaosa u sfere znanstvenog života. Tu granu matematike, nastalu još u antičko doba, zamijenili su fraktali – toliko kompleksne, a opet prekrasne funkcije koje sadrže red i sklad.

Naziv fraktal potječe od latinske riječi fractus (razlomljen, slomljen, nepovezan), a osmislio ga je 1978. godine francuski matematičar Benoit Mandelbrot (Slika 2.2) koji se bavio proučavanjem neobičnog i neuhvatljivog oblika u prirodi koji se nije „pokoravao“ oblicima Euklidovog gledanja svijeta preko pravaca, krugova, trokuta i kvadrata.



Slika 2.2: Benoit Mandelbrot

Preciznu definiciju fraktala i fraktalne geometrije teško je dati. Sam Mandelbrot rekao je da su fraktali skupovi čija je fraktalna dimenzija veća od topološke. Možda preciznija definicija glasi da je fraktal objekt koji ima necjelobrojnu fraktalnu dimenziju.

Kako su fraktali usko vezani uz teoriju determinističkog kaosa, učenicima su navedene definicije često nejasne. Zbog toga u srednjoškolskom obrazovanju bi možda bio primjereniji opis da su fraktali neravni ili izlomljeni geometrijski oblici koje je moguće podijeliti na još sitnije dijelove takve da je svaki umanjeni dio približna kopija cjeline, tj. to su geometrijski oblici koji su u različitim mjerilima i dalje slični sami sebi (ako povećamo neki fraktal on će uvijek izgledati isto). Uzastopnim ponavljanjem, iteracijom, nekog računskog ili geometrijskog postupka nastaju fraktalne slike. Njihovim uvećanjem vide se detalji koji nisu bili vidljivi prethodno i tako u beskonačnost.

2.2 Svojstva fraktala

Fraktali kao geometrijski oblici imaju tri važna svojstva: samosličnost, fraktalnu dimenziju i oblikovanje iteracijom.

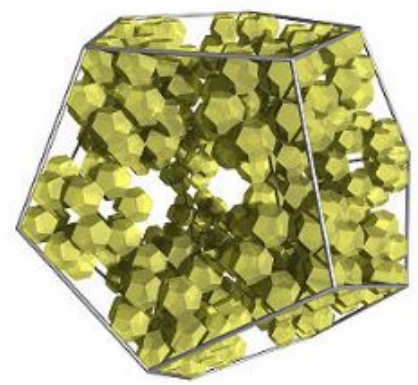
Svojstvo samosličnosti fraktala govori da je jedan njegov dio (bez obzira koji dio fraktala uzeli i promatrali) sličan cijelom objektu, odnosno uvijek ćemo dobiti sliku koja sliči početnoj bez obzira koliko puta uvećali objekt. Kao primjer samosličnosti možemo navesti primjere iz prirode – list i brokulu (Slika 2.3). List, kao ni brokula, nisu klasični matematički primjeri fraktala, ali njihovom podjelom na manje dijelove možemo uočiti sličnost svakog tog dijela sa cjelinom.



Slika 2.3: List i brokula kao primjeri na kojima možemo uočiti svojstvo samosličnosti

Drugo svojstvo, fraktalna dimenzija, je broj koji pokazuje u kojoj mjeri neki fraktal ispunjava prostor u kojem se nalazi. Treba uočiti da je taj broj u većini slučajeva razlomak (jer više nismo u Euklidovoj dimenziji). Glavna karakteristika fraktalne dimenzije je to da ona ostaje konstantna bez obzira na mjerilo koje koristimo.

Kao posljednje svojstvo fraktala navodi se oblikovanje iteracijom. Sam postupak iteracije, odnosno uzastopnog ponavljanja nekog računskog ili geometrijskog postupka, nije težak za shvatiti ali može biti težak za provesti. Kao primjer tog svojstva možemo uzeti dodekaedarski fraktal (Slika 2.4). To je primjer matematičkog fraktala gdje se dodekaedar zamijeni s dvadeset manjih dodekaedara u svakom vrhu. Svaki od tih manjih dodekaedara može se zamijeniti s još dvadeset dodekaedara u svakom kutu i tako u beskonačnost.



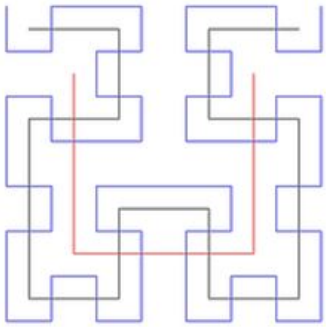
Slika 2.4: Dodekaedarski fraktal kao primjer svojstva oblikovanja iteracijom

2.3 Podjela fraktala

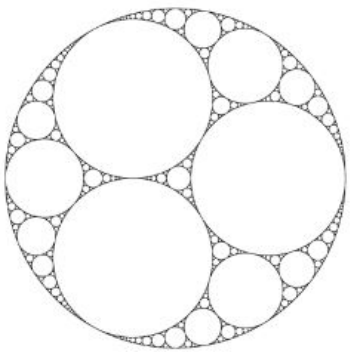
Postoje razni načini podjele fraktala. U ovom diplomskom radu dotaknut ću se dvije klasifikacije fraktala i to: po stupnju samosličnosti i prema načinu nastanka. Po stupnju samosličnosti fraktale razlikujemo kao:

- potpuno samoslične,
- kvazi samoslični,
- statistički samoslični.

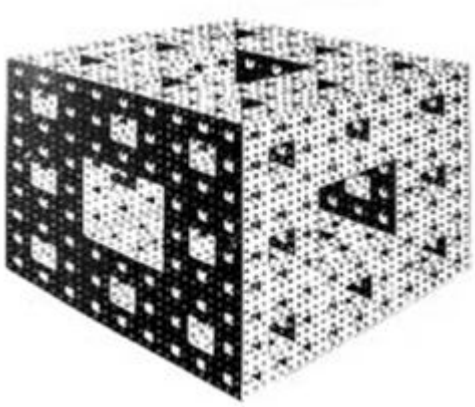
Samosličnost je u velikom broju fraktala očita. Potpuno samoslični kristali su oni koji sadrže kopije samih sebe koje su slične cijelom fraktalu. Dakle, povećanjem određenog dijela možemo vidjeti da je on identičnog oblika kao početni dio.



Slika 2.5: Hilbertova krivulja (druga iteracija



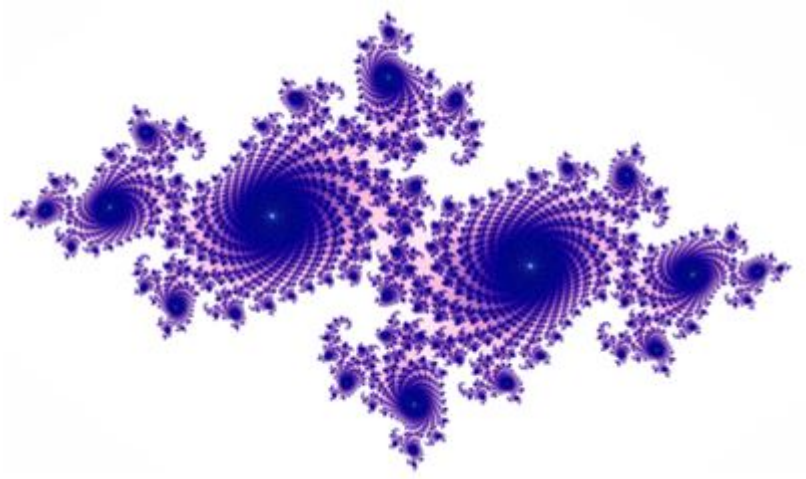
Slika 2.6: Apolonijeva mreža



Slika 2.7: Mengerova spužva

U skupinu potpuno samosličnih fraktala svrstavamo Cantarov skup, Kochovu krivulju, trokut Sierpińskog, Hilbertovu krivulju (Slika 2.5), Apolonijevu mrežu (Slika 2.6), Mengerovu spužvu (Slika 2.7) i mnoge druge.

Kvazi samoslični fraktali se dobivaju iz rekurzivnih relacija, odnosno funkcija koje pozivaju same sebe. Takvi fraktali sadrže male kopije sebe koje nisu potpuno identične cijelom fraktalu već se pojavljuju u djelomično iskrivljenom obliku. Kao primjeri takvih fraktala navode se Mandelbrotov i Julijin skup (Slika 2.8) u kojima nije moguće odmah uočiti identične dijelove, ali povećanjem tih pojedinih dijelova možemo primijetiti manje verzije početne figure.



Slika 2.8: Julijin skup

Kod statički samosličnih fraktala ne pojavljuju se umanjene kopije samog fraktala, ali moguće je uočiti neka njegova svojstva, kao što je fraktalna dimenzija, koja ostaje jednaka pri različitim mjerilima i procjenama. Kao primjer takvog fraktala ističe se Perlinov šum (Slika 2.9).



Slika 2.9: Perlinov šum

Prema načinu nastanka fraktale dijelimo na:

- iterativne,
- rekurzivne,
- slučajne.

Iterativni fraktali ili drugim nazivom sustavi iterativnih funkcija (kraće IFS) nastaju kopiranjem, homotetijom, rotacijom i/ili translacijom kopije te mogućim zamjenjivanjem pojedinih elemenata kopijom. Fraktali nastali iteracijom su potpuno samoslični i još se nazivaju geometrijskim fraktalima. Takve fraktale možemo također razlikovati te ih dodatno klasificiramo na geometrijske fraktale na pravcu, u ravnini i u trodimenzionalnom prostoru.

Rekurzivni fraktali ili fraktali definirani rekurzivnim relacijama su oni koji su određeni rekurzivnom matematičkom formulom koja određuje pripada li određena točka prostora skupu ili ne. U tako određene fraktale spadaju kvazi samoslični fraktali. Dodatno se još nazivaju i algebarskim kristalima.

Fraktali nastali crtanjem grafova nekih stohastičnih procesa, kao što je Brownovo gibanje, munje, drveće i dr. nazivaju se slučajni fraktali (Slika 2.10). Takvi fraktali imaju najmanji stupanj samosličnosti i najčešće ih možemo pronaći u prirodi.

