Stohastička procjena tortuoziteta digitalnog uzorka stijene

Šimić, Robert

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mining, Geology and Petroleum Engineering / Sveučilište u Zagrebu, Rudarsko-geološko-naftni fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:169:554672

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-04-19**

Repository / Repozitorij:

Faculty of Mining, Geology and Petroleum Engineering Repository, University of Zagreb

SVEUČILIŠTE U ZAGREBU

RUDARSKO-GEOLOŠKO-NAFTNI FAKULTET

Diplomski studij naftnog rudarstva

STOHASTIČKA PROCJENA TORTUOZITETA DIGITALNOG UZORKA STIJENE

Diplomski rad

Robert Šimić

N377

Zagreb, 2022.

Sveučilište u Zagrebu
Rudarsko-geološko-naftni fakultet

Diplomski rad

STOHASTIČKA PROCJENA TORTUOZITETA DIGITALNOG UZORKA STIJENE

ROBERT ŠIMIĆ

Diplomski rad izrađen: Sveučilište u Zagrebu
Rudarsko-geološko-naftni fakultet
Zavod za naftno-plinsko inženjerstvo i energetiku
Pierottijeva 6, 10000 Zagreb

Sažetak

Izrađen je računalni kod, koji računa tortuozitet, tj. koeficijent zakrivljenosti pora iz digitalnog uzorka segmentirane i binarizirane CT slike stijene, za koju su, na temelju prethodno izrađenih modula (*in-house* kód RGN fakulteta), određeni povezani putevi kroz porni sustav. Algoritam se služi stohastičkim pristupom tj. koristi modificiranu metodu slučajnog hoda te se za skretanje putanje unutar pornih kanala koristi normalna raspodjela za koju je moguće definirati različite standardne devijacije. Uz standardnu devijaciju veću od 1,0 ustanovljeno je da rezultati postaju sličniji i da se poklapaju s nekim algoritmima drugih autora, koji su također testirani na istom digitalnom uzorku.

Ključne riječi: digitalna jezgra, digitalna fizika stijena, analiza mikro-tomografskih slika, koeficijent zakrivljenosti pora, tortuozitet, slučajni hod, stohastička simulacija.

Diplomski rad sadrži: 55 stranica, 30 slika, 2 tablice, 34 reference i prilog

Diplomski rad pohranjen: Knjižnica Rudarsko-geološko-naftnog fakulteta
Pierottijeva 6, 10000 Zagreb

Voditelj: Dr. sc. Domagoj Vulin, redoviti profesor RGNF-a

Pomoć pri izradi: Marko Gaćina, mag. ing. naft. rud.

Ocjenjivači: Dr. sc. Domagoj Vulin, redoviti profesor RGNF-a
Dr. sc. Vladislav Brkić, izvanredni profesor RGNF-a
Dr. sc. Borivoje Pašić, izvanredni profesor RGNF-a

Datum obrane: 15. rujna 2022., Rudarsko-geološko-naftni fakultet, Sveučilište u Zagrebu

University of Zagreb
Faculty of Mining, Geology and Petroleum Engineering

Master's Thesis

STOCHASTIC PREDICTION OF TORTUOSITY OF DIGITAL ROCK SAMPLE

ROBERT ŠIMIĆ

Thesis completed at: University of Zagreb
Faculty of Mining, Geology and Petroleum Engineering
Department of Petroleum and Gas Engineering and Energy
Pierottijeva 6, 10000 Zagreb

Abstract

Tortuosity, or the curvature coefficient of pores, was computed using a digital sample of a segmented and binarized CT image of the rock and interconnected pathways across the pore system, which were found using previously created modules (in-house code of the RGN Faculty). The program employs a stochastic approach, i.e., a modified random walk technique with a normal distribution, for which different standard deviations can be chosen to simulate the drift from the shortest path inside the pore channels. When the standard deviation exceeds 1.0, the outputs become more comparable and correspond with several algorithms developed by other authors that have been evaluated using the same digital sample.

Key words: digital core, digital rock physics, analysis of micro-tomographic images, pore curvature coefficient, tortuosity, random walk, stochastic simulation.

Thesis consists of: 55 pages, 30 figures, 2 tables, 34 references and one appendix

Thesis archived at: The Library of the Faculty of Mining, Geology and Petrroleum Engineering
Pierottijeva 6, 10000 Zagreb

Supervisor: Full Professor Domagoj Vulin, PhD

Co-supervisor: Marko Gaćina, mag. ing. petrol.

Reviewers: Full Professor Domagoj Vulin, PhD
Associate Professor Vladislav Brkić, PhD
Associate Professor Borivoje Pašić, PhD

Date of defense: September 15, 2022, Faculty of Mining, Geology and Petroleum Engineering, University of Zagreb

SADRŽAJ

POPIS SLIKA ..... **I**

POPIS TABLICA ..... **III**

POPIS KORIŠTENIH OZNAKA I JEDINICA ..... **IV**

POPIS KORIŠTENIH KRATICA ..... **V**

1. UVOD **1**

2. TEORIJSKA POZADINA **4**

2.1 Geometrijski tortuozitet **4**

2.1.1 Direktna metoda traženja najkraćeg puta 5

2.1.2 Metoda traženja najkraćeg puta u skeletu pore *6*

2.1.3 Fast marching metoda *8*

2.1.4 Metoda težišta pore *9*

2.2 Fizički tortuozitet **9**

2.2.1. Hidraulički tortuozitet *10*

2.2.2. Električni tortuozitet *11*

2.2.3. Difuzni tortuozitet *12*

2.2.4. Toplinski tortuozitet *14*

2.3 Stohastičke metode u analizi digitalnog uzorka stijene **15**

3. ALGORITAM *SLUČAJNOG HODA* **19**

4. REZULTATI **24**

5. DISKUSIJA REZULTATA **33**

6. ZAKLJUČAK **35**

7. LITERATURA **37**

8. PRILOG **40**

POPIS SLIKA

Slika 1-1. Karakterizacija pornog prostora pri različitim rezolucijama pojedinih tehnika snimanja 2

Slika 2-1. Shema pornog kanala duljine L_{eff} i ravne linije L između krajnjih točaka 5

Slika 2-2. 3D mikrostruktura s pornim prostorom označenim žutom bojom (a), pretvorba 3D podataka u 2D graf (b), najkraći porni kanali pronađeni Dijkstrinim algoritmom (c) 6

Slika 2-3. Najkraći pronađeni put Dijkstrinim algoritmom u uskoj pori (a) i širokom pornom kanalu (b) 7

Slika 2-4. Porni prostor poroznog medija (a), izvađeni skelet pora (b), pronađeni najkraći put između gornje i donje plohe (c) 7

Slika 2-5. 3D mikrostruktura (a), porni prostor poroznog medija (b), karta udaljenosti izračunata *fast marching* metodom (c) 8

Slika 2-6. 3D digitalna snimka poroznog medija (a), udaljenosti L_i između težišta slojeva (b), efektivna porna putanja određena metodom težišta pore (c) 9

Slika 2-7. 3D porna mikrostruktura (a), rubni uvjeti za simulaciju (b), polje ustaljenih brzina fluida unutar poroznog medija (c) 11

Slika 2-8. 3D porozni medij (a), rubni uvjeti za simulaciju električne vodljivosti (b), gustoća električnog toka unutar poroznog medija (c) 12

Slika 2-9. Shema rubnih uvjeta za simulaciju molekularne difuzije u poroznom mediju (a), gustoća difuznog toka u poroznom mediju (b), gustoća difuznog toka u homogenom mediju (c) 14

Slika 2-10. 3D porozni medij (a), rubni uvjeti za simulaciju prijenosa topline (b), prijenos topline unutar poroznog medija (c) 15

Slika 2-11. Histogram normalne raspodjele 16

Slika 2-12. Uzorak od 100 slučajno generiranih brojeva prema normalnoj raspodjeli (centralna vrijednost $\boldsymbol{\mu = 0}$, raspršenost, tj. standardna devijacija $\boldsymbol{\sigma = 10}$, broj stupaca u histogramu 10), s parametrima dobivenim testiranjem raspodjele ($\boldsymbol{\mu}$ *i* $\boldsymbol{\sigma}$ prikazani u samom dijagramu) 16

Slika 2-13. Uzorak od 5000 slučajno generiranih brojeva prema normalnoj raspodjeli ($\boldsymbol{\mu = 0, \sigma = 10}$, broj stupaca u histogramu 10), s parametrima dobivenim testiranjem raspodjele. 17

Slika 2-14. Uzorak od istih kao u prethodnoj slici 5000 slučajno generiranih brojeva, broj stupaca u histogramu 50. 17

Slika 3-1. Shematski prikaz mogućih pojavljivanja pora..... 20

Slika 3-2. Shema određivanja pomaka u sloju: stanje oko točke u sljedećem sloju (a), identificiranje mogućih pomaka (b), dodjeljivanje postotne šanse mogućim pomacima (c), konačni odabir pomaka (d)..... 21

Slika 3-3. Dijagram tijeka algoritma..... 23

Slika 4-1. Model pornog prostora promatranog uzorka..... 24

Slika 4-2. Prikaz postotnih šansi za pomak za devijacije 1,0 i 0,1 25

Slika 4-3. Putanje za devijaciju 1,0 (plava) i devijaciju 0,1 (crvena) iz nasumično odabrane početne točke..... 26

Slika 4-4. Pronađene putanje za sve točke iz prvog sloja, lijevo za devijaciju 1,0, desno za 0,1 27

Slika 4-5. Pronađene putanje za sve točke iz prvog sloja za uzorak rotiran 90° oko x osi, lijevo za devijaciju 1,0, desno za 0,1..... 28

Slika 4-6. Pronađene putanje za sve točke iz prvog sloja za uzorak rotiran 90° oko y osi, lijevo za devijaciju 1,0, desno za 0,1..... 28

Slika 4-7. Histogram tortuoziteta za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)... 29

Slika 4-8. Histogram tortuoziteta uzorka rotiranog oko x osi za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)..... 30

Slika 4-9. Histogram tortuoziteta uzorka rotiranog oko y osi za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)..... 30

Slika 4-10. Raspodjela tortuoziteta za devijacije 0,1, 1,0 i 10,0..... 30

Slika 4-11. Podaci o izračunatim tortuozitetima za pojedini isječak za devijaciju 0,1..... 32

Slika 4-12. Podaci o izračunatim tortuozitetima za pojedini isječak za devijaciju 1,0..... 32

POPIS TABLICA

Tablica 4-1. Izračunati tortuozitet za uzorke veličine 50, 100, 200 i 300 voksela 31

Tablica 5-1. Usporedni rezultati tortuoziteta dobivenih različitim metodama 33

POPIS KORIŠTENIH OZNAKA I JEDINICA

Oznaka	Jedinica	Opis
D_0	cm^2/s	difuzijski koeficijent
D_{eff}	cm^2/s	efektivni difuzijski koeficijent
k	m^2	propusnost
L	m	duljina
L_{eff}	m	efektivna duljina
S	m^{-1}	specifična površina
x_i	-	x koordinata
y_i	-	y koordinata
z_i	-	z koordinata
β	-	faktor oblika
λ_0	W/(m·K)	toplinska vodljivost
λ_{eff}	W/(m·K)	efektivna toplinska vodljivost
μ_x	-	devijacija po x osi
μ_y	-	devijacija po y osi
σ_0	S/m	električna vodljivost
σ_{eff}	S/m	efektivna električna vodljivost
σ_x	-	očekivanje po x osi
σ_y	-	očekivanje po y osi
τ	-	tortuozitet
τ_d	-	difuzni tortuozitet
τ_e	-	električni tortuozitet
τ_g	-	geometrijski tortuozitet
τ_h	-	hidraulički tortuozitet
τ_t	-	toplinski tortuozitet
ϕ	-	poroznost

POPIS KORIŠTENIH KRATICA

AFM – Mikroskopija atomskih sila (engl. *Atomic Force Microscopy*)

CDF – Funkcija kumulativne distribucije (engl. *Cumulative Distribution Function*)

DSPSM – Direktna metoda traženja najkraćeg puta (engl. *Direct Shortest Path Searching Method*)

FDM – Metoda konačnih razlika (engl. *Finite Difference Method*)

FEM – Metoda konačnih elemenata (engl. *Finite Elements Method*)

FIB-SEM – Skenirajuća elektronska mikroskopija koja koristi fokusiranu ionsku zraku (engl. *Focused Ion Beam Scanning Electron Microscope*)