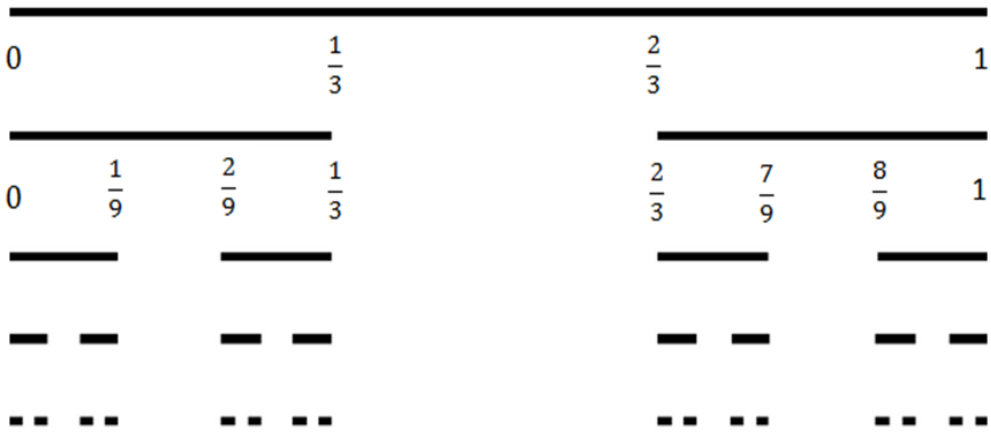


Slika 2.10: Primjeri slučajnih fraktala (munje i drveće)

2.4 Cantorov skup

Jedan od prvih matematičkih primjera fraktala iz kojeg su nastali mnogo složeniji fraktali jest Cantorov skup. Ovaj fraktal naziv je dobio po poznatom njemačkom matematičaru Georgu Cantoru.

Cantorov skup može se konstruirati na sljedeći način: Iz intervala $[0, 1]$ makne se interval $[\frac{1}{3}, \frac{2}{3}]$, potom iz preostalih dvaju intervala ukloni se njihova središnja trećina te se dobivaju intervali $[0, \frac{1}{9}]$ i $[\frac{2}{9}, \frac{1}{3}]$, odnosno $[\frac{2}{3}, \frac{7}{9}]$ i $[\frac{8}{9}, 1]$, itd. Ono što se može uočiti jest da se taj postupak može ponavljati beskonačno mnogo puta jer se svaku od novonastalih trećina može dodatno podijeliti na trećine. Nakon $n-$tog koraka podjele imamo 2^n zatvorenih intervala čija je ukupna dužina jednaka $\frac{1}{3}$.

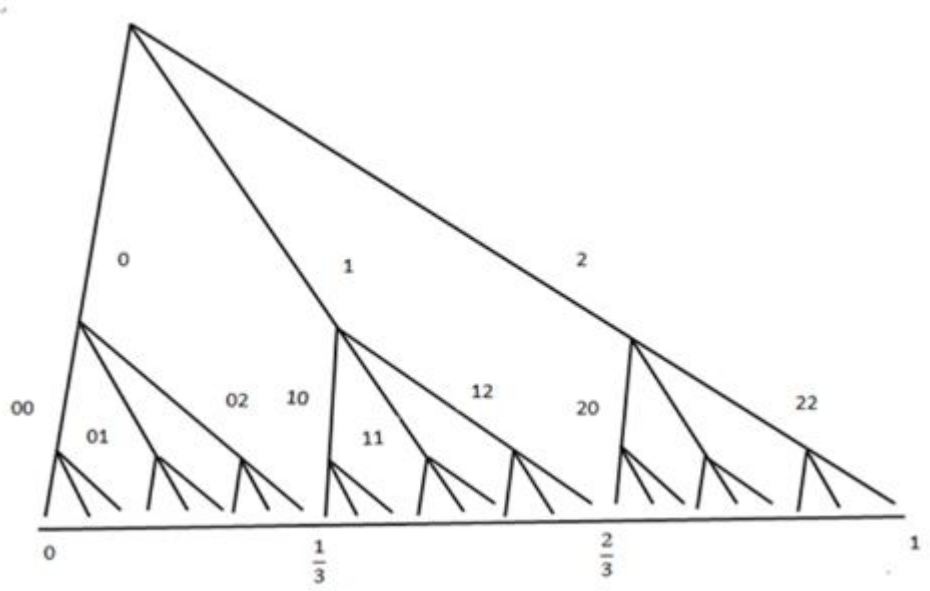


Slika 2.11: Prikaz konstrukcije Cantorovog skupa [0,1]

Označimo li sa C Cantorov skup, tada je točka x iz tog skupa upravo točka koja preostaje nakon $n-$tog koraka.

Za prvih nekoliko koraka taj skup se može prikazati i vizualno, no nakon nekog vremena ako nemamo mogućnost upotrebe računala nastaje problem sa samim grafičkim prikazom Cantorovog skupa (Slika 2.11).

Bilo koji broj iz skupa $[0, 1]$ može se raspisati s bazom 3. Taj raspis možemo vizualno prikazati pomoću grananja (Slika 2.12).



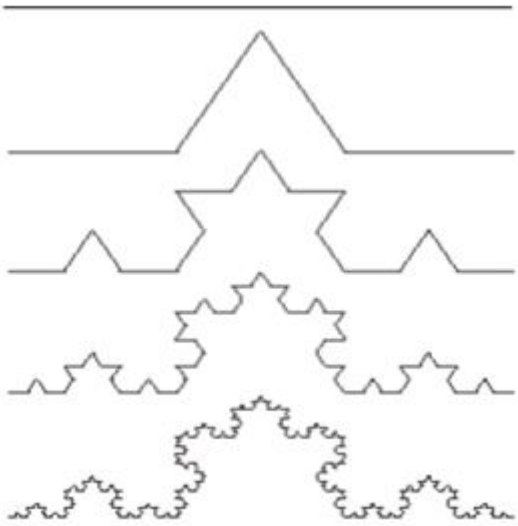
Slika 2.12: Prikaz razvoja brojeva iz intervala [0,1] zapisanih s bazom 3

2.5 Kochova krivulja i Kochova pahuljica

Uz Cantorov skup, jedan od najpoznatijih fraktala jest svakako onaj iz 1904. godine kojeg je predstavio švedski matematičar Niels Fabian Helge von Koch, tzv. Kochova krivulja.

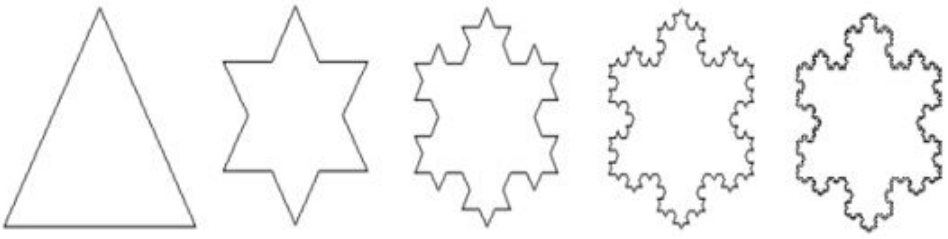
Kochova krivulja može se konstruirati na način da se neka linija podijeli na tri jednaka dijela, a potom srednja trećina te linije zamijeni sa dvjema novim linijama duljina $\frac{1}{3}$. Svaka od tih dviju linija zatvara kut od $60°$. Isti postupak ponavlja se za svaku od novonastalih linija. Ponavljanjem tog postupak beskonačno mnogo puta dobiva se Kochova krivulju (Slika 2.13). Kao zanimljiva karakteristika ove krivulje je da se ni u jednoj njezinoj točki ne može povući tangenta.

Analognim postupkom može se konstruirati Kochova pahuljica. Umjesto krivulje, kao početni korak uzima se jednakostranični trokut. Svaka stranica jednakostraničnog trokuta



Slika 2.13: Koraci u konstrukciji Kochove krivulje

podijeli se na tri jednaka dijela. Na području srednje trećine svake stranice prvobitnog trokuta konstruira se novi jednakostranični trokut čije su stranice duljine $\frac{1}{3}$ prethodne. Ponavljanjem tog postupka za svaku stranicu novog trokuta dobivamo zatvorenu krivulju koju se naziva Kochova pahuljica (Slika 2.14).

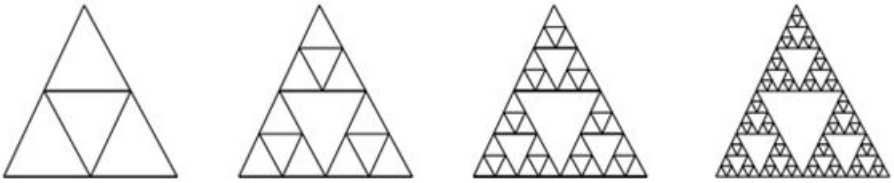


Slika 2.14: Koraci u konstrukciji Kochove pahuljice

Zanimljiva činjenica koja se javlja kod Kochove pahuljice je ta da joj se svakom novom iteracijom njezin opseg sve veći i veći, dok joj je površina ograničena kružnicom koja opisuje početni trokut.

2.6 Trokut Sierpińskog

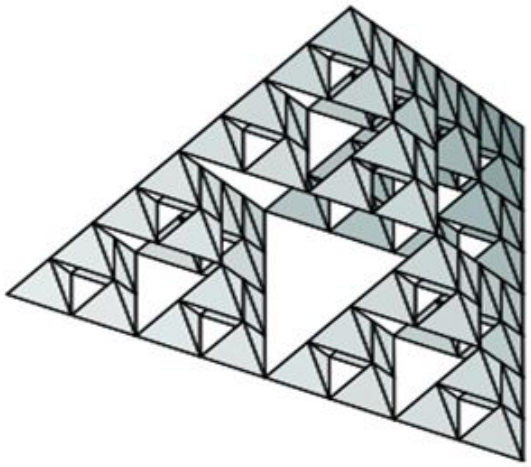
Trokut Sierpińskog vjerojatno je najeksponiraniji (nalazi se na odjeći, bilježnicama, plakatima, zgradama, crtežima,...) fraktal, a ljudi uopće nisu svjesni da ga viđaju svakodnevno.



Slika 2.15: Koraci u konstrukciji trokuta Sierpińskog

Naziv je dobio po poljskom matematičaru Wacławu Franciszsku Sierpińskom koji ga je prvi konstruirao 1915. godine. Njegova konstrukcija (Slika 2.15), baš kao ni konstrukcija Cantarovog skupa i Kochove pahuljice, nije nimalo teška, a započinje tako da se unutar početnog jednakostraničnog trokuta napravi novi jednakostraničan trokut čiji se vrhovi nalaze u polovištima stranica početnog trokuta. Trokut koji se dobije na takav način izbaci se van, pa se lakše mogu uočiti tri jednakostranična trokuta koji su ostali unutar početnog trokuta. Ponavljanjem ovog postupka već u sljedećem koraku imamo 9 jednakostraničnih trokuta, zatim 27, 81,... Postupak možemo ponavljati beskonačno mnogo puta.

Analogijom trokuta Sierpińskog možemo konstruirati i tetraedar (Slika 2.16), odnosno prebaciti se u trodimenzionalni prostor.



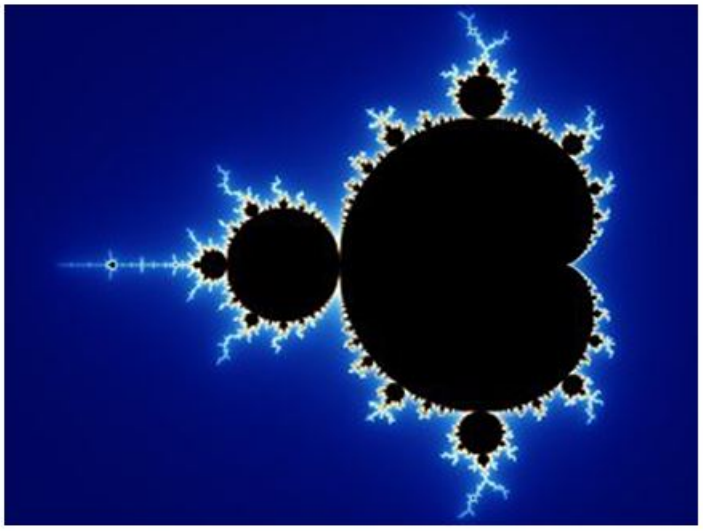
Slika 2.16: Tetraedar Sierpińskog

2.7 Mandelbrotov skup

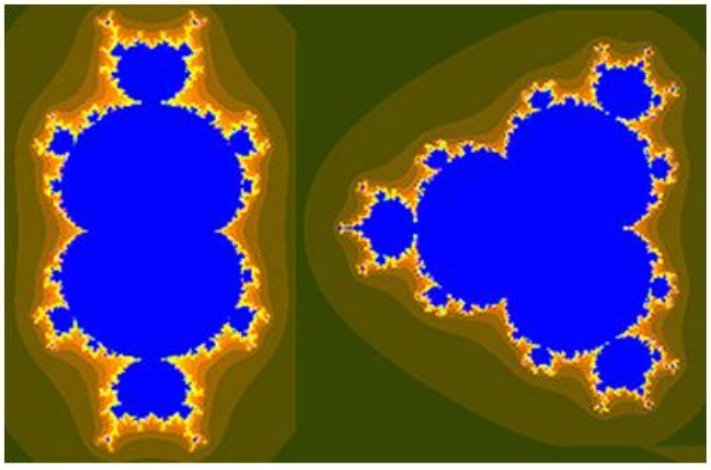
Mandelbrotov skup smatra se jednim od najkompleksnijih, a istodobno najljepšim i najsavršenijim fraktalom, koji ima svojstvo potpune samosličnosti, odnosno sa svakim povećanjem

uočavaju se oblici koji su identični početnom.

Mandelbrotov skup je zapravo zatvoren skup koji obuhvaća samo točno zadane točke u ravnini. Prikažemo li te točke crnom bojom, a sve ostale bijelom (ili ih uopće ne nacrtamo) dobivamo sliku koja predstavlja taj neobičan i očaravajući – Mandelbrotov fraktal (Slika 2.17). Postoje i različite varijacije na temu Mandelbrotovog skupa, tzv. multibrot skupovi (Slika 2.18).



Slika 2.17: Mandelbrotov skup



Slika 2.18: Varijacije 3. i 4. stupnja Mandelbrotovog skupa, tzv. multibrot skupovi

2.8 Fraktalna dimenzija

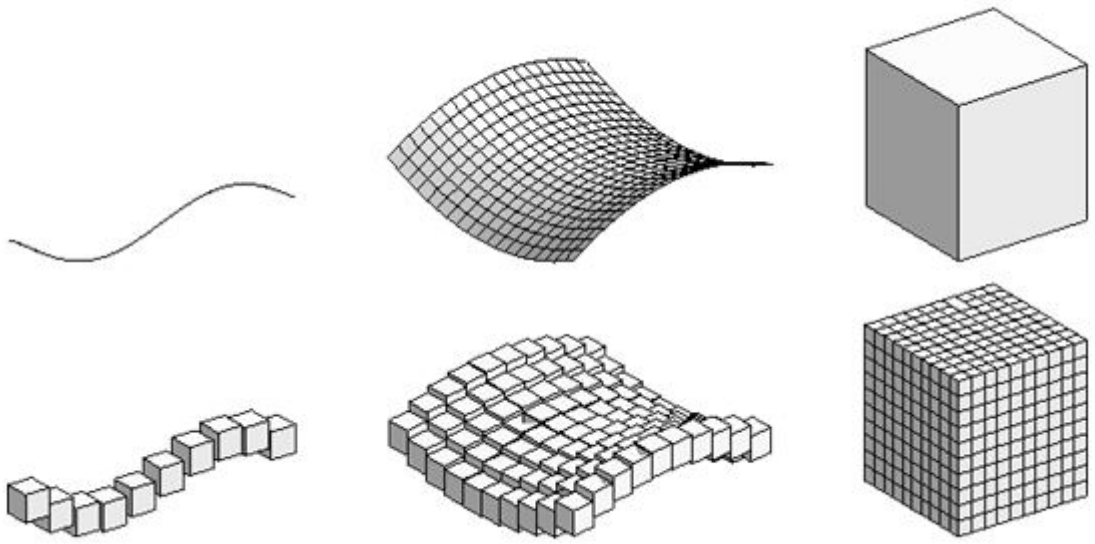
Da bi se mogao razumjeti pojam fraktalne dimenzije, mora se shvaćati što općenito predstavlja pojam dimenzije. Sam pojam dimenzije prisutan je u matematici, prvenstveno u geometriji, još od početaka njezina razvoja, a kasnije se pojavio i u fizici.

Dimenzija je izraz koji prikazuje kako se neka fizikalna veličina može izraziti pomoću minimalnog broja međusobno nezavisnih varijabli koje su potrebne da se opiše neki oblik. Takva dimenzija naziva se još i Euklidova dimenzija. Iz euklidske geometrije poznato je da točka nema dimenziju, pravac ima jednu dimenziju (duljinu), ravnina ima dvije dimenzije (duljinu i širinu), a geometrijska tijela imaju tri dimenzije.

Za poznate geometrijske oblike, kao što su dužina, kvadrat i kocka, nije problem odrediti dimenziju. Ona može biti bilo koji prirodni broj, uključujući i nulu.

U odnosu na Euklidovu dimenziju, fraktalna dimenzija je dimenzija koja nam govori u kojoj mjeri neki fraktal ispunjava prostor u kojem se nalazi, odnosno koliko se novih dijelova pojavljuje prilikom povećanja danog fraktala. Također, ta dimenzija ne mora biti cijeli broj i najčešće je veća od Euklidove dimenzije.

Poznato je više metoda kojima se može izračunati fraktalna dimenzija, a neke od njih su metoda samosličnosti, metoda šestara, metoda brojanja kvadratića, metoda brojanja kockica,... Detaljnije ću opisati način izračuna fraktalne dimenzije pomoću metode brojanja kockica.



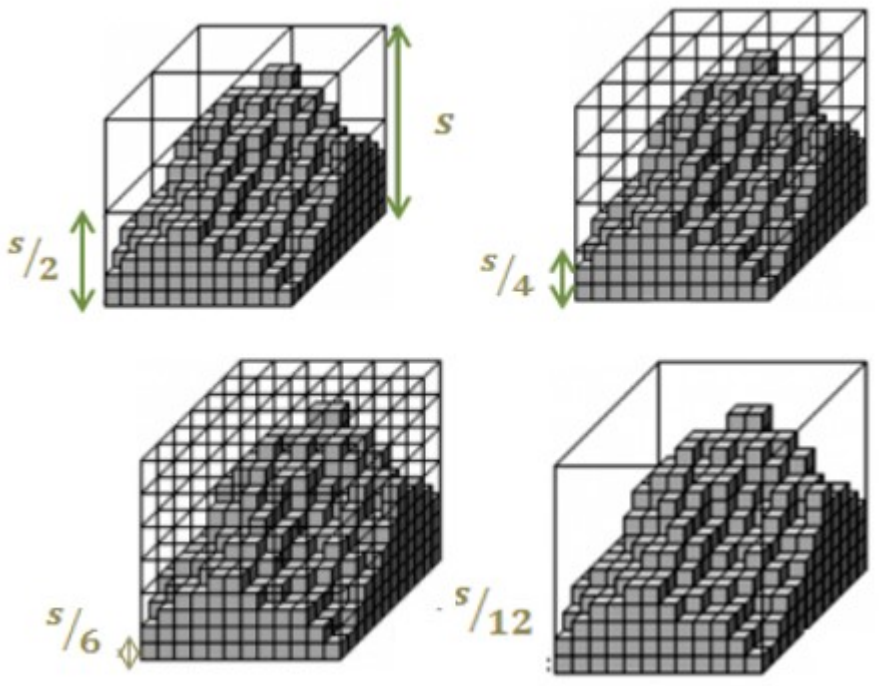
Slika 2.19: Primjer ”metode brojanja kockica” za liniju, plohu i kocku

Metoda brojanja kockica ili engl. *Box counting method* (Slika 2.19) koristi se kod izračuna fraktalne dimenzije nekog nepravilnog objekta na način da se on opiše pomoću poznatih pravilnih objekata. Pravilne objekte u takvom izračunu predstavljaju kvadratići, za dvodimenzionalnu analizu, i kockice, za trodimenzionalnu analizu.