FMM – *fast marching* metoda

FVM – Metoda konačnih volumena (engl. *Finite Volume Method*)

LBM – Boltzmannova metoda rešetke (engl. *Lattice Boltzmann Method*)

Micro-CT – Mikro-kompjuterska tomografija (engl. *Micro-Computed Tomography)*

MRI – Magnetska rezonancija (engl. *Magnetic Resonance Imaging)*

PDF – Funkcija gustoće vjerojatnosti (engl. *Probability Density Function*)

REV – Reprezentativni elementarni volumen

RVS – Slučajni uzorak (engl. *Random Variable Sample*)

RW – Slučajni hod (engl. *Random Walk*)

RWM – Metoda slučajnog hoda (engl. *Random Walk Method*)

SAXS – Raspršenje rendgenskih zraka pod malim kutom (engl. *Small Angle X-ray Scattering)*

SEM – Skenirajuća elektronska mikroskopija (engl. *Scanning Electron Microscope*)

SSPSM – Skeleton metoda traženja najkraćeg puta (engl. *Skeleton Shortest Path Search Method*)

TEM – Transmisijska elektronska mikroskopija (engl. *Transmission electron microscopy*)

1. UVOD

Tortuozitet (*koeficijent zakrivljenosti pora*) je bitan koncept u mnogim znanstvenim područjima. U geoznanosti, tortuozitet je blisko povezan s prodiranjem fluida, molekularnom difuzijom, električnom vodljivošću, faktorom otpornosti formacije i drugo. Kod skladištenja energije, tortuozitet je ključan faktor kod dizajniranja elektroda kako bi se maksimizirao skladišni kapacitet baterije (Kehrwald et al., 2011). U slučaju kemijskih katalizatora, tortuozitet ima veliki utjecaj na efikasnost odvijanja reakcije (Kolitcheff et al., 2017). Kod tretiranja vode, tortuozitet mikrostrukture membrane značajno utječe na efikasnost destilacije (Karanikola et al., 2017).

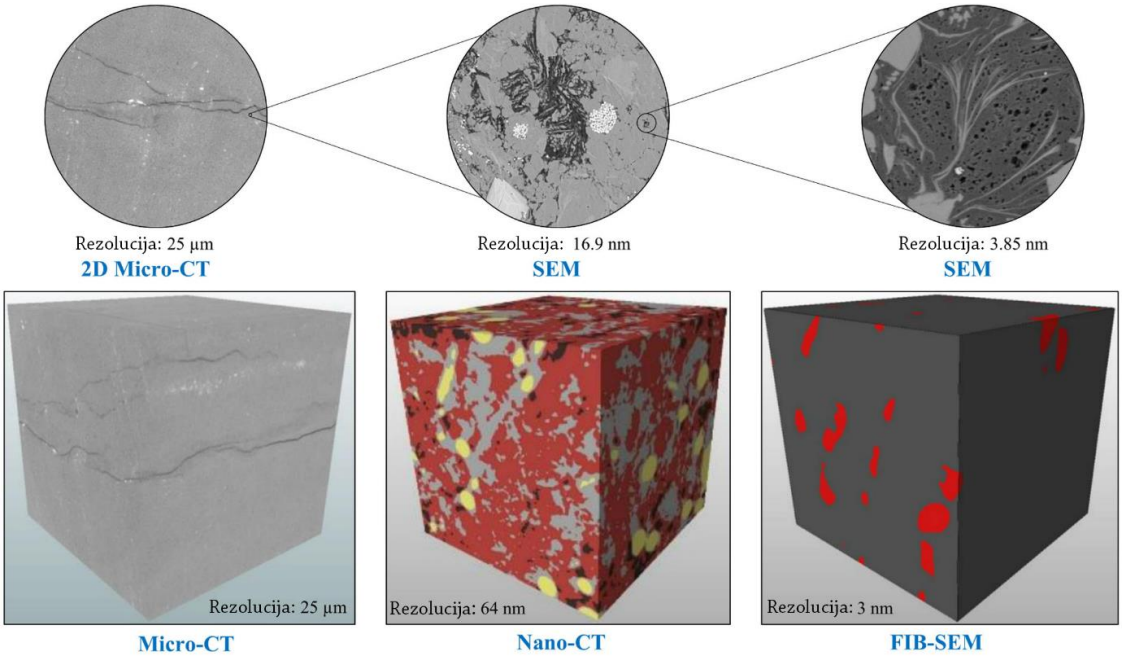
Unatoč širokoj prisutnosti u mnogim područjima, tortuozitet nema konkretnu definiciju. Postoje mnogi pokušaji da se odredi tortuozitet određenog medija, no metode i rezultati se razlikuju ovisno o tipu tortuoziteta (Da Silva et al., 2022).

Postoje dva široko upotrebljavana eksperimentalna pristupa određivanja tortuoziteta. Prvi pristup je test električne vodljivosti, gdje se testni, porozni medij zasiti elektrolitom te tortuozitet odredi mjerenjem efektivne vodljivosti. Drugi pristup je mjerenje koeficijenta difuzije fluida koji se difuzijom širi kroz uzorak pornog medija, gdje se tortuozitet određuje na temelju izmjerene efektivne difuzije. U usporedbi s testom električne vodljivosti, test difuzije je mnogo sporiji te je sam test osjetljiv na odabranu vrstu fluida i temperaturu pri kojoj je rađen. Često se vrijednosti tortuoziteta dobivene ovim testovima razlikuju, pa je na temelju toga moguće definirati nekoliko vrsta tortuoziteta koji se razlikuju u fizikalnom procesu i metodi mjerenja.

Karakteristike mikrostrukture, poput oblika (morfologije), veličine i prostorne distribucije zrna i pora utječu na tortuozitet poroznog medija. Porozni medij sadrži veliki broj geometrijski nepravilnih pora, koje mogu biti povezane, izolirane ili slijepe. Tipično je kontakt stijena-pora hrapav, što, uz kemijska svojstva fluida i stijene ima značajan učinak otpora na protok fluida i stijene jako sličnih morfologija pora mogu imati različite razine korelacije propusnosti i tortuoziteta. Zbog toga je gotovo nemoguće napraviti jedinstveni matematički model koji može točno izračunati tortuozitet poroznog medija. Vrijednosti tortuoziteta izračunatih matematičkim modelima se znatno razlikuju od tortuoziteta

dobivenih eksperimentalnim mjerenjima ili numeričkim simulacijama, te su u većini slučajeva pogrešne.

Moderne mikroskopske tehnike snimanja pružaju slike visoke kvalitete tj. rezolucije i karakteriziraju porni prostor u rezolucijama od razine milimetara do nekoliko nanometara. Tu spadaju mikro-kompjuterska tomografija (engl. *Micro-Computed Tomography*, Micro-CT), skenirajuća elektronska mikroskopija (engl. *Scanning Electron Microscope*, SEM) kao i ona koja koristi fokusiranu ionsku zraku (engl. *Focused Ion Beam Scanning Electron Microscope*, FIB-SEM), transmisijska elektronska mikroskopija (engl. *Transmission electron microscopy*, TEM), mikroskopija atomskih sila (engl. *Atomic Force Microscopy*, AFM), magnetska rezonancija (engl. *Magnetic Resonance Imaging*, MRI) i raspršenje rendgenskih zraka pod malim kutom (engl. *Small Angle X-ray Scattering*, SAXS). Iako se postupci razlikuju, dobivene digitalne slike su usporedive pri istoj rezoluciji. Mikroskopske slike se obrađuju i spremaju kao pikseli ili vokseli u nijansama sive boje, te se daljnjom obradom mogu pretvoriti u binarni oblik, gdje je jasna razlika pornog prostora od matriksa stijene. Segmentirana digitalna mikrostruktura dovoljne veličine i rezolucije može se koristiti kao reprezentativni elementarni volumen (REV) za daljnje analize i simulacije. Na slici 1-1. prikazane su rezolucije pojedinih tehnika snimanja.



Slika 1-1. Karakterizacija pornog prostora pri različitim rezolucijama pojedinih tehnika snimanja (Jinlong et al., 2020)

Posljednjih godina, s povećanom dostupnošću digitalnih snimki mikrostruktura, efekti kompleksnih mikrostruktura na makroskopska protočna svojstva privukli su značajnu pažnju. Zbog velike efikasnosti, evaluacija tortuoziteta kroz analizu digitalne snimke postala je preferirana metoda. Za opis zakrivljenosti puta fluida u poroznom mediju koristi se geometrijski tortuozitet, koji predstavlja omjer duljine efektivnog puta fluida i dužine ravne linije u smjeru toka. Geometrijski tortuozitet se najčešće računa tako da se pronađe najkraći put kroz porni prostor uzorka i pretpostavi da on predstavlja stvarni put fluida. Neke od najpopularnijih metoda za računanje geometrijskog tortuoziteta su direktna metoda traženja najkraćeg puta (engl. *Direct Shortest Path Searching Method*, DSPSM), skeleton metoda traženja najkraćeg puta (engl. *Skeleton Shortest Path Search Method*, SSPSM), *fast marching* metoda (FMM) i metoda centra pora. Treba napomenuti da je geometrijski tortuozitet karakteristika mikrostrukture, neovisan o specifičnom procesu toka fluida u poroznom mediju. Suženja i uska grla pornih kanala mogu značajno utjecati na tok fluida. Naime, za izračun geometrijskog tortuoziteta uzima se u obzir samo duljina puta, ne i utjecaj poprečnog presjeka pornog kanala na protok. Također, putanja određena analizom digitalne snimke može se značajno razlikovati od stvarnog toka fluida u poroznom mediju. Ova ograničenja mogu objasniti zašto je geometrijskim tortuozitetom teško točno modelirati propusnost, difuznost i vodljivost.

Motivacija ovog rada je izraditi računalni kod koji će stohastički, metodom slučajne putanje (engl. *random walk*) pronaći statističke raspodjele mogućih putanja čestice fluida od početnog sloja do zadnjeg sloja razmatranog digitalnog uzorka.

Hipoteza rada glasi: razmatranjem različitih distribucija disperzije čestica fluida (povezivo sa stupnjem turbulencije čestica u protjecanju), moguće je odrediti tortuozitet pora i elementarni volumen koji pokazuje da je procjena konzistentna za snimke uzoraka različite veličine.

Sami računalni kod, zbog svoje jednostavnosti, zamijenit će potrebu za skupim softverom i snažnim računalima i poslužiti u boljoj evaluaciji propusnosti digitalnih uzoraka.

2. TEORIJSKA POZADINA

Visoko kvalitetne snimke mikrostrukture omogućuju izradu numeričkih simulacija toka fluida, kako bi se tortuozitet mogao procijeniti iz simulacija prodiranja fluida, električne vodljivosti, molekularne difuzije i prijenosa topline. Ovakvi tortuoziteti na bazi toka fluida (hidraulički, električni, difuzni te toplinski tortuozitet) mogu se objediniti pod nazivom fizički tortuoziteti. Neke od popularnijih metoda numeričkih simulacija su Boltzmannova metoda rešetke (engl. *Lattice Boltzmann Method*, LBM), metoda konačnih elemenata (engl. *Finite Elements Method*, FEM), metoda konačnih volumena (engl. *Finite Volumes Method,* FVM), metoda konačnih razlika (engl. *Finite Difference Method*, FDM) i metoda slučajnog hoda (engl. *Random Walk Method,* RWM) (Nemati et al., 2020).

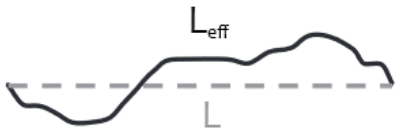
U usporedbi s geometrijskim tortuozitetom, fizički tortuozitet je povezan i s mikrostrukturom i s protokom fluida unutar pora. Za određeni protok fluida, izračunati fizički tortuozitet predstavlja otpor mikrostrukture na protok fluida. Razvijeni su mnogi računalni programi i programski moduli za analizu mikrostrukture. Neki od njih su Tort3D, TauFactor, Pytrax, Avizo, Simpleware i GeoDict. Iako svi navedeni programi mogu pružiti podatke o morfološkim karakteristikama ili fizikalnim svojstvima poroznih medija, numeričke metode korištene u proračunu se razlikuju, čime se i vrijednosti izračunatih tortuoziteta razlikuju od programa do programa.

2.1 Geometrijski tortuozitet

Geometrijski tortuozitet je definiran kao omjer najkraćeg puta povezanih točaka i udaljenosti ravne linije između početne i krajnje točke. Drugim riječima, geometrijski tortuozitet je omjer zakrivljenog puta protoka (L_{eff}) i ravne linije u smjeru protoka (L) (Clennell, 1997). Jednadžba za računanje geometrijskog tortuoziteta glasi:

$$\tau = \frac{L_{eff}}{L} \tag{2-1}$$

Na slici 2-1. prikazana je shema pornog kanala duljine L_{eff} i direktne, ravne linije L između krajnjih točaka.



Slika 2-1. Shema pornog kanala duljine L_{eff} i ravne linije L između krajnjih točaka (Shanti et al., 2014)

Geometrijski tortuozitet je karakteristika mikrostrukture pora i određena je isključivo geometrijskim i morfološkim obilježjima poroznog medija. Određuje se na temelju digitalne snimke uzorka, koja je pretvorena u binarni zapis. U nastavku su opisane neke od najčešćih metoda za računanje geometrijskog tortuoziteta.

2.1.1 Direktna metoda traženja najkraćeg puta

Metoda traženja direktnog najkraćeg puta traži najkraći porni kanal direktno na podacima koristeći neki od algoritama za traženje najkraćeg puta, poput Dijkstrinog algoritma, algoritma širenja valne fronte i A* algoritam (Cherkassky et al., 1996). U nastavku je dan kratak opis Dijkstrinog algoritma.

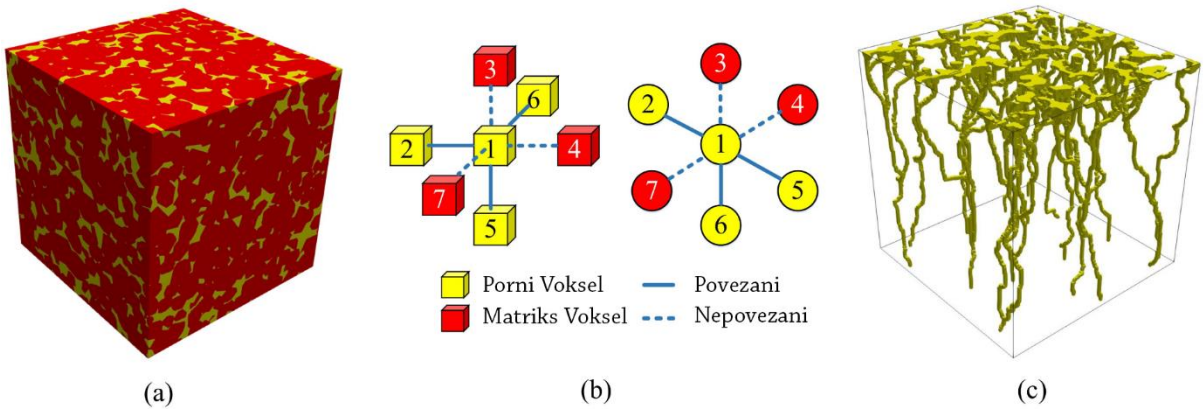
Najkraći porni kanal je definiran kao najkraća putanja koja počinje na površinskom vokselu na jednoj strani digitalne mikrostrukture i završava na površinskom vokselu na suprotnoj strani te se cijelo vrijeme nalazi unutar pornog prostora. Promjenom početnog ili krajnjeg voksela nastaje nova najkraća putanja, pa se geometrijski tortuozitet τ_g računa prema jednadžbi:

$$\tau_g = \langle \frac{L_{min}}{L} \rangle \qquad (2\text{-}2)$$

Gdje je L_{min} prosječni omjer najkraćeg pornog kanala, a L udaljenost ravne linije.

Algoritam prvo pretvara set podataka o 3D mikrostrukturi u 2D matrice susjedstva, kako bi se mikrostruktura mogla prikazati u obliku grafa. Svaki čvor u grafu predstavlja jedan voksel, a veze između čvorova predstavljaju veze između voksela. Euklidska udaljenost između dva voksela pora u mikrostrukturi se koristi kao težinska veza između odgovarajućih

čvorova na grafu. Zatim Dijkstrin algoritam traži najkraći put od zadanog početnog čvora do završnog čvora na suprotnoj strani uzorka. Put se traži za sve voksele koji predstavljaju porni prostor na početnoj površini te se tortuozitet računa prema jednadžbi (2-2). Na slici 2-2. prikazana je 3D mikrostruktura gdje je porni prostor označen žutom bojom (a), pretvaranje 3D podataka u 2D graf (b) i najkraći pronađeni put Dijkstrinim algoritmom između gornje i donje plohe (c).

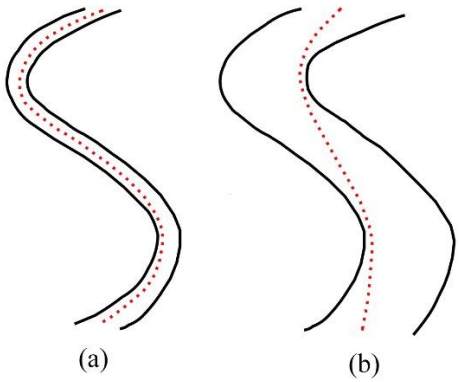


(a) (b) (c)

Slika 2-2. 3D mikrostruktura s pornim prostorom označenim žutom bojom (a), pretvorba 3D podataka u 2D graf (b), najkraći porni kanali pronađeni Dijkstrinim algoritmom (c) (Jinlong et al., 2020)

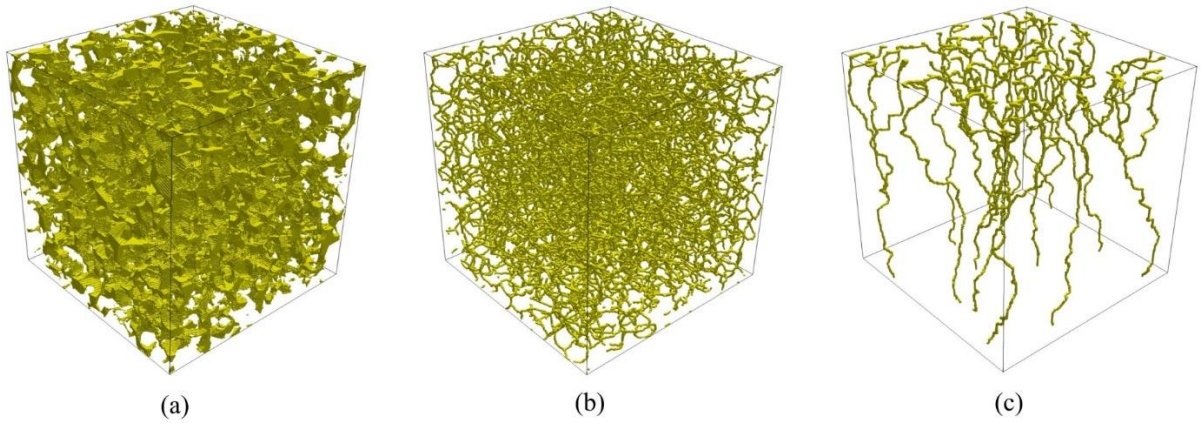
2.1.2 Metoda traženja najkraćeg puta u skeletu pore

Najkraći porni kanal određen direktnom metodom traženja može značajno odstupati od središta porne mreže, osobito u medijima s velikom poroznošću. Kao što je prikazano na slici 2-3., najkraći put pronađen Dijkstrinim algoritmom u širokom kanalu odstupa od centralne osi, dok je to odstupanje za uže kanaliće mnogo manje. Zbog toga direktna metoda može podcijeniti vrijednost geometrijskog tortuoziteta.



Slika 2-3. Najkraći pronađeni put Dijkstrinim algoritmom u uskoj pori (a) i širokom pornom kanalu (b) (Jinlong, 2020)

Jedan od načina kako riješiti ovaj problem je koristeći metodu traženja najkraćeg puta u skeletu pore umjesto u pornom prostoru. Skelet pore predstavlja središnju os pornog prostora te zadržava većinu geometrijskih svojstava pornog prostora. Za izradu kostura pore koristi se homotopski algoritam za stanjivanje, te se na dobivenom kosturu koristi prethodno opisani Dijkstrin algoritam (Lindquist et al., 1996). Geometrijski tortuozitet se računa prema jednadžbi (2-2). Prednost ove metode je jednostavniji prikaz pornog prostora, zbog čega je sami algoritam brži. U načelu, tortuozitet izračunat ovom metodom je veći nego onaj izračunat direktnom metodom traženja, pogotovo u medijima s velikom poroznošću. Primjer je prikazan na slici 2-4**.**



Slika 2-4. Porni prostor poroznog medija (a), izvađeni skelet pora (b), pronađeni najkraći put između gornje i donje plohe (c) (Jinlong et al., 2020)

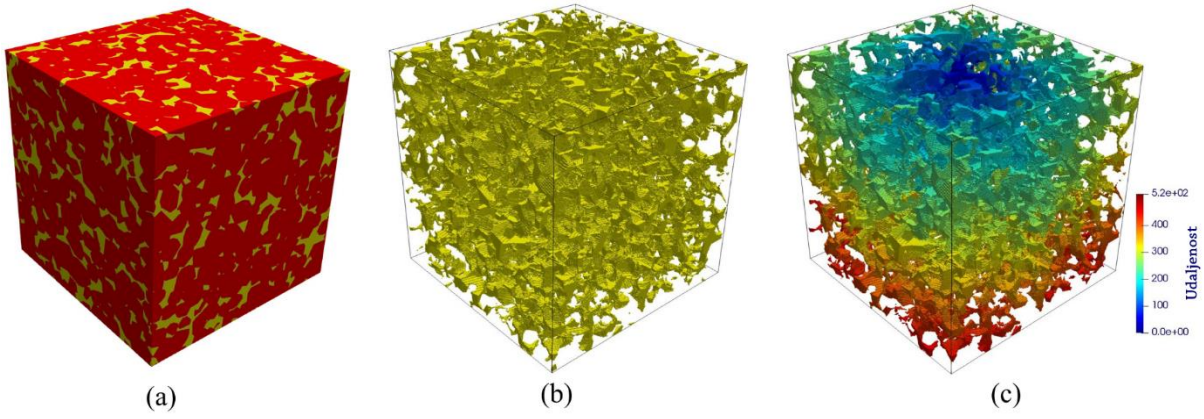
2.1.3 Fast marching metoda

Fast marching metoda je učinkovit algoritam za praćenje i modeliranje širenja valne fronte koja se kreće u određenom smjeru i znanom brzinom (Sethian, 1996). Metoda se temelji na eikonal jednadžbi koja glasi:

$$\|\nabla t(\boldsymbol{x})\| F(\boldsymbol{x}) = 1, F(\boldsymbol{x}) > 0 \qquad (2\text{-}3)$$

Gdje je $\|\nabla t(\boldsymbol{x})\|$ veličina vektora gradijenta funkcije dolaska $t(\mathbf{x})$, a $F(\mathbf{x})$ je brzina širenja valne fronte na lokaciji x. Pretpostavljanjem rješenja jednadžbe, može se izračunati vrijeme $t(\mathbf{x})$ potrebno da valna fronta dosegne određenu točku. Ako se postavi da je $F(\mathbf{x}) = 1$, vrijeme dolaska je jednako udaljenosti, čime se dobije udaljenost od ishodišne točke do svake točke unutar pornog prostora.

Vokseli na početnoj površini se odabiru kao ishodište valne fronte, a suprotna površina kao odredište. Duljina najkraćeg pornog kanala L_{min} dobije se tako da se postavi $F(\mathbf{x}) = 1$ za porni prostor i $F(\mathbf{x}) = 10^{-6}$ za matriks stijene, čime je brzina valne fronte mnogo veća u pornom prostoru nego u matriksu. Duljina ravne linije L se izračuna tako da se postavi $F(\mathbf{x}) = 1$ za porni prostor i matriks stijene, čime je brzina širenja valne fronte jednaka u oba medija. Geometrijski tortuozitet se računa po jednadžbi (2-2). Na slici 2-5. prikazana je 3D mikrostruktura, porni prostor te širenje valne fronte.