Odaberimo neki nepravilni objekt u trodimenzionalnom prostoru kojem želimo izračunati fraktalnu dimenziju i prekrijmo ga mrežom kockica čija je duljina brida dana sa s. Sada trebamo prebrojiti kockice koje su potrebne da bi se prekrio odabrani nepravilni objekt. Taj broj označimo sa N. Nakon toga smanjujemo duljinu bridova kockica, a time i samu veličinu kockica (Slika 2.20). Ono što želimo istražiti je kako se mijenja broj kockica u odnosu na veličinu tih kockica.

Uzmimo liniju sa slike 2.19. Znamo da je linija jednodimenzionalna. Većina ljudi će pretpostaviti da će se broj kockica koje prekrivaju liniju povećati smanjivanjem veličine same kockice te će tu ovisnost zapisati kao

$$N(s) = \frac{1}{s}. \tag{2.1}$$



Slika 2.20: Primjer promjene broja kockica ako se mijenja njihova veličina

No, je li ta ovisnost točno napisana? Nije. Primijetimo da je ploha koju mi moramo izračunati zapravo dvodimenzionalna, pa prema tome za nju ne vrijedi ovisnost koja je dana za liniju. Ovisnost broja kockica o veličini tih istih kockica za plohu je dana izrazom

$$N(s) = \left(\frac{1}{s}\right)^2. \tag{2.2}$$

S obzirom da eksponent predstavlja dimenziju objekta kojeg analiziramo, možemo zaključiti da za bilo koji objekt vrijedi jednadžba

$$N(s) = k \times \left(\frac{1}{s}\right)^d. \tag{2.3}$$

Kako je fraktalna dimenzija sakrivena u eksponentu, tada moramo logaritmirati lijevu i desnu stranu jednadžbe. Time dobivamo izraz

$$log(N(s)) = d \times log\left(\frac{1}{s}\right) + log(k). \tag{2.4}$$

Kako $s \to 0$, tada dobivamo da je formula za izračun fraktalne dimenzije metodom prebrojavanja kockica dana izrazom

$$d_b = \frac{log(N(s))}{log\left(\frac{1}{s}\right)} \tag{2.5}$$

Crtanjem grafa $log(N(s)) - log(1/s)$ možemo vidjeti da nagib predstavlja fraktalnu dimenziju zadanog objekta.

2.9 Fraktali i fraktalna dimenzija u nastavi

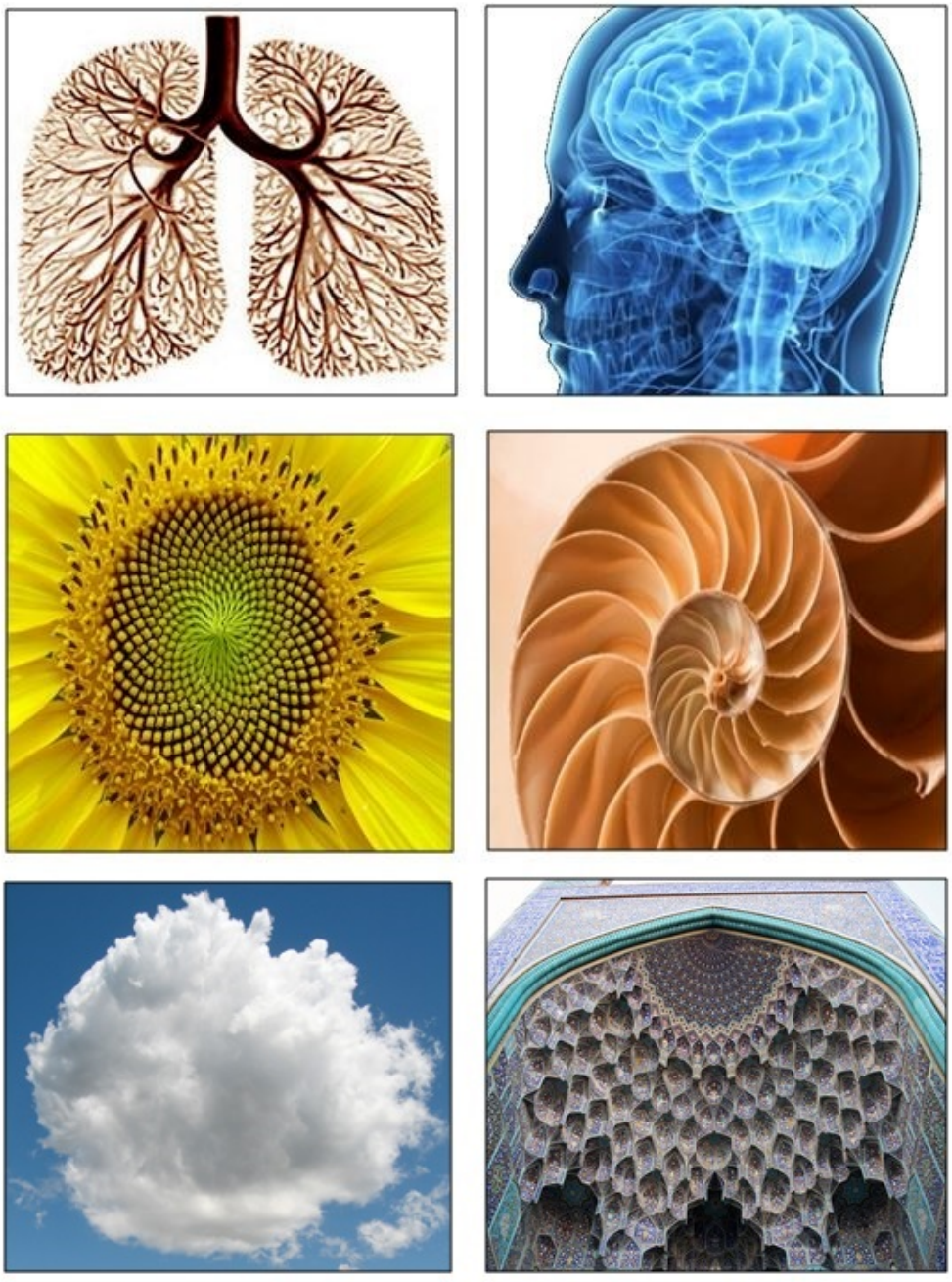
Kao nastavna cjelina, fraktali su tek neznatno obuhvaćeni planom i programom nastave fizike na samom kraju srednjoškolskog obrazovanja u općim i prirodoslovno – matematičkim gimnazijama. Kako ta nastavna cjelina dolazi na kraju četvrtog razreda srednje škole najčešće se ne stigne podrobnije obraditi, a time učenici na neki način ostaju zakinuti za mogućnost samostalnih radova i proširivanja vlastitog znanja iz fizike.

Za razliku od nastave fizike, geometrija fraktala nije obuhvaćena planom i programom nastave matematike (iako bi se mogla spomenuti npr. kod kompleksnih brojeva i beskonačnog geometrijskog reda).

Mnogi nastavnici učenicima, većinom zbog njihove preopterećenosti nastavnim sadržajima i nezainteresiranosti nastavom, ne spominju razne zanimljivosti iz svakodnevnog života s kojima bi se mogli povezati pojam fraktala i fraktalne dimenzije (Slika 2.21).

Upravo zbog nedostatka vremena za detaljniju obradu tih pojmova te da bi ih se zainteresiralo za nastavu, učenicima bi se mogli podijeliti projekti (npr. u obliku domaće zadaće u trajanju tjedan dana) u kojima bi oni samostalno dodatno proučavali geometriju fraktala i povezali fraktale sa stvarnim prirodnim pojavama i oblicima (Tablica 2.1).

Kako uvesti pojam fraktala i fraktalne dimenzije na satu fizike? Povezivanjem kompleksnih funkcija kao što su fraktali sa svakodnevnim prirodnim pojavama i oblicima što nas okružuju, nerazumljivi i nejasni predmeti poput matematike i fizike svakom učeniku mogu postati zanimljivima i očaravajućima. Nastavni sat na tu temu može se započeti s primjerima iz svakodnevnog života, npr. na projektoru prikazati sliku lista ili brokule, a potom pitati učenike što pretpostavljaju da će vidjeti ako se zumira dani objekt? Hoće li se dijelovi brokule promijeniti? Što smatraju da će vidjeti na listu njegovim povećanjem? Nakon iznošenja učeničkih pretpostavki i nakon povećanja objekta, učenike svakako treba pitati da usporede svoje pretpostavke s onim što su vidjeli. Nakon primjera objekata fraktalne prirode iz svakodnevnog života, pokazati nekoliko matematičkih primjera fraktala kao što su Kochova pahuljica, trokut Sierpińskog, Cantorov skup, Mandelbrotov skup, Julijin skup, Pitagorino stablo, Hilbertovu krivulju, Peanovu krivulju,... Objasniti kako se mogu konstruirati pojedini fraktali te što su općenito fraktali. Za domaću zadaću učenicima se može zadati da provedu npr. prvih pet koraka konstrukcije za Kochovu pahuljicu, Cantorov skup, Peanovu krivulju ili Pitagorino stablo. Neka razmisle kakav lik bi dobili za prvih sto iteracija ili neka probaju prikazati računalom. Daljnjim provođenjem interaktivne nastave pitati učenike imaju li fraktali dimenziju. Prije uvođenja fraktalne dimenzije, učenici bi sami trebali pokušati objasniti što je pojam dimenzije u fizici i razlikuje li se on od matematičkog pojma te ga potkrijepiti primjerima. Tek nakon iznošenja učeničkih mišljenja definirati fraktalnu dimenziju i na primjeru objasniti kako ju izračunati. Kroz pokazane primjere, učenici mogu uočiti važnost fraktala. Napomenuti im da se fraktalna dimenzija koristi u geoznanostima (kod predviđanja pokusa, uragana, poplava, vulkanskih erupcija,...), neuroznanosti, dijagnostici,...



Slika 2.21: Primjeri fraktala u nama i oko nas (pluća, mozak, cvijet, školjka, oblak, zgrade)

PROJEKT:	**FRAKTALI I FRAKTALNA DIMENZIJA**
ZADATAK:	Izračunati fraktalnu dimenziju danog objekta.
PRIBOR I MATERIJAL:	Ravnalo, slika neke prirodne pojave ili oblika koja ima svojstva fraktalne geometrije, računalo
UPUTE ZA RAD:	Sliku podijelite na 8 ili 16 identičnih kvadrata, a zatim svaki od tih kvadrata podijelite na jednaki broj manjih kvadratića (ne većih od 2 mm) tako da dobijete veću i manju kvadratnu mrežu. Zatim prebrojite koliko ste zauzetih kvadratića dobili na većoj, a koliko na manjoj mreži. Dobivene brojeve označite redom s N_1 i N_2. Pomoću formule $d = \frac{log\left(\frac{N_2}{N_1}\right)}{log\left(\frac{a_1}{a_2}\right)}$ izračunajte fraktalnu dimenziju danog oblika. Grafički prikažite ovisnost broja kvadratića o dimenziji stranice kvadratića (x-os predstavlja logaritmiranu vrijednost stranice kvadratića, oznake $log\left(\frac{1}{a}\right)$, a y-os logaritmiranu vrijednost zauzetih kvadratića, oznake $log(N)$). Usporedite vrijednosti dobivene računski s vrijednostima koje vam se javljaju u jednadžbi pravca. Što uočavate?
DISKUSIJA:	Pismeno prokomentirajte dobivene rezultate, pritom svakako prokomentirajte i sljedeća pitanja: • Što ste naučili iz ovog projekta? • Što ste naučili o fraktalima? • Imaju li fraktali dimenziju?

Tablica 2.1: Primjer zadavanja projekta učenicima

Poglavlje 3

Projekt: Trodimenzionalna fraktalna analiza reljefa u kršu

3.1 Instaliranje programske podrške

Upotreba računala omogućuje nam jednostavan i brži izračun i prikaz fraktalne dimenzije raznih kompleksnih objekata. U ovom radu krenuli smo od programa za proračun 2D fraktalne dimenzije (fraktalna analiza slike) koji smo proširili na 3D (Radić, 2013). Središnji dio programa je metoda brojanja kvadratića (objašnjena u 3. poglavlju). To je u osnovi jednostavna metoda, ali ju treba koristiti s oprezom zbog odgovarajućeg podešavanja parametara. Za isti set podataka potrebno je napraviti više analiza kako bi se utvrdili optimalni parametri.

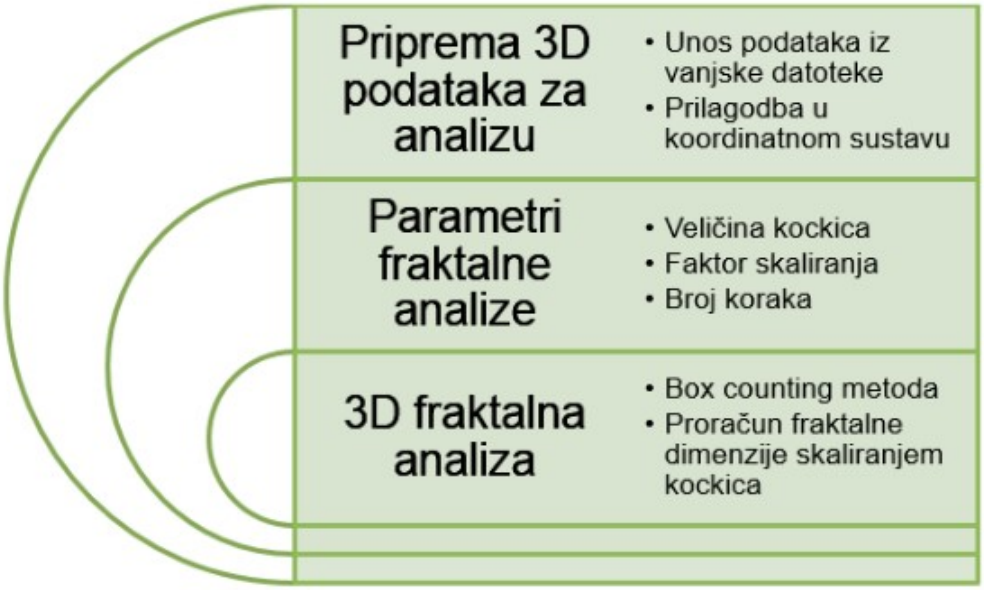
Python je logičan izbor programskog jezika (zbog svoje jednostavnosti i dostupnosti) posebice ako se cilja na upotrebu u osnovnim i srednjim školama. Odabrana je njegova verzija 2.7 iz razloga kompatibilnosti s pojedinim paketima koji su se koristili pri proračunu fraktalne dimenzije. Da bi se program mogao pokrenuti potrebni su paketi matplotlib, math, NumPy, SciPy.Stats, decimal, os i re koji su omogućili proračun i obradu podataka u obliku grafova i slika. Svi paketi su dostupni preko python.org.

Inačice svih aplikacija i biblioteka korištenih kod proračuna fraktalne dimenzije i analize slika:

- Python 2.7
- NumPy 1.7.1
- Matplotlib 1.3.1
- Math
- Decimal
- Os
- Re
- SciPy

3.2 Program za proračun 3D fraktalne dimenzije

Kod programa dan je u Prilogu. Osnovni dijelovi programa prikazani su na slici (Slika 3.1). Neki dijelovi koda posebno su istaknuti u ovom potpoglavlju.



Slika 3.1: Osnovni dijelovi programa

Priprema 3D podataka za analizu predstavlja ključan dio programa. Za potrebe ovog rada ustupljeni su podaci o pružanju špiljskih kanala najdulje hrvatske špilje koja se zove Jamski sustav Kita Gaćešina – Draženova puhaljka. Trodimenzionalna struktura prikazana je kao niz podataka u Kartezijevom koordinatnom sustavu. Za potrebe ove analize bitno je samo u kojim točkama prostora postoje špiljski kanali i koje su koordinate tih točaka. Dani podaci dostupni su u obliku ulazne tekstualne datoteke *kita*8.*xyz*. Prva tri stupca u toj datoteci predstavljaju koordinate navedenih točaka.

Stoga prvi dio programa priređuje te podatke za daljnju analizu: učitava ih iz datoteke i pomiče u prvi kvadrant radi jednostavnije analize. Analizirana špilja ili dio njenih kanala se stavljaju u prostor omeđen kvadrom (koji kreće iz ishodišta duljina stranica *kxmax*, *kymax* i *kzmax*).

```
f = open(filename)
for line in f:
    floatline = [float(xnumber) for xnumber in re.findall(match_number,
    line)]
```

```
    kx.append(floatline[0])    # x koordinata
    ky.append(floatline[1])    # y koordinata
    kz.append(floatline[2])    # z koordinata
    i=i+1;
    if i==imax:
        break
f.close()
```

Slijedi odabir parametara za analizu te proračun 3D fraktalne dimenzije metodom brojanja kockica (engl. *box counting method*). U ovom dijelu programa od korisnika se traži da upiše početnu veličinu kockice u metodi brojanja kockica, faktor skaliranja kockica i broj točaka na grafu koji će se koristiti za proračun fraktalne dimenzije numeričkom prilagodbom. Da bi analiza bila što točnija preporuča se unos što većeg broja točaka, no to treba uskladiti s vremenom koje je raspoloživo za analize (posebno ako učenici analize izvode u okviru nastave). Faktor skaliranja određuje za koliko ćemo povećavati kockicu u sljedećem koraku analize. Metodom brojanja kockica promatra se za koliko se promijeni broj kockica potreban za prekrivanje promatrane špilje kada se promijeni veličina kockica.

```
# proracun 3D fraktalne dimenzije uporabom Box counting metode
b=int(raw_input('Unesite pocetnu velicinu kockica u pikselima b:'))
f=int(raw_input('Unesite faktor skaliranja kockica f:'))
print 'Maksimalni broj tocaka za izracun je nmax=',
int(math.log(imgx)/math.log(f))
n=int(raw_input('Unesite broj tocaka za izracun n:'))
```

Dio programa za analizu prirodne pojave definira veličinu kockica (duljina brida u pikselima) i odgovarajućim petljama prolazi kroz 3D prostor i provjerava da li se dio špiljskog kanala nalazi određenoj kockici. Ako je špiljski kanal pronađen u određenoj kockici, povećava se brojač kockica. Cijeli postupak se ponavlja dok kockicama ne prekrijemo cijeli analizirani dio špilje. Program je ovdje dodatno prilagođen tako da analizira samo dio prostora u kome se nalazi špilja (na temelju podataka o pružanju špiljskih kanala učitanih iz datoteke), a ne pretražuje velike dijelove prostora gdje nema ničega (što bi zahtijevalo puno veće memorijske i procesorske zahtjeve).

```
for ib in range(n):
    bs = b * f ** ib # velicina kockice u pikselima
    bnx = int(imgx / bs) # broj kockica u x smjeru slike
    bny = int(imgy / bs) # broj kockica u y smjeru slike
    bnz = int(imgz / bs) # broj kockica u z smjeru slike
    print 'Broj kockica x, y, z', bnx, bny, bnz
```

```
if bnx<2 or bny<2 or bny<2:
    break
ix=[]
iy=[]
iz=[]
ikx=0
boxCount = 0
for bx in range(bnx):
    bxmin=bs*bx
    bxmax=bs*bx+bs
    for ib in range(nrpoints):
        kxyz_x=kxyz[ib,0]
        if (bxmin<=kxyz_x) and (kxyz_x<bxmax):
            ikx += 1 # generiram listu tocaka sa zadanim x uvjetom
            ix.append(ib)
    if ikx == 1:
        boxCount += 1
    else:   # eliminacija tocaka koje se nalaze u istim kockicama
        iydel=[]
        for i2 in range(ikx):
            for i3 in range (i2+1,ikx):
                kxyz_y1=kxyz[ix[i2],1]
                kxyz_y2=kxyz[ix[i3],1]
                if abs(kxyz_y1-kxyz_y2)<bs:
                    kxyz_z1=kxyz[ix[i2],2]
                    kxyz_z2=kxyz[ix[i3],2]
                    if abs(kxyz_z1-kxyz_z2)<bs:
                        iydel.append(i3)

        for i2 in range(ikx):
            if i2 not in iydel:
                boxCount += 1

print "bnx=",bnx," bny=",bny," bs=",bs
print "ib=",ib,"  boxCount=",boxCount
if boxCount>0:
    gx.append(math.log(1.0 / bs))
    gy.append(math.log(boxCount))
```

Nakon svakog prolaska kroz cijeli promatrani dio prostora i prije idućeg koraka u kojem

se povećaju kockice, podaci se upisuju u listu pomoću funkcije *append*(). U listu *gx* se upisuju logaritamske vrijednosti od 1/bs, gdje je *bs* varijabla koja predstavlja veličinu kockica. U listu *gy* se upisuju logaritamske vrijednosti broja kockica, tj. brojača *boxCount*. Program ispisuje i veličinu kockica (duljina brida u pikselima) i broj kockica u x, y i z smjeru, kao i pronađeni broj kockica u kojima se nalaze špiljski kanali. Na taj način možemo pratiti program i saznati što se u kojem prolazu kroz petlju događalo.