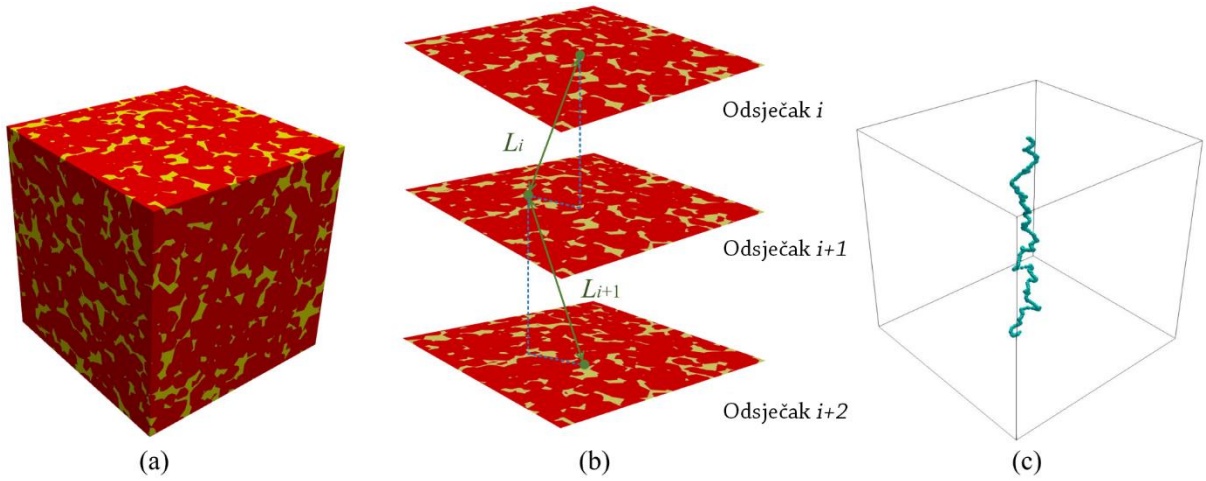
Slika 2-5. 3D mikrostruktura (a), porni prostor poroznog medija (b), karta udaljenosti izračunata *fast marching* metodom (c) (Jinlong et al., 2020)

2.1.4 Metoda težišta pore

Metoda težišta pore određuje geometrijski tortuozitet računajući udaljenost između težišta pora za svaki 2D sloj u 3D mikrostrukturi (Gostovic et al., 1996). Efektivni porni kanal se određuje računanjem koordinata (x_i, y_i, z_i) težišta pora na svakom sloju te njihovim spajanjem u putanju. Duljina L_g putanje pora se računa prema:

$$L_g = \sum_{i=1}^{N-1} L_i = \sum_{i=1}^{N-1} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \qquad (2\text{-}4)$$

Gdje je *i* oznaka 2D sloja (*slice)*, a *N* ukupni broj 2D slojeva u 3D mikrostrukturi. Geometrijski tortuozitet je omjer L_g i L, kako je prikazano u jednadžbi (2-2). Na slici 2-6. prikazana je shema računanja težišta pora.



Slika 2-6. 3D digitalna snimka poroznog medija (a), udaljenosti L_i između težišta slojeva (b), efektivna porna putanja određena metodom težišta pore (c) (Jinlong et al., 2020)

2.2 Fizički tortuozitet

Za razliku od geometrijskog tortuoziteta koji je karakteristika porne mikrostrukture, fizički tortuozitet je karakteristika protoka fluida unutar mikrostrukture, uključujući prijenos volumena, naboja, mase i energije. Na digitalnoj mikrostrukturi poroznog medija provode se numeričke simulacije protoka fluida kako bi se izračunao tortuozitet.

2.2.1. Hidraulički tortuozitet

Pojam tortuoziteta u poroznim medijima prvi je uveo Kozeny 1927. godine pokusima na kapilarnoj cijevi. Carman je problematiku 1937. godine dodatno razradio, tako što je promatrao protok kroz cijev zapunjenu pijeskom te je predložio tortuozitet kao faktor elongacije puta fluida. Pretpostavljajući da se cijev konstantne debljine L može promatrati kao snop kapilara jednake dužine L_{eff} i konstantnog poprečnog presjeka, predložio je sljedeću Kozeny-Carman formulu:

$$k = \frac{\varphi^3}{\beta \cdot \tau^2 \cdot S^2} \tag{2-5}$$

Formula povezuje propusnost, k s četiri parametra: poroznosti ϕ, specifičnom površinom *S*, faktorom oblika β i tortuozitetom τ čija je formula:

$$\tau = \frac{L_{eff}}{L} \tag{2-6}$$

Gdje je L_{eff} duljina efektivnog hidrauličkog puta toka fluida, a *L* ravna linija koja predstavlja udaljenost kroz porozni medij u smjeru toka. Hidraulički tortuozitet se može smatrati bezdimenzionalnim faktorom koji opisuje prosječno produljenje strujnica u poroznom mediju. Treba napomenuti da tok fluida u poroznom mediju varira duž putanje toka te ovisi o obliku, orijentaciji, poprečnom presjeku, grananju i spajanju pora, zbog čega je teško identificirati strujnice na razini pornog toka.

Tok fluida u digitalnoj mikrostrukturi može se simulirati Boltzmannovom metodom rešetke (engl. *Lattice Boltzmann Method*, LBM) i metodom konačnih volumena (engl. *Finite Volume Method*, FVM), nakon čega se hidraulički tortuozitet računa analizirajući strujnice iz simulacije (Krüger et al., 2017). Postoji nekoliko metoda za računanje tortuoziteta iz strujnica. Efektivna hidraulička duljina L_H se dobije iz težinskog prosjeka duljina strujnica te se tortuozitet računa prema jednadžbi:

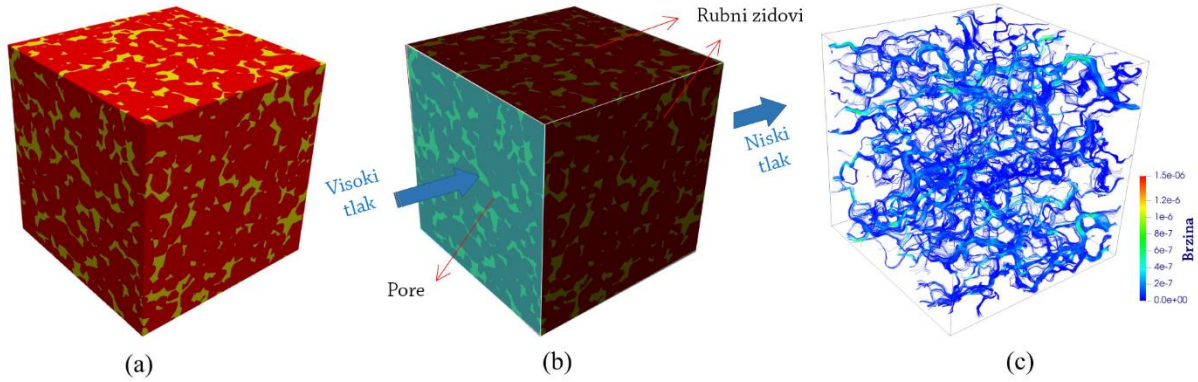
$$\tau_h = \frac{1}{L} \cdot \left(\frac{\sum_i w_i l_i}{\sum_i w_i} \right) \tag{2-7}$$

Gdje i označava strujnicu, l_i duljinu strujnice, a w_i težinu strujnice koja predstavlja njen doprinos ukupnom toku. Postoji nekoliko načina određivanja težine strujnice, a jedan od

najčešće korištenih je recipročna vrijednost vremena puta t_i čestice fluida duž strujnice. Novija jednadžba za računanje hidrauličkog tortuoziteta je:

$$\tau_h = \frac{\int v(r)d\Omega}{\int v_x(r)d\Omega} = \frac{\langle v \rangle}{\langle v_x \rangle} \tag{2-8}$$

Gdje je $v_x(r)$ makroskopska komponenta brzine duž toka fluida u točki r, a zagrade $\langle\ \rangle$ predstavljaju prostorni prosjek u pornom prostoru Ω (Jinlong et al., 2020). Ovom jednadžbom direktno se računa hidraulički tortuozitet iz stacionarnog polja brzine fluida, čime se izbjegava problem identificiranja individualnih strujnica. Ova metoda ima široku primjenu jer pojednostavljuje računanje hidrauličkog tortuoziteta u poroznom mediju s kompleksnom geometrijom. Treba napomenuti da jednadžba sadrži dvije pretpostavke: fluid je nestlačiv i tok fluida nije konkavan. Na slici 2-7. prikazana je shema pronalaska hidrauličkog tortuoziteta.



Slika 2-7. 3D porna mikrostruktura (a), rubni uvjeti za simulaciju (b), polje ustaljenih brzina fluida unutar poroznog medija (c) (Jinlong et al., 2020)

2.2.2. Električni tortuozitet

Električni tortuozitet se može izmjeriti pokusom električne vodljivosti, u kojem se uzorak poroznog medija zasiti elektrolitom poznate električne vodljivosti σ_o te se, zbog razlike napona, naboj prenosi kroz porni prostor u kojem je elektrolit. Izmjerena efektivna vodljivost σ_{eff} je manja od vodljivosti elektrolita, zbog usporavajućeg učinka porne mikrostrukture na prijenos naboja (Garrouch et al., 2001). Razlika između σ_o i σ_{eff} predstavlja električni tortuozitet τ_e:

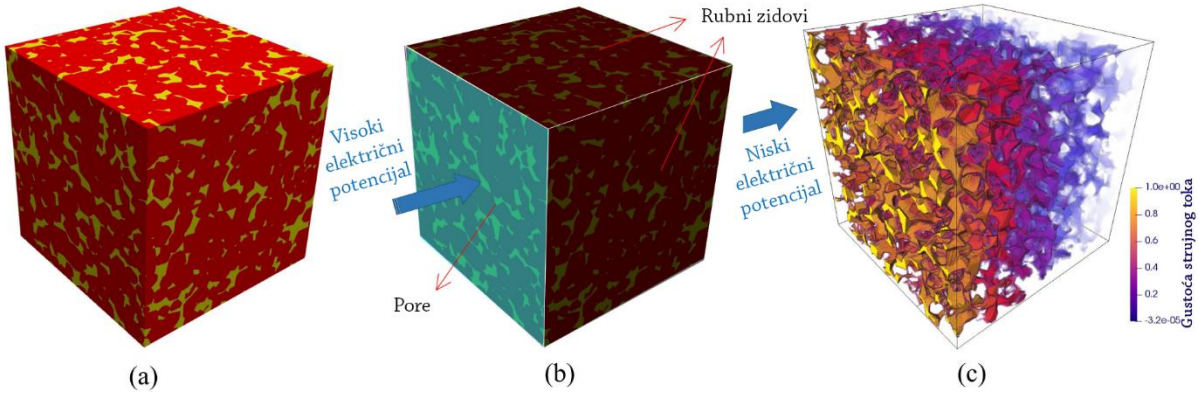
$$\sigma_{eff} = \sigma_0 \frac{\phi}{{\tau_e}^2} \tag{2-9}$$

Gdje je φ poroznost. Kako bi električni tortuozitet bio dosljedan hidrauličkom, redefiniran je kao:

$$\tau_e = \frac{L_e}{L} \tag{2-10}$$

Gdje je L_e efektivna duljina puta električnog prijenosa.

Električni tortuozitet u poroznom mediju se također može simulirati mnogim numeričkim metodama. Jedna od njih je metoda konačnih volumena (engl. *Finite Volume Method*, FVM). Matriks stijene se smatra homogenim i izolirajućim, a porni prostor je zasićen elektrolitom. Električna vodljivost u ustaljenom stanju ovisi o očuvanju naboja i Ohmovom zakonu (Moukalled et al., 2016). Električni tortuozitet se često koristi kao zamjena za hidraulički tortuozitet jer je proračun jednostavniji i brži. Na slici 2-8. prikazan je primjer simuliranog električnog toka unutar poroznog medija.



Slika 2-8. 3D porozni medij (a), rubni uvjeti za simulaciju električne vodljivosti (b), gustoća električnog toka unutar poroznog medija (c) (Jinlong et al., 2020)

2.2.3. Difuzni tortuozitet

Tortuozitet poroznog medija se također može izmjeriti testom difuznosti, u kojem se koriste plinovi zbog velike brzine difuzije. Efektivna difuznost D_{eff} u poroznom mediju je manja od ukupne difuznosti plina D_0 (Garrouch et al., 2001). Difuzni tortuozitet se računa prema:

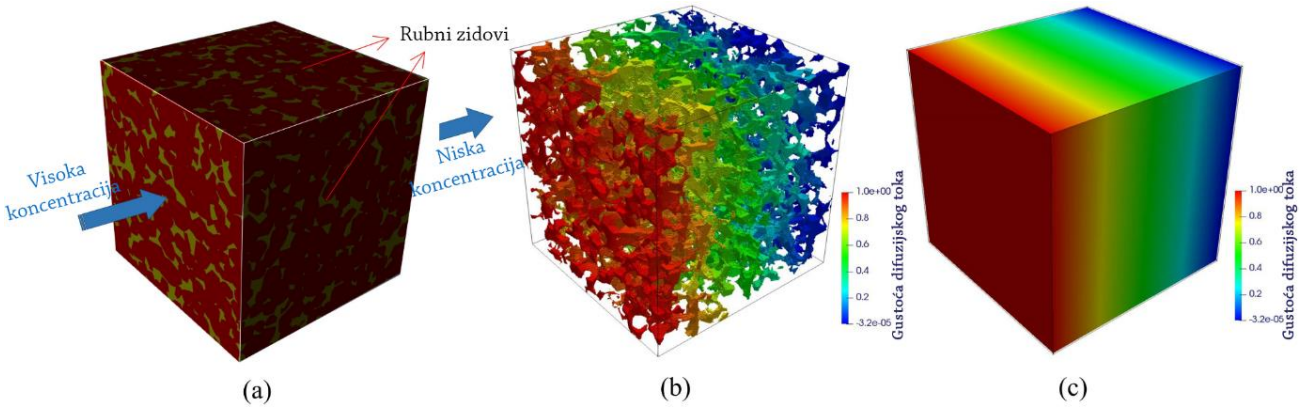
$$D_{eff} = D_0 \frac{\phi\delta}{{\tau_d}^2} \quad (2\text{-}11)$$

Gdje je δ koeficijent suženja zbog varijacija u promjeru pora duž difuznog puta. U modelu poroznog medija s cilindričnim kapilarama, difuzni tortuozitet je definiran kao omjer efektivne duljine difuznog puta L_d i udaljenosti ravne linije L:

$$\tau_d = \frac{L_d}{L} \quad (2\text{-}12)$$

Difuzija unutar poroznog medija može biti klasična molekularna difuzija i Knudsenova difuzija, ovisno o omjeru puta molekule i promjera pore. Ako je put relativno kratak u usporedbi s veličinom pore, dolazi samo do molekularne difuzije, te se proces difuzije može opisati Fickovim zakonom. Do Knudsenove difuzije dolazi u slučajevima kada je velika duljina puta u usporedbi s veličinom pore. Molekule se sudaraju s zidovima pora češće nego međusobno, čime se usporava tok molekula (Malek i Coppens, 2003). Zbog toga, rezultati tortuoziteta iz više testova difuzije mogu davati različite rezultate.

Prethodno opisane numeričke metode koriste se i za proračun difuznog tortuoziteta. Difuzni tortuozitet se računa uspoređujući ustaljene difuzijske tokove kroz kubični porozni medij. Zbog razlike u koncentraciji, čestice kreću od ulaza u mikrostrukturu do izlaza. Kad se dostigne ustaljeno stanje, iz Fickova zakona je razvidan uvjet difuznog protjecanja kroz porozni i homogeni medij. Iz njihova omjera se dobije difuzni tortuozitet. Na slici 2-9. prikazana je shema simuliranja difuznog toka na uzorku poroznog medija.



Slika 2-9. Shema rubnih uvjeta za simulaciju molekularne difuzije u poroznom mediju (a), gustoća difuznog toka u poroznom mediju (b), gustoća difuznog toka u homogenom mediju (c) (Jinlong et al., 2020)

2.2.4. Toplinski tortuozitet

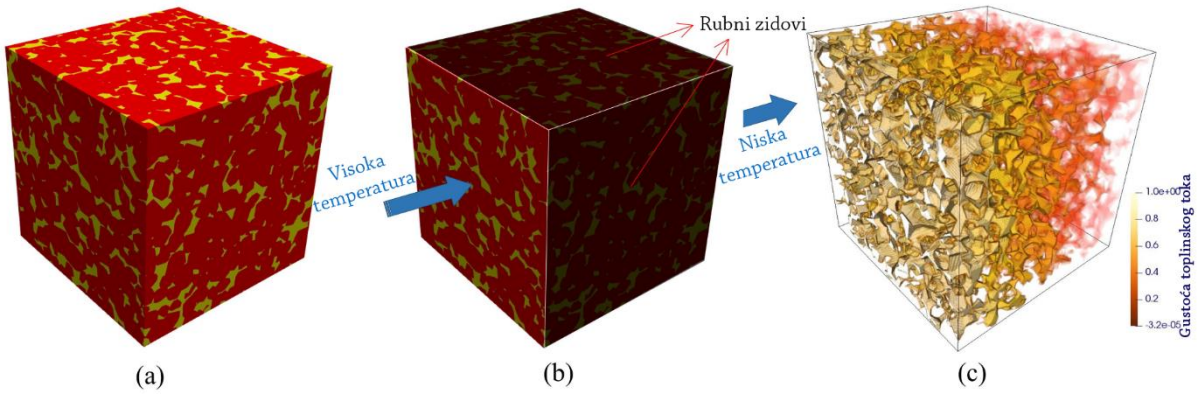
Prijenos topline u poroznom mediju uglavnom se određuje kondukcijom. Konvekcija i radijacija se mogu ignorirati jer je toplinska vodljivost čvrste faze mnogo veća od pora ispunjenih zrakom (Bodla et al., 2012). Međutim, kako se traži tortuozitet pornog prostora, potrebno je postaviti čvrstu fazu kao toplinski izolator. Efektivna toplinska vodljivost λ_{eff} se određuje iz simulacije toplinskog toka, iz čeg se određuje toplinski tortuozitet τ_t:

$$\lambda_{eff} = \lambda_0 \frac{\phi}{{\tau_t}^2} \tag{2-13}$$

Gdje je λ_0 ukupna toplinska vodljivost zraka.

Prijenos topline kroz porni prostor se numerički simulira metodom konačnih volumena (Cooper et al., 2014). Porni prostor je zasićen zrakom toplinske vodljivosti λ_0, dok se za čvrstu fazu uzima da je homogeni toplinski izolator. Postavlja se konstantna razlika temperature između ulaza i izlaza kubičnog poroznog medija. Toplina će se prenositi iz područja veće temperature u područje manje temperature, potaknuta razlikom u temperaturi. Kada se dosegne ustaljeno stanje, prijenos topline može se opisati zakonom očuvanja energije i Fourierovim zakonom. Toplinski tortuozitet se računa kao omjer prijenosa topline u poroznom mediju i prijenosa topline za homogeni medij koji se sastoji od zraka istih

kubičnih dimenzija i istog temperaturnog gradijenta. Na slici 2-10. prikazan je postupak simuliranja prijenosa topline kroz porozni medij.

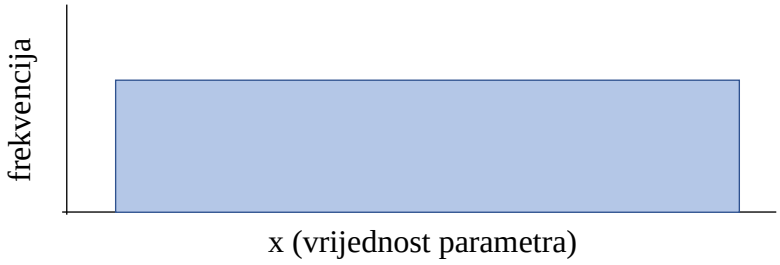


Slika 2-10. 3D porozni medij (a), rubni uvjeti za simulaciju prijenosa topline (b), prijenos topline unutar poroznog medija (c) (Jinlong et al., 2020)

2.3 Stohastičke metode u analizi digitalnog uzorka stijene

Primjena statističkih metoda u fizici, pa tako i u analizi petrofizikalnih svojstava sastoji se od *randomizacije* u samom postupku, pri čemu se određeni parametri *uzorkuju* generiranjem slučajnih brojeva koji nakon dovoljnog broja generiranih brojeva zadovoljavaju određenu statističku raspodjelu.

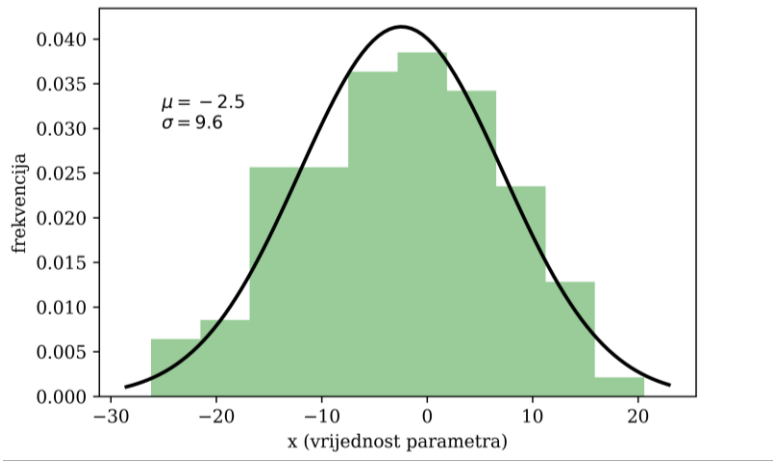
Najjednostavnija raspodjela je uniformna, tj. ona kod koje je jednaka vjerojatnost pojave nekog parametra jednaka unutar zadanog raspona očekivanih vrijednosti (slika 2-11.). Vjerojatnost pojave se pokazuje dijagramom, tj. histogramom vrijednosti x i broja pojava vrijednosti x u određenom rasponu, što, kad se pokaže u normiranom obliku daje frekvenciju, tj. vjerojatnost pojave parametra u svakom od definiranih raspona (razreda vrijednosti).



Slika 2-11. Histogram normalne raspodjele

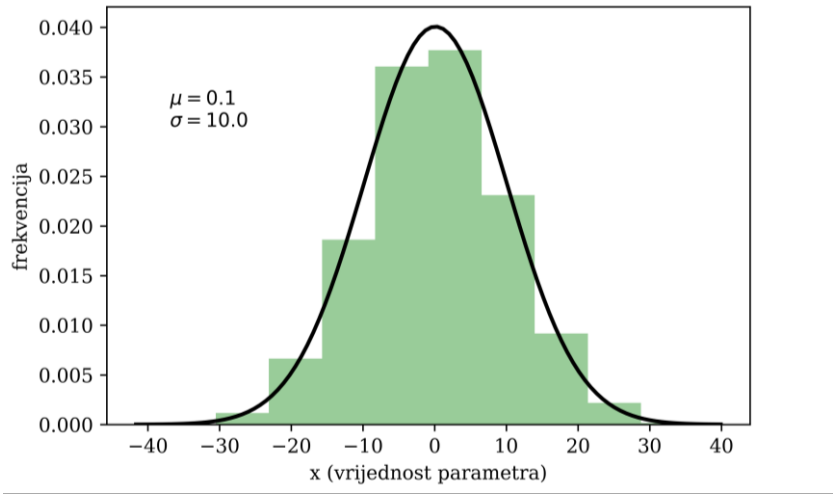
Koliko će se slučajno uzorkovanje približiti teoretski idealnoj raspodjeli ovisi o broju generiranih slučajnih brojeva. Tip raspodjele može se odrediti na temelju prethodnog iskustva (publikacija u kojima je rađena analiza raspodjele na velikom, tj. reprezentativnom broju ulaznih podataka), vizualno ili testiranjem određene raspodjele. Testiranje raspodjele je matematički najdosljedniji način određivanja raspodjele, međutim nedovoljnim poznavanjem svojstava raspodjele, njene primjenjivosti ili testiranjem na nedovoljnom broju podataka (slika 2-12., 2-13. i 2-14.), parametri odabira raspodjele mogu voditi ka krivo odabranoj raspodjeli.

Stoga, u prirodi, tj. fizici, najčešće se koristi normalna distribucija ili nagnuta normalna distribucija (poput log-normalne).