Rekli smo da je na početku dijela koda za prebrojavanje kvadratića definiran broj točki na grafu potrebnih za linearnu regresiju. To je ujedno broj prolaza kroz prvu for petlju. Nakon što je program izvrtio dio koda gdje broji kvadratiće izlazi iz *for* petlje i kreće u numeričku prilagodbu ovisnosti *gy* o *gx*.

```
slope, intercept, r_value, p_value, std_err=linregress(gx,gy)
getcontext().prec = 1 #postavlja preciznost pogreske na
                        1 znamenku
Derr=Decimal(std_err)

a=Derr
i=0
while (a%1 != 0):  #nalazi broj znamenki u pogresci
 a=a*10
 i=i+1

D=round(slope, i) #zaokruzuje dimenziju na broj znamenki
                    kao kod   pogreske
postotak=(float(Derr)/D)*100  #daje pogresku u postotku

print "Procijenjena fraktalna dimenzija: " + str(slope)
print "Pogreska: ", +std_err

print ""
print 'D ± Derr: %s ± %s' % (str(D), round(Derr, i))
print 'Pogreska u postotku:' + str(round(postotak, i)) + '%'
```

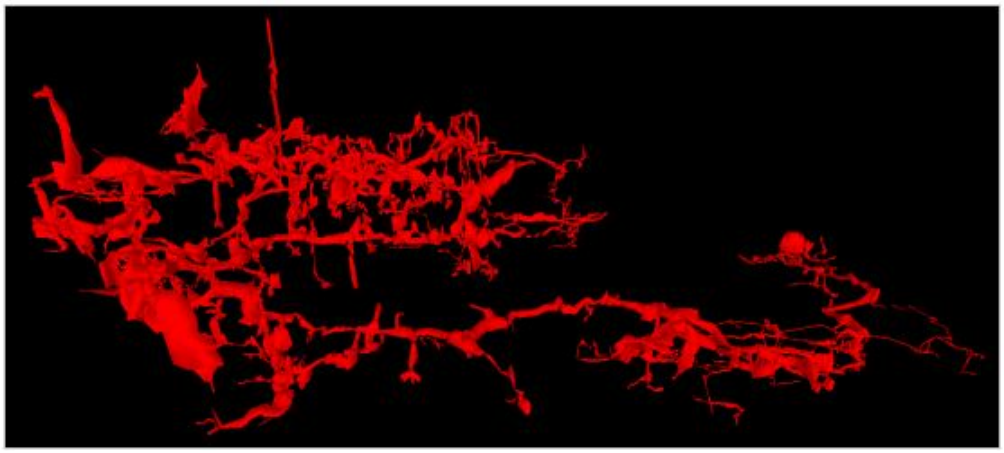
Program izračunava nagib pravca koji je dobio iz točaka koje se nalaze u listama *gx* i *gy*. Numeričkom prilagodbom na pravac dobiva se iznos koeficijenta smjera i njegovu standardnu devijaciju. Kao što smo već spomenuli, taj nagib pravca predstavlja fraktalnu dimenziju slike koju smo analizirali.

Standardna devijacija se zaokružuje na prvu preciznu znamenku, a zatim se fraktalna dimenzija zaokružuje na broj znamenki iza decimalne točke kao kod pogreške. Program također proračunava postotak pogreške i ispisuje ju. Nakon toga se prikazuje graf linearne regresije i sprema na računalo.

```
gx=np.array(gx)
gy=np.array(gy)
points=np.linspace(gx.min(), gx.max(), n)
plt.plot(gx, gy, '.', color='red')
plt.plot(points, slope*points+intercept)
plt.savefig(slika+"-rezultat.jpg")
plt.show()
```

3.3 Primjer proračuna 3D fraktalne dimenzije

Sada ćemo program opisan u prethodnom potpoglavju primijeniti na konkretnom primjeru. Da bi u okviru projektne nastave učenicima pojedina tema bila zanimljivija, ali i da bi se stekla praktična znanja primijenjena na realne primjere iz naše okoline, ovdje je odabrana prirodna pojava u kršu koja već na prvi pogled daje dojam fraktalnosti. Radi se o prirodno formiranoj špilji koja se sastoji od kompleksa špiljskih kanala.



Slika 3.2: 3D prikaz analiziranih podataka u programu MeshLab

Na slici (Slika 3.2) je dan 3D prikaz Jamskog sustava Kita Gaćešina – Draženova puhaljka, najdulje hrvatske špilje koja se nalazi na Velebitu. Ukupna duljina svih prikazanih kanala je , a dubina . Kompleksna mreža kanala nastala je složenim geološkim procesima u kršu (procesi speleogeneze – formiranja špilja). Cilj našeg istraživanja će biti da ustanovima radi li se o 3D objektu ili fraktalu. Ako fraktalna dimenzija neće biti cjelobrojna (cjelobrojne su prostorne dimenzije 1, 2, 3) zaključak će biti da se radi o fraktalu te ćemo imati primjer fraktala kojeg su formirali prirodni procesi. Slijedi output iz programa koji sadrži unesene parametre i proračunate varijable te pravac linearne regresije (Slika 3.3) za profil Jamskog sustava Kita Gaćešina – Draženova puhaljka (za 1000 točaka):

```
Broj tocaka u datoteci: 68259
Koliko tocaka obraditi: 1000

granice x  0.0 384.75
granice y  0.0 1057.949974
granice z  0.0 802.460022

Unesite pocetnu velicinu kockica u pikselima b:5
Unesite faktor skaliranja kockica f:2
Maksimalni broj tocaka za izracun je nmax= 8
Unesite broj tocaka za izracun n:10
```

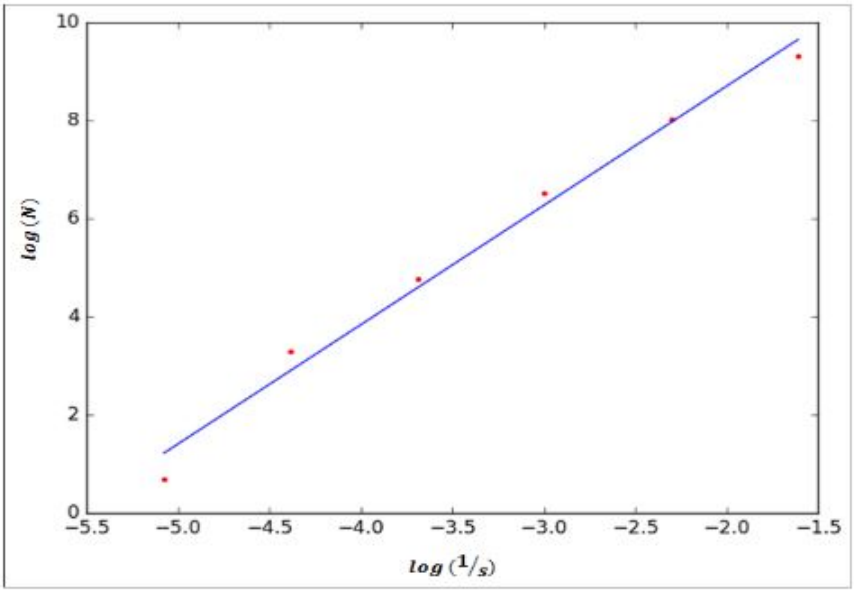
Ispis koji smo dobili pomoću dodatnih varijabli kako bismo mogli pratiti prolazak programa kroz for petlje:

```
Broj kockica x, y, z   76  211 160
bs= 5  ib= 999   boxCount= 11046
Broj kockica x, y, z 38 105 80
bnx= 38  bny= 105  bs= 10
ib= 999   boxCount= 3033
Broj kockica x, y, z 19 52 40
bnx= 19  bny= 52  bs= 20
ib= 999   boxCount= 682
Broj kockica x, y, z 9 26 20
bnx= 9  bny= 26  bs= 40
ib= 999   boxCount= 118
Broj kockica x, y, z 4 13 10
bnx= 4  bny= 13  bs= 80
ib= 999   boxCount= 27
Broj kockica x, y, z 2 6 5
bnx= 2  bny= 6  bs= 160
ib= 999   boxCount= 2
```

Oznake *bnx* i *bny* su veličine kvadratića u x i y smjeru slike, *bs* je veličina kvadratića u pikseliima, *ib* je broj pronalazaka kroz glavnu for petlju, odnosno broj točke na grafu, a *boxCount* je broj kvadratića potrebnih za prekrivanje slike fraktala. Nakon ispisa računa se fraktalna dimenzija i njezina pogreška te se ispisuje postotak pogreške:

```
Procijenjena fraktalna dimenzija: 2.43205982233
Pogreska:  0.138062084209
```

```
D  Derr: 2.4 +- 0.1
Pogreska u postotku:5.8%
```



Slika 3.3: Pravac linearne regresije za profil Jamskog sustava Kita Gaćešina – Draženova puhaljka (za 1000 točaka)