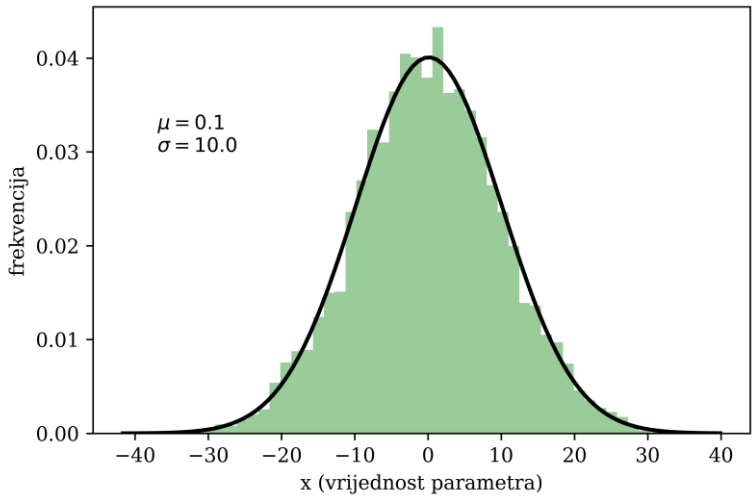


Slika 2-12. Uzorak od 100 slučajno generiranih brojeva prema normalnoj raspodjeli (centralna vrijednost $\boldsymbol{\mu = 0}$, raspršenost, tj. standardna devijacija $\boldsymbol{\sigma = 10}$, broj stupaca u

histogramu 10), s parametrima dobivenim testiranjem raspodjele (***μ i σ*** prikazani u samom dijagramu).



Slika 2-13. Uzorak od 5000 slučajno generiranih brojeva prema normalnoj raspodjeli (***μ* = 0**, ***σ* = 10**, broj stupaca u histogramu 10), s parametrima dobivenim testiranjem raspodjele.



Slika 2-14. Uzorak od istih kao u prethodnoj slici 5000 slučajno generiranih brojeva, broj stupaca u histogramu 50.

Mjerila, tj. parametri kojima se može kvantificirati svojstva neke raspodjele ili nekog statističkog uzorka su:

- funkcija gustoće vjerojatnosti (engl. *Probability Density Function*, PDF)
- funkcija kumulativne distribucije (engl. *Cumulative Distribution Function*, CDF)
- percentilna funkcija (tj. CDF inverzna)
- statistika raspodjele tj. centralna vrijednost, varijanca te nakošenost ili kurtoza
- slučajni uzorak (engl. *Random Variable Sample*, RVS)

3. ALGORITAM *SLUČAJNOG HODA*

Slučajni hod (engl. *random walk*) metoda je koja je razvijena kako bi se analiziralo statističku vjerojatnost kretanja naizgled nepredvidivim, ili kompleksnim putanjama. To je proces gdje trenutna vrijednost neke varijable y_t sadrži u sebi informaciju o istoj varijabli u prethodnom trenutku (položaju, y_{t-1}) i šum (a_t, engl. *noise*). Nepredvidivost, ili šum u najjednostavnijem obliku se definira kao normalna varijabla sa centralnom vrijednosti nula i varijancom = 1:

$$y_t = y_{t-1} + a_t \tag{3-1}$$

Ovako definiran proces slučajnog hoda implicira kako je najbolja predikcija varijable za sljedeći period upravo sadašnja vrijednost, čineći varijablu y_t djelomično determinističkom a djelomično stohastičkom.

Najčešće, kada se analizira određeni proces, on ima već određen trend. To znači da deterministički dio procesa (y_{t-1}) može biti definiran zasebnom funkcijom. Najčešće se ovakav proces zove Wienerov proces koji je prvotno služio za opis geometrijskog Brownovog gibanja (engl. *Geometric Brownian Motion*, GBM method):

$$dy = \mu \cdot dt + \sigma \cdot dz \tag{3-2}$$

Pri čemu je *dy* promjena varijable y u vremenu *dt*, μ *predstavlja trend* (engl. *drift*), σ je standardna devijacija, tj. stohastički element, a *dz* je Wienerov proces u infinitezimalno kratkom vremenu Δt:

$$dz = \varepsilon_t \sqrt{dt} \tag{3-3}$$

pri čemu slučajna varijabla ε_t, N (0,1) ima normalnu raspodjelu.

Generalizirani Wienerov proces glasi:

$$dy = a(y,t) \cdot dt + b(y,t) \cdot dz \tag{3-4}$$

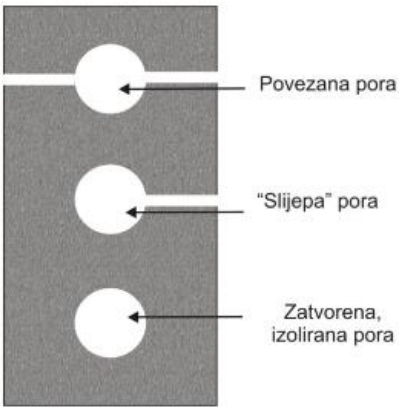
A specijalan slučaj kada je $\mu \cdot S = a(S,t)$ i $\sigma \cdot S = b(S,t)$, uz konstantne μ i σ je tada GBM:

$$dy = \mu \cdot y \cdot dt + \sigma \cdot y \cdot dz \tag{3-5}$$

Metode slučajnog hoda (engl. *Random Walk*, RW) ponavljajuće su Monte Carlo metode koje se koriste za modeliranje konvektivnog i difuznog transporta u složenim heterogenim medijima, u medicini, meteorologiji i financijskim analizama. RW tehnike također se mogu koristiti kao izravna tehnika simulacije *tracera* ili transport onečišćivača u vodonosnicima (Kinzelbach, 1988; Zimmermann et al., 2001; Delay et al., 2005). Na laboratorijskoj razini, nedavni napredak u tehnikama snimanja tekstura stijena daje slike visoke rezolucije prostora pora iz čega se mogu RW metodom kvalitetno procijeniti električna vodljivost (Kim i Torquato, 1992) ili propusnost u raspucanim stijenama (Noetinger i Estebenet, 2000; Noetinger et al., 2001a; Noetinger et al., 2001b)

Za proračun tortuoziteta napravljen je algoritam temeljen na slučajnom (nasumičnom) hodu čestice kroz porni prostor. Algoritam funkcionira tako da na „ulaznom“, prvom sloju (engl. *sliceu)* duž z osi, nasumično odabere točku koja predstavlja voksel unutar pore. Zatim nasumičnim odabirom voksela pore u sljedećem sloju, koja je u kontaktu s prethodno odabranom porom (tj. vokselom) nastoji proći kroz cijeli uzorak i doći do „izlaznog“, zadnjeg sloja duž z osi. Pređena putanja se sprema i računa se njen tortuozitet. Algoritam je implementiran u programskom jeziku *Python*.

Prije upotrebe programa potrebno je digitalnu snimku uzorka prebacit iz RAW formata (slike) u seriju .npz (*numpy array* binarni zapis, Harris et al., 2020) datoteka gdje svaka datoteka sadrži informacije o pornom prostoru određenog sloja duž z-osi. Porni prostor se sastoji od mreže povezanih pora, no unutar digitalne snimke uzorka nalaze se izolirane pore, prikazane na slici 3-1., koje mogu uzrokovati probleme prilikom nasumičnog odabira pora te ih program, prije pronalaženja putanje, izbacuje iz podataka.



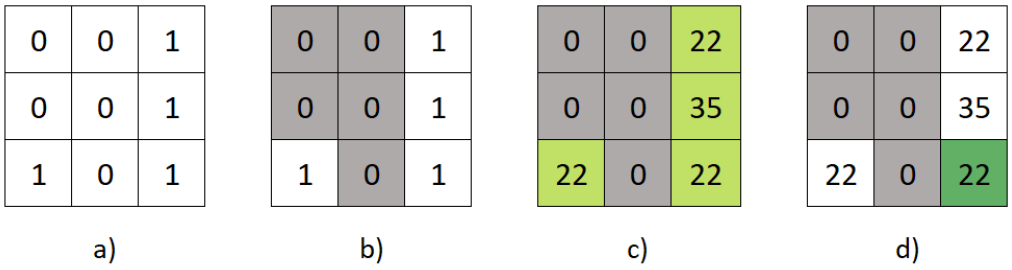
Slika 3-1. Shematski prikaz mogućih pojavljivanja pora (Vulin, 2022)

Algoritam prvo uzima točku iz pornog prostora u prvom sloju. Za tu se točku promatra u sljedećem sloju gdje se nalazi matriks stijene, a gdje se čestica može slobodno gibati. Mogući smjerovi kretanja su ravno, gore, dolje, lijevo, desno te dijagonalno u kutove. Prikaz mogućih kretanja se odvija u obliku 3×3 matrice, u kojoj centralna vrijednost predstavlja pravocrtno kretanje te se, za pojedini smjer kretanja, sprema 1 ako je pomak moguć, a 0 ako nije. Odabir točke od mogućih opcija se odvija preko normalne raspodjele. Svakoj ćeliji 3×3 matrice dodjeljuje se određena postotna šansa za mogućnost pomaka na tu lokaciju, gdje su postotne šanse određene formulom za normalnu raspodjelu dvodimenzionalnog polja:

$$f(x) = \frac{1}{2\pi \cdot \sigma_x \cdot \sigma_y} \cdot e^{-\frac{(x-\mu_x)^2}{2\sigma_x{}^2} + \frac{(y-\mu_y)^2}{2\sigma_y{}^2}} \tag{3-6}$$

Gdje su σ_x i σ_y devijacije normalne raspodjele po x i y osi, μ_x i μ_y očekivanja po x i y osi, a x i y relativne koordinate unutar 3×3 matrice.

Na slici 3-2. dan je primjer određivanja točke pomoću 3×3 matrice u sljedećem sloju. Na prvoj slici prikazano je stanje u narednom sloju oko trenutne točke, gdje je 0 matriks stijene, a 1 porni prostor. U sljedećem koraku identificiraju se mogući pomaci označeni jedinicom. Nakon toga se svakom mogućem pomaku dodjeljuje postotna šansa pomoću jednadžbe (3-6), te se, uzimajući u obzir dodijeljene šanse, nasumičnim odabirom izabire sljedeća točka putanje. Postupak se ponavlja dok se ne dođe do zadnjeg sloja.



Slika 3-2. Shema određivanja pomaka u sloju: stanje oko točke u sljedećem sloju (a), identificiranje mogućih pomaka (b), dodjeljivanje postotne šanse mogućim pomacima (c), konačni odabir pomaka (d)

Nakon odabira točke iz matrice kao sljedeći pomak, spremaju se koordinate odabrane točke te se točka (ukoliko je na toj koordinati pora) označava kao posjećena. Postupak se ponavlja sve dok zadnja odabrana točka ne bude iz posljednjeg sloja duž z osi, čime je algoritam

došao do kraja uzorka i pronašao jednu od mogućih putanja čestice. U slučaju da je algoritam *zalutao* u slijepu poru te ne postoji put u sljedeći sloj, algoritam pokušava pronaći put oko sebe u istom sloju. Ako ne uspije pronaći put, vraća se korak unazad i odabire točku koju nije odabrao kao odabrani pomak te ponavlja postupak sve dok ne dođe do zadnjeg sloja po z osi. Spremljene točke predstavljaju točke putanje čestice. Poznavajući točke putanje, moguće je izračunati efektivnu duljinu putanje L_{eff} prema jednadžbi:

$$L_{eff} = \sum_{i=1}^{n} \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 + (z_{i+1} - z_i)^2} \quad (3\text{-}7)$$

Gdje su x_i, y_i i z_i koordinate točke u sloju i, a x_{i+1}, y_{i+1} i z_{i+1} koordinate točke u sloju $i+1$.

Tortuozitet putanje τ se računa kao omjer efektivne duljine putanje L_{eff} i duljine uzorka po z osi L:

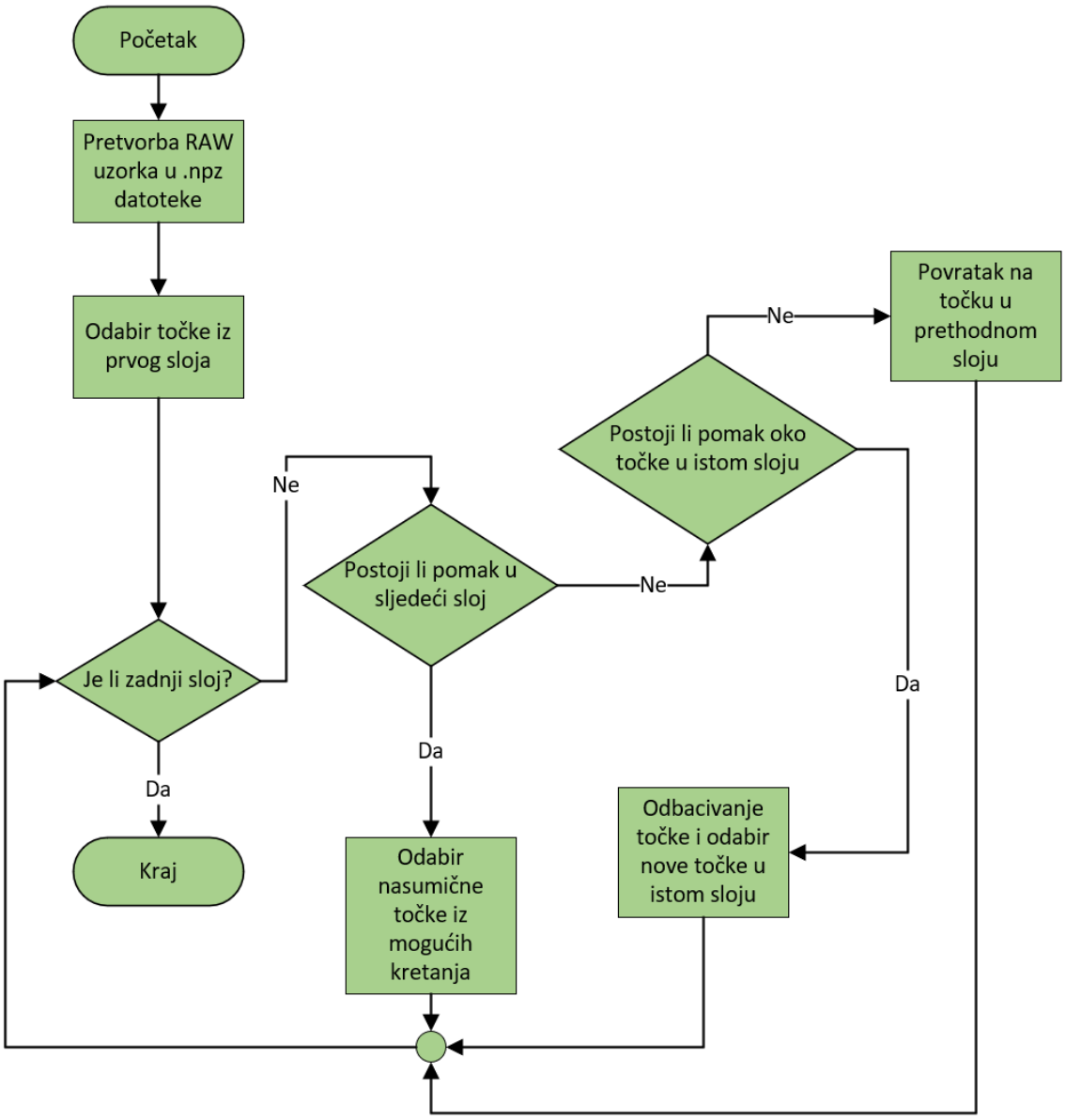
$$\tau = \frac{L_{eff}}{L} \quad (3\text{-}8)$$

Opisani postupak se provodi za svaku točku koja predstavlja poru u prvom sloju, čime se dobije lista putanja za svaku moguću početnu točku te pripadajuće vrijednosti tortuoziteta svake putanje. Iz ovoga je moguće izračunati srednju vrijednost tortuoziteta cijelog uzorka preko jednadžbe:

$$\tau_{avg} = \frac{1}{n} \sum_{i=1}^{n} \tau_i \quad (3\text{-}9)$$

Gdje je n broj putanja, a τ_i vrijednost tortuoziteta pojedine putanje.

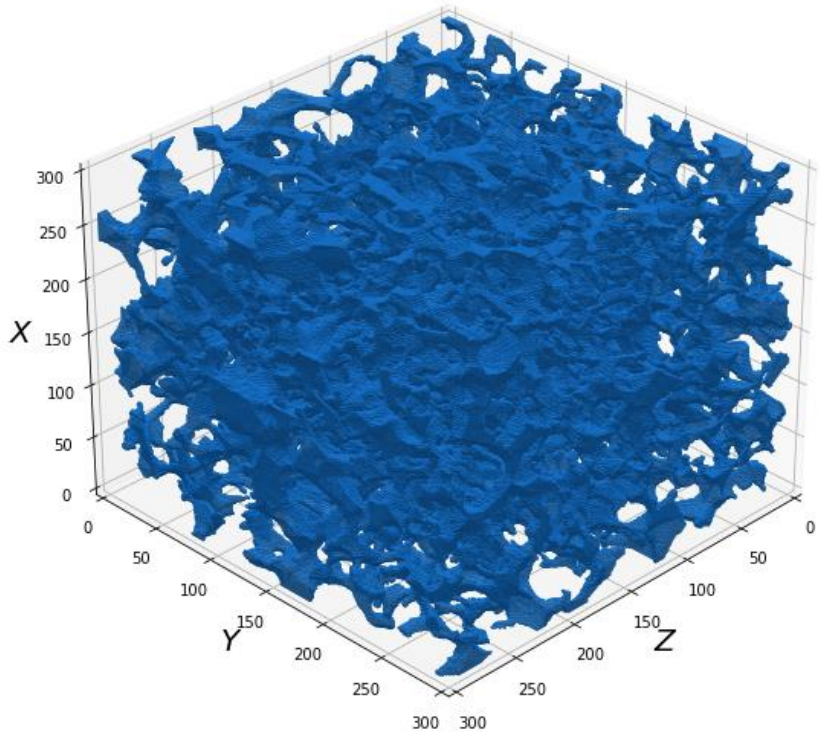
Na slici 3-3. dan je dijagram tijeka algoritma.



Slika 3-3. Dijagram tijeka algoritma

4. REZULTATI

Prethodno opisani stohastički algoritam korišten je za izračun tortuoziteta na referentnom uzorku (u analizama označen kao „S1“, Øren and Bakke, 2002., 2003). Radi se o Berea pješčenjaku, koji je snimljen rezolucijom od 8,638 μm i tako obuhvatio dimenzije 850×850×400 voksela. Dostupan uzorak je veličine 300 × 300 × 300 voksela (Dong, 2008). Efektivna poroznost uzorka iznosi 16,36%. Na slici 4-1. prikazan je model pornog prostora uzorka na kojemu je rađen proračun.

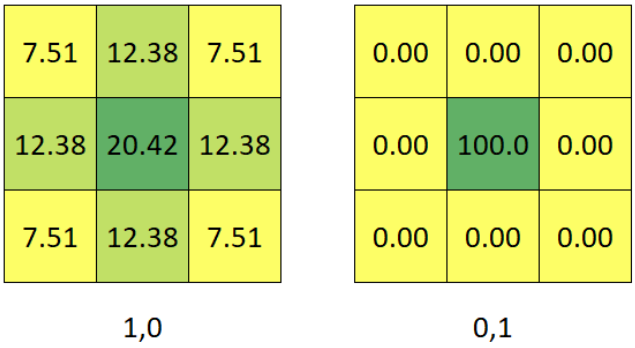


Slika 4-1. Model pornog prostora promatranog uzorka

Uzorak je podijeljen na 300 slojeva po z osi, duž kojih je rađen proračun tortuoziteta. U prvom, početnom sloju, nalazi se ukupno 11905 točaka koje predstavljaju porni prostor. Za svaku od tih točaka rađen je stohastički hod kroz uzorak s jednom od tih točaka kao početnom te je zabilježena nastala putanja. Zatim je izračunat tortuozitet svake putanje te srednji tortuozitet svih putanja.

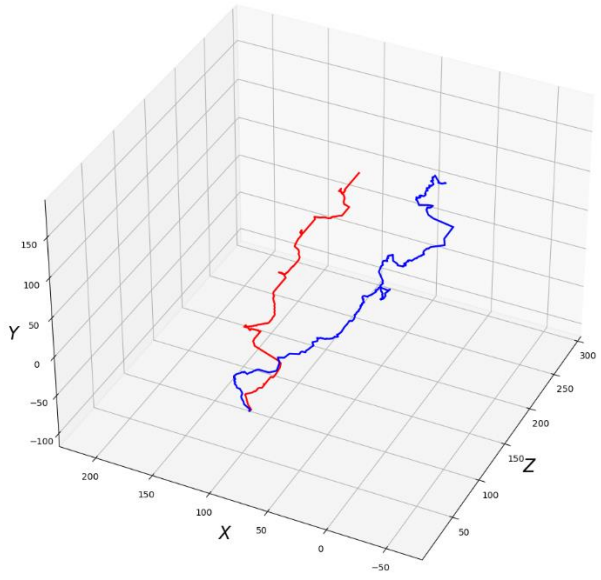
Proračun je rađen dva puta, jedanput sa standardnom devijacijom 1,0 i drugi put sa standardnom devijacijom 0,1. U teoriji, pronađene putanje raspodjelom sa standardnom

devijacijom 1,0 trebale bi imati veći tortuozitet od putanja pronađenih sa devijacijom 0,1. Do toga dolazi jer putanje s većom devijacijom imaju veću vjerojatnost da „skrenu“, tj. manja je šansa za pravocrtno gibanje. Na slici 4-2. dan je prikaz dvaju setova 3×3 matrica koje predstavljaju postotne šanse za odabir točke u sljedećem sloju. Lijevo su prikazane postotne šanse za devijaciju 1,0, a desno za devijaciju 0,1. Vidljivo je da je pri devijaciji 1,0 šansa za pravocrtan pomak oko 20%, dok je za devijaciju 0,1 šansa 100%. Iz ovoga se može zaključiti da bi tortuozitet u slučaju devijacije 0,1 trebao biti mnogo manji nego za devijaciju 1,0 jer je putanja ravnija te ima manje zakrivljenja.



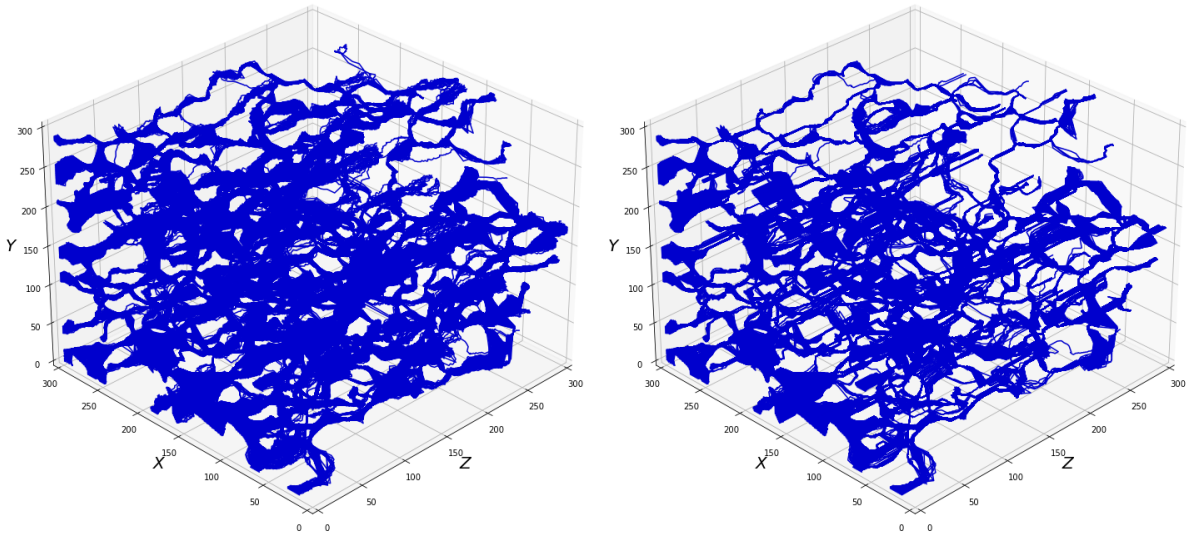
Slika 4-2. Prikaz postotnih šansi za pomak za devijacije 1,0 i 0,1

Na slici 4-3. prikazana je usporedba dviju pronađenih putanja iz nasumično odabrane točke. Crvenom bojom prikazana je putanja devijacije 0,1, a plavom bojom putanja devijacije 1,0.