Rezultat proračuna je fraktalna dimenzija $D = (2.4 \pm 0.01)$. Treba napomenuti da nisu analizirani svi špiljski kanali već prvih 1000 točaka. Detaljnija istraživanja koja mogu provesti učenici mogu obuhvatiti analize različitih dijelova špilje (različitih brojeva podataka) te usporediti kako to utječe na fraktalnu dimenziju. S obzirom na općenita svojstva fraktala za očekivati je da to neće bitno promijeniti zaključke. Glavni zaključak je da je dana fraktalna dimenzija necjelobrojna, dakle možemo sa sigurnošću potvrditi da analizirani dio špilje ima fraktalnu strukturu. Usporedimo što će se dogoditi ako imamo tri puta veći broj točaka, dakle 3000 točaka. Slijedi output iz programa koji sadrži unesene parametre i proračunate varijable te pravac linearne regresije (Slika 3.4) za profil Jamskog sustava Kita Gaćešina – Draženova puhaljka (za 3000 točaka):

```
Broj tocaka u datoteci: 68259
Koliko tocaka obraditi: 3000

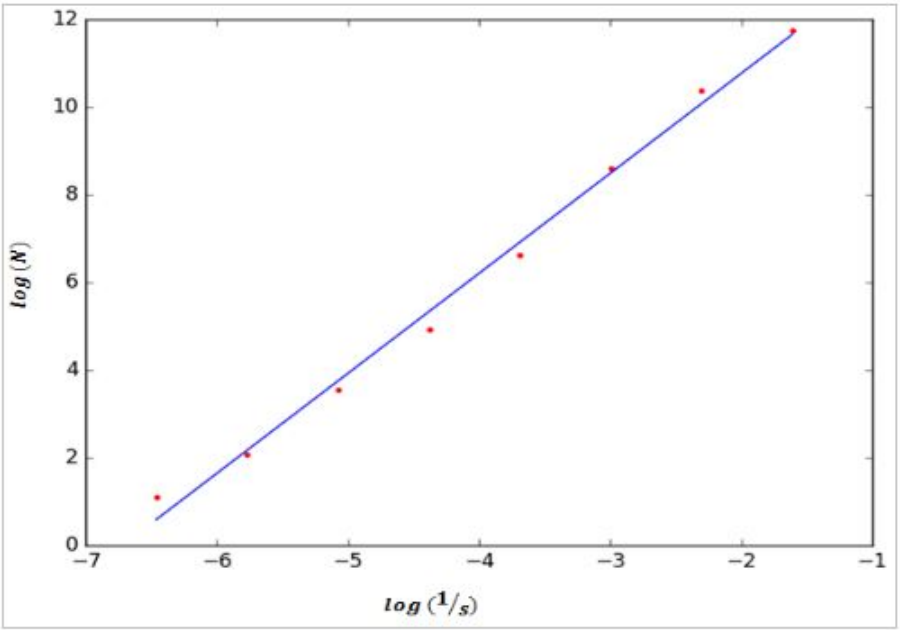
granice x  0.0 1691.829986
granice y  0.0 1402.839966
granice z  0.0 2121.550019
```

```
Unesite pocetnu velicinu kockica u pikselima b:5
Unesite faktor skaliranja kockica f:2
Maksimalni broj tocaka za izracun je nmax= 10
Unesite broj tocaka za izracun n:10

Broj kockica x, y, z   338   280   424
bs= 5 ib= 2999   boxCount= 125194
Broj kockica x, y, z 169 140 212
bnx= 169  bny= 140  bs= 10
ib= 2999   boxCount= 32319
Broj kockica x, y, z 84 70 106
bnx= 84  bny= 70  bs= 20
ib= 2999   boxCount= 5471
Broj kockica x, y, z 42 35 53
bnx= 42  bny= 35  bs= 40
ib= 2999   boxCount= 749
Broj kockica x, y, z 21 17 26
bnx= 21  bny= 17  bs= 80
ib= 2999   boxCount= 136
Broj kockica x, y, z 10 8 13
bnx= 10  bny= 8  bs= 160
ib= 2999   boxCount= 35
Broj kockica x, y, z 5 4 6
bnx= 5  bny= 4  bs= 320
ib= 2999   boxCount= 8
Broj kockica x, y, z 2 2 3
bnx= 2  bny= 2  bs= 640
ib= 2999   boxCount= 3
Broj kockica x, y, z 1 1 1
```

Nakon ispisa računa se fraktalna dimenzija i njezina pogreška te se ispisuje u IDLE prozoru zajedno sa zaokruženim vrijednostima fraktalne dimenzije, standardnom dimenzijom i postotkom pogreške:

```
Procijenjena fraktalna dimenzija: 2.28176970241
Pogreska:  0.0751179242526
D  Derr: 2.28 +- 0.08
Pogreska u postotku:3.29%
```



Slika 3.4: Pravac linearne regresije za profil Jamskog sustava Kita Gaćešina – Draženova puhaljka (za 3000 točaka)

Iz obrade dobivenih podataka dobivamo da fraktalna dimenzija iznosi $D = (2.28 \pm 0.08)$, pa vidimo da povećanjem analiziranog dijela špilje nismo dobili bitno drugačiji rezultat.

Time smo pokazali da je riječ o fraktalnoj strukturi. Ono što bi moglo biti zanimljivo u nekim detaljnim analizama je da se uspoređuju fraktalne dimenzije pojedinih glavnih dijelova špilje koji na temelju geoloških analiza vjerojatno pripadaju različitim geološkim razdobljima. Za očekivati je da bi fraktalna dimenzija mogla detektirati različite faze razvoja špilje. Učenicima se mogu zadati projektni zadaci kako u okviru nastave tako i samostalnog rada posebice u kontekstu povezivanja s poznatim geološkim procesima i prirodnim pojavama (kao na primjeru špiljskih kanala).

PROJEKT:	**3D FRAKTALNA ANALIZA PODATAKA IZ PRIRODE**
ZADATAK:	Upotrebom programa za fraktalnu analizu analizirati dani skup podataka
PRIBOR I MATERIJAL:	Računalo, program za fraktalnu analizu i trodimenzionalni podaci (špiljski kanali kao prirodna pojava u kršu).
UPUTE ZA RAD:	Ponoviti sve glavne značajke Pythona i detaljnije proći kroz glavne segmente programa. Mijenjajući ulazne parametre usporedite dobivene vrijednosti fraktalnih dimenzija. Što zaključujemo iz činjenice da one nisu cjelobrojne? Što najviše utječe na pogrešku mjerenja? Komentirajte pogreške mjerenja. Što ste dobili mijenjajući ulazne parametre?
DISKUSIJA:	Pismeno prokomentirajte dobivene rezultate, pritom svakako prokomentirajte i sljedeća pitanja: • Što ste naučili iz ovog projekta? • Zašto je teško kvantitativno opisati kompleksne prirodne pojave? • Što izračunata dimenzija govori o analiziranom primjeru iz prirode? • Da li bi na isti način mogli uspoređivati proizvoljno odabrane trodimenzionalne objekte?

Tablica 3.1: Primjer projekta o proračunu fraktalne dimenzije špiljskih kanala

Zaključak

Ovaj rad pokazuje mogućnosti osmišljavanja projektne nastave koja povezuje znanja iz različitih područja kao što su matematika, fizika i programiranja s prirodnim pojavama. Uloga projektne nastave je upravo da se učenik stavi u središte nastavnog procesa te da ga se zainteresira za osnovnoškolsku i srednjoškolsku nastavu primjenjujući nova znanja na problemima iz svakodnevnog života. Odabrani problem u slučaju ovog rada je specifičan, to su prirodne pojave u kršu – špilje – za koje smo određivali imaju li prirodnu fraktalnost. No iste ili neke druge odabrane analize možemo primijeniti na različite aspekte proučavanja prirode i svijeta oko nas.

Ovaj rad bi nastavniku trebao pokazati da znanstvena analiza kompleksnih pojava u prirodi ne mora biti toliko složena da ju uz odgovarajuće alate ne može predstaviti učeniku. Fraktalna analiza se dotiče bitnog problema u proučavanju prirode. Prirodne pojave su kompleksne i za njihovo razumijevanja su potrebna višegodišnja znanstvena istraživanja. Nedostaju metode kojima bi se one mogle na brži način kvantitativno opisati. Metoda određivanja fraktalne dimenzije je jedna od perspektivnih tehnika ako se na odgovarajući način poveže sa geološkim, hidrološkim i fizikalnim procesima u prirodi.

Dimenzije u slučaju pravilnih geometrijskih likova poput linija, kvadrata i kocki su cjelobrojne: 1, 2, 3. Međutim, u prirodi postoje strukture (fraktali) koje nemaju cjelobrojnu prostornu dimenziju. Tehnika prezentirana u okviru ovog rada usmjerena je ka određivanju te dimenzije te analiza odabrane špilje to potvrđuje. Za Jamski sustav Kita Gaćešina-Draženova puhaljka određena je fraktalna dimenzija oko 2,3 – 2,4 što potvrđuje da je mreža špiljskih kanala fraktalna. To je značajan rezultat koji zahtijeva daljnju interpretaciju u okviru geoloških studija.

Literatura

1. Andreychouk, V., Błachowicz, T., Domino, K., Fractal dimensions of cave for exemplary gypsum cave-mazes of Western Ukraine. Landform Analysis, Vol. 22, 2003, 3–8

2. Barišić, Teo; Bakšić, Darko; Paar, Dalibor. Cave system of Kita Gaćešina -Draženova Puhaljka, the longest cave in the Dinaric karst. 16th International Congress of Speleology, Proceedings, Vol 3. / Filippi, Michal ; Bosak, Pavel (ur.). Brno : Czech Speleological Society and UIS, 2013. 17-20.

3. Barun, P., Fraktali i fraktalna dimenzija, diplomski rad, Fizički odsjek, Zagreb, 2001.

4. Gleick, J., Kaos – rađanje nove znanosti prijevod s engleskog, Izvori, Zagreb, 1996.

5. Josipović, M., Određivanje fraktalne dimenzije jadranske obale, diplomski rad, PMF, Fizički odsjek, Zagreb, 1996.

6. Paar, V. , Udžbenik za 4.razred gimnazije, Školska knjiga, Zagreb, 2006.

7. Radić, A., Fraktalna analiza u kršu, diplomski rad, PMF, Fizički odsjek, 2014.

8. Turcotte, D. L., Fractals and chaos in geology and geophysics, 2ed, Cambridge University Press, New York, 1999.

9. Kurikulum, dostupno na http://mzos.hr/datoteke/10-Predmetni_ kurikulum-Fizika.pdf (lipanj 2016.)

10. Kurikulum, dostupno na http://mzos.hr/datoteke/10Predmetni_ kurikulum-Matematika.pdf (lipanj 2016.)

11. http://www.math.sunysb.edu/ scott/Book331/Fractal_Dimension.html (lipanj 2016.)

12. Fraktali, dostupno na http://matematika.fkit.hr/novo/izborni/referati/ dobrinic_joskic_brdar_fraktali.pdf (srpanj 2016.)

13. Fraktalna dimenzija, dostupno na http://e.math.hr/frakdim/index.html (srpanj 2016.)

14. Koch, dostupno na http://e.math.hr/math_e_article/br16/ plavcic_skrtic_pavrlisak/koch (srpanj 2016.)

15. Thomas, J.W.: A Review of Research on Procjet-Based Learning, dostupno na http://www.bobpearlman.org/BestPractices/PBL_Research.pdf (srpanj 2016.)

16. Paar, N., Numeričke metode i matematičko modeliranje, dostupno na http://www.phy.pmf.unizg.hr/ npaar/teaching/nmmm1.pdf (srpanj 2016)

17. Vanderbilt, dostupno na http://www.vanderbilt.edu/AnS/psychology/ cogsci/chaos/workshop/ Fractals.html (srpanj 2016.)

18. http://kolegij.fizika.unios.hr/run/files/2011/02/Primjena-racunala-uobrazovanju2.pdf (srpanj 2016.)

19. http://jwilson.coe.uga.edu/emat6680/parsons/mvp6690/essay1/ sierpinski.html (srpanj 2016.)

20. http://www.physik.tu-dresden.de/ baecker/ papers/python.pdf (rujan 2016.)

21. https://www.python.org/ (rujan 2016.)

Prilog: Program za proračun fraktalne dimenzije trodimenzionalnog skupa podataka

```
# 3D FRAKTALNA DIMENZIJA { Python 2.7
import matplotlib.pyplot as plt
import os
import math
import numpy as np
from scipy.stats import linregress
from decimal import *
import re
filename = 'd:/program/kita8.xyz'  # odabir datoteke s podacima za analizu
match_number = re.compile('-?\ *[0-9]+\.?[0-9]*(?:[Ee]\ *-?\ *[0-9]+)?')
kx = []
ky = []
kz = []
f = open(filename)
num_lines = f.read().count('\n')
f.close()
print 'Broj tocaka u datoteci:', num_lines

i=0
imax=input('Koliko tocaka obraditi: ')
f = open(filename)
for line in f:
#   print line
    floatline = [float(xnumber) for xnumber in re.findall(match_number, line)]
    kx.append(floatline[0])    # x koordinata
    ky.append(floatline[1])    # y koordinata
```

```
    kz.append(floatline[2])    # z koordinata
    i=i+1;
    if i==imax:
        break
f.close()

# pomak 3D strukture u 1. kvadrant; opcija je skalirati prije
konverzije u pixele
kxmin=kx[np.argmin(kx)]
kx[:] = [x1 - kxmin for x1 in kx]
kymin=ky[np.argmin(ky)]
ky[:] = [y1 - kymin for y1 in ky]
kzmin=kz[np.argmin(kz)]
kz[:] = [z1 - kzmin for z1 in kz]

# definiranje granica strukture
kxmin=kx[np.argmin(kx)]
kymin=ky[np.argmin(ky)]
kzmin=kz[np.argmin(kz)]
kxmax=kx[np.argmax(kx)]
kymax=ky[np.argmax(ky)]
kzmax=kz[np.argmax(kz)]

print 'granice x ', kxmin, kxmax
print 'granice y ', kymin, kymax
print 'granice z ', kzmin, kzmax
imgx = kxmax
imgy = kymax
imgz = kzmax

# proracun 3D fraktalne dimenzije uporabom Box counting metode
b=int(raw_input('Unesite pocetnu velicinu kockica u pikselima b:'))
f=int(raw_input('Unesite faktor skaliranja kockica f:'))
print 'Maksimalni broj tocaka za izracun je nmax=', int(math.log(imgx)/math.log
n=int(raw_input('Unesite broj tocaka za izracun n:'))
gx = [] # koordinate tocaka na x osi
gy = [] # koordinate tocaka na y osi
gz = [] # koordinate tocaka na z osi
```

```
kxyz = np.array(zip(kx,ky,kz))
nrpoints = len(kxyz)

# x kxyz[i,0]   i=0..num_lines
# y kxyz[i,1]
# z kxyz[i,2]

for ib in range(n):
    bs = b * f ** ib # velicina kockice u pikselima
    bnx = int(imgx / bs) # broj kockica u x smjeru slike
    bny = int(imgy / bs) # broj kockica u y smjeru slike
    bnz = int(imgz / bs) # broj kockica u z smjeru slike
    print 'Broj kockica x, y, z', bnx, bny, bnz
    if bnx<2 or bny<2 or bny<2:
        break
    ix=[]
    iy=[]
    iz=[]
    ikx=0
    boxCount = 0
    for bx in range(bnx):
        bxmin=bs*bx
        bxmax=bs*bx+bs
        for ib in range(nrpoints):
            kxyz_x=kxyz[ib,0]
            if (bxmin<=kxyz_x) and (kxyz_x<bxmax):
                ikx += 1 # generiram listu tocaka sa zadanim x uvjetom
                ix.append(ib)
        if ikx == 1:
            boxCount += 1
        else:   # eliminacija tocaka koje se nalaze u istim kockicama
            iydel=[]
            for i2 in range(ikx):
                for i3 in range (i2+1,ikx):
                    kxyz_y1=kxyz[ix[i2],1]
                    kxyz_y2=kxyz[ix[i3],1]
                    if abs(kxyz_y1-kxyz_y2)<bs:
                        kxyz_z1=kxyz[ix[i2],2]
                        kxyz_z2=kxyz[ix[i3],2]
```

```
                        if abs(kxyz_z1-kxyz_z2)<bs:
                            iydel.append(i3)

            for i2 in range(ikx):
                if i2 not in iydel:
                    boxCount += 1

    print "bnx=",bnx," bny=",bny," bs=",bs
    print "ib=",ib,"  boxCount=",boxCount
    if boxCount>0:
        gx.append(math.log(1.0 / bs))
        gy.append(math.log(boxCount))

# linearna regresija

slope, intercept, r_value, p_value, std_err=linregress(gx,gy)
getcontext().prec = 1 #postavlja preciznost pogreske na 1 znamenku
Derr=Decimal(std_err)
a=Derr
i=0
while (a%1 != 0):  #nalazi broj znamenki u pogresci
    a=a*10
    i=i+1
D=round(slope, i) #zaokruzuje dimenziju na broj znamenki kao kod pogreske
postotak=(float(Derr)/D)*100  #daje pogresku u postotku
print "Procijenjena fraktalna dimenzija: " + str(slope)
print "Pogreska: ", +std_err
print ""
print 'D +- Derr: %s +- %s' % (str(D), round(Derr, i))
print 'Pogreska u postotku:' + str(round(postotak, i)) + '%'

gx=np.array(gx)
gy=np.array(gy)
points=np.linspace(gx.min(), gx.max(), n)
plt.plot(gx, gy, '.', color='red')
plt.plot(points, slope*points+intercept)
plt.savefig("rezultat.jpg")
plt.show()
```

Sažetak

Ovaj rad pokazuje mogućnosti osmišljavanja projektne nastave koja povezuje znanja iz različitih područja kao što su matematika, fizika i programiranja s prirodnim pojavama. Uloga projektne nastave je upravo da se učenik stavi u središte nastavnog procesa te da ga se zainteresira za osnovnoškolsku i srednjoškolsku nastavu primjenjujući nova znanja na problemima iz svakodnevnog života. Odabrani problem u slučaju ovog rada je specifičan, to su prirodne pojave u kršu – špilje – za koje smo postavili pitanje imaju li prirodnu fraktalnost. Fraktalna dimenzija kao rezultat analize prirodne pojave je kvantitativni podatak koji može imati veliku važnost, ne samo radi utvrđivanja fraktalnosti (a time i kvantitativna mjera razine kompleksnosti) dane pojave već ona može biti indikator različitih geoloških, hidroloških i fizikalnih procesa koji su utjecali na razvoj špilja. Metoda se može primijeniti i na druge pojave u kršu poput mreže vodenih tokova, distribucije ponikvi itd. Nešto od toga je u svijetu publicirano u znanstvenoj literaturi pa uvođenjem učenika u tu tematiku zapravo ih uvodimo u aktualnu znanstvenu metodu, odnosno učimo ih neke od temeljnih koncepata znanstvenog istraživanja: izbor metode, izbor pojave koja se proučava, analiza i interpretacija. U tom kontekstu ovakav projektni zadatak može imati i širi okvir. Uloga ovog rada je i da nastavniku pokaže da znanstvena analiza kompleksnih pojava u prirodi ne mora biti toliko složena da ju uz odgovarajuće alate ne može predstaviti učeniku.

Summary

This work shows the possibilities for designing project-based learning that connects knowledge of mathematics, physics and programming with natural phenomena. The role of project-based learning is to increase the student interest in the theme by applying new knowledge to real problems. Selected problem in the this work is specific, these are natural phenomena in karst - caves - which we asked the question about their natural fractality. Fractal dimension as a result of the analysis of natural phenomena is quantitative information that can have great importance, not only to determine fractality (and thus a quantitative measure of the level of complexity) of a given phenomena but it can be an indicator of various geological, hydrological and physical processes that influenced the development of the cave . The method can be applied to other phenomena in the karst. Some studies (not related to Croatia) were published in the scientific literature. Introducing students to this topic actually introduce them into the current scientific method, or teach them some of the basic concepts of scientific research: methods of investigation, selection of phenomena under study, analysis and interpretation. In this context, this project task can have a broader framework. The role of this work is that the teachers can understand that scientific analysis of complex phenomena in nature does not have to be so complex and it can be with the proper tools presented to students.

Životopis

Andreja Mikuš rođena je 02.05.1988. u Zagrebu. Od 1995. do 2003. godine pohađa Osnovnu školu „Marija Bistrica“. Nakon završetka osnovne škole, upisuje srednju školu „Gimnazija Sesvete“, općeg usmjerenja gdje maturira 2007. godine. Iste godine upisuje Integrirani preddiplomski i diplomski studij Matematika i fizika, smjer: nastavnički, na Prirodoslovno-matematičkom fakultetu Sveučilišta u Zagrebu.