Slika 4-3. Putanje za devijaciju 1,0 (plava) i devijaciju 0,1 (crvena) iz nasumično odabrane početne točke

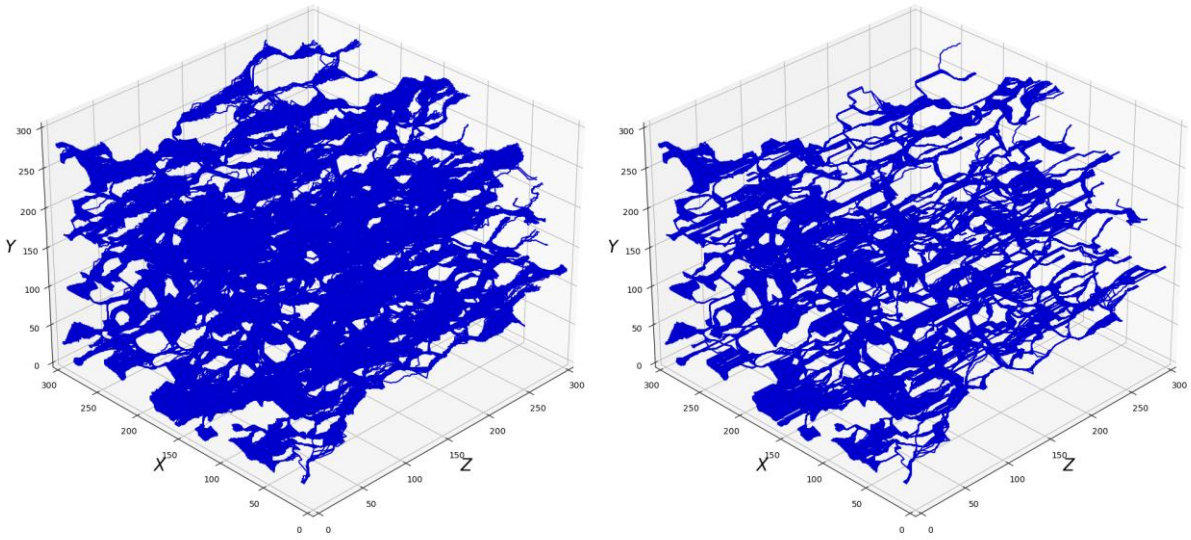
Iako je šansa da će točka duž crvene putanje birati pravocrtan smjer gotovo 100%, sama porna mikrostruktura nije pravocrtna, nego je isprepletena mreža pora. U jednom trenutku točka će pravocrtnim kretanjem doći do „zida“, odnosno matriksa stijene te neće imati izbora nego se kretati uz matriks sve dok ne bude opet mogla nastaviti pravocrtni pomak. Zbog toga čak i pri maloj devijaciji postoji određena zakrivljenost krivulje. Izračunati tortuozitet crvene krivulje iznosi 1,57, dok je tortuozitet plave krivulje 2,09. To znači da je putanja duž plave krivulje dvaput dulja od duljine uzorka duž z osi, odnosno 33% dulja od putanje duž crvene krivulje. Na slici 4-4. prikazane su sve pronađene putanje iz svih točaka koje predstavljaju porni prostor u prvom sloju, lijevo za devijaciju 1,0 i desno za devijaciju 0,1.



Slika 4-4. Pronađene putanje za sve točke iz prvog sloja, lijevo za devijaciju 1,0, desno za 0,1

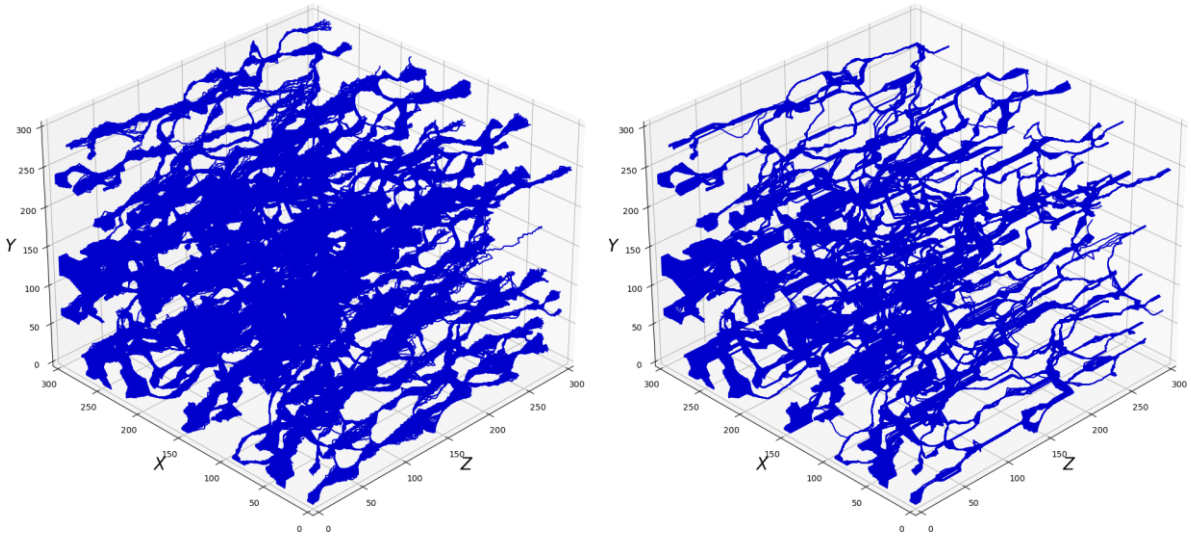
Vidljiva je razlika u gustoći između dva slučaja: u lijevom primjeru su čestice prilikom pronalaženja putanje imale veće šanse da skrenu te su ocrtavale dulje putanje, dok su se u desnom slučaju čestice nastojale kretati ravno sve dok ne dođu do fizičke prepreke, uz koju su se onda kretali sve dok pravocrtno kretanje nije ponovno bilo moguće. Srednji tortuozitet za devijaciju 1,0 iznosi 1,86, dok za devijaciju 0,1 iznosi 1,61.

Uzorak je zatim rotiran za 90° oko x osi te je ponovljen proračun putanja i tortuoziteta za sve točke u prvom sloju. U ovom slučaju su se rotacijom uzorka promijenile točke koje predstavljaju pore iz prvog sloja, jer su rotacijom gornja i donja ploha koje su predstavljale granice uzorka postale prvi i zadnji sloj. Pronađene putanje iz prvog sloja za uzorak rotiran oko x osi za obje devijacije prikazani su na slici 4-5. Srednji tortuozitet za devijaciju 1,0 iznosi 1,64, dok za devijaciju 0,1 iznosi 1,40.



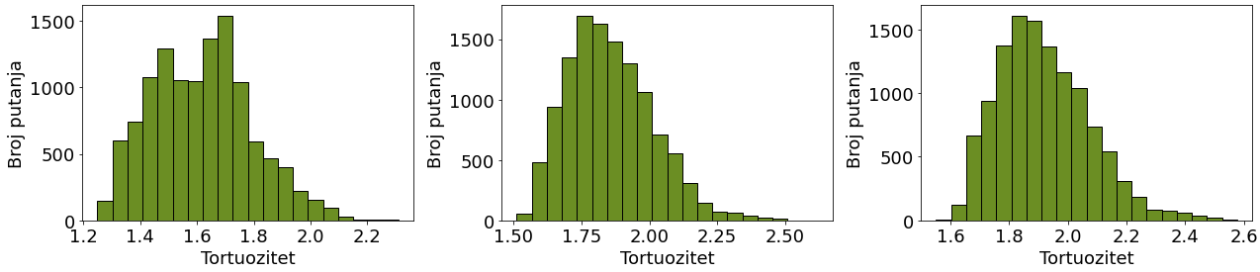
Slika 4-5. Pronađene putanje za sve točke iz prvog sloja za uzorak rotiran 90° oko x osi, lijevo za devijaciju 1,0, desno za 0,1

Postupak je ponovljen na originalnom uzorku, no ovaj put je napravljena rotacija za 90° oko y osi. Time su bočne plohe koje su predstavljale granice postale prvi i zadnji sloj. Na slici 4-6. prikazane su proračunate putanje za rotaciju uzorka oko y osi. U ovom slučaju tortuozitet za devijaciju 1,0 iznosi 1,65, dok za devijaciju 0,1 iznosi 1,4.



Slika 4-6. Pronađene putanje za sve točke iz prvog sloja za uzorak rotiran 90° oko y osi, lijevo za devijaciju 1,0, desno za 0,1

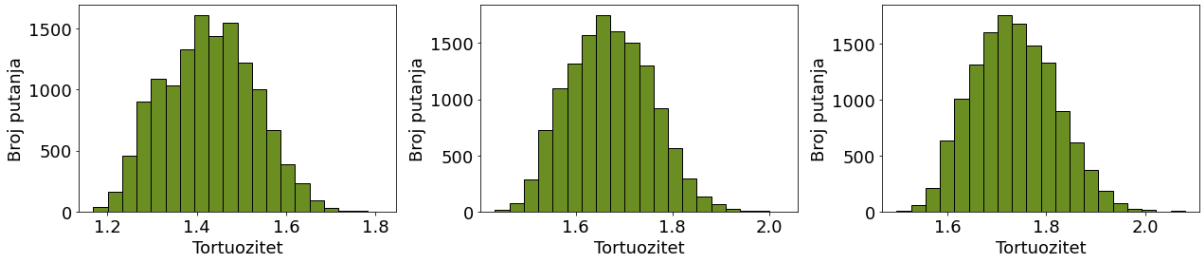
Na slici 4-7. prikazan je histogram tortuoziteta svih 11905 pronađenih putanja. Lijevi histogram prikazuje raspodjelu tortuoziteta za devijaciju 0,1, srednji histogram za devijaciju 1,0, a desni histogram za devijaciju 10,0.



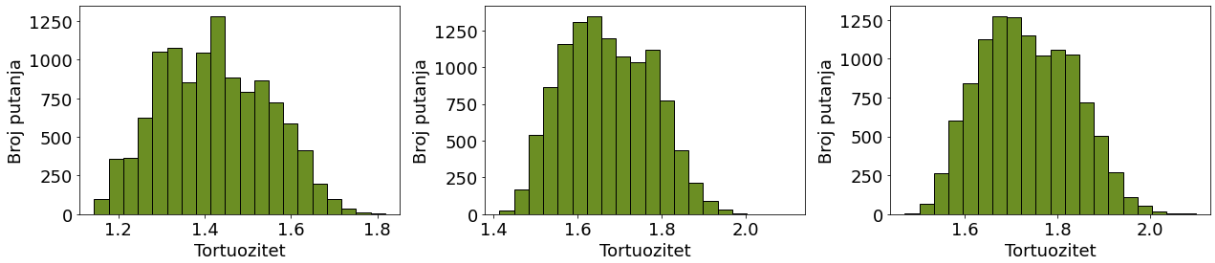
Slika 4-7. Histogram tortuoziteta za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)

Za svaku od 11905 početnih točaka koje predstavljaju porni prostor iz prvog sloja određena je putanja te tortuozitet te putanje. Tortuoziteti su raspodijeljeni u 20 grupa (engl. *bin* ili *bucket*) sa približno jednakim vrijednostima tortuoziteta po osi x. Na osi y je prikazan broj putanja u pojedinoj grupi sa sličnim vrijednostima tortuoziteta. Dobiveni graf prikazuje raspodjelu tortuoziteta u uzorku.

Za putanje devijacije 1,0 očekivanje raspodjele iznosi oko 1,8, za devijaciju 0,1 iznosi oko 1,6, dok za devijaciju 10,0 iznosi oko 1,9. Iz ovih histograma je vidljivo da, kada točke imaju više „slobode“ da se kreću u prostoru, postoji veća vjerojatnost krivudavog kretanja. Histogram izračunatih tortuoziteta poprima oblik sličan normalnoj raspodjeli, uz blagu iskošenost prema lijevo. No, kada se malom devijacijom ograniči krivudava putanja, tj. uvjetuje pravocrtno kretanje kad god je to moguće, dobivaju se teoretski najkraće moguće duljine putanja, čime se dobivaju i najmanji mogući tortuoziteti. Ipak, kako je vidljivo na histogramu devijacije 0,1, mogu se razaznati dva očekivanja, jedan oko 1,5 i jedan oko 1,7. To je svojstvo samog pornog sustava, te se može zaključiti da digitalni uzorak jezgre ima područje veće zakrivljenosti pora i područje manje zakrivljenosti pora. Postupak je ponovljen na uzorku rotiranom za 90° oko osi x (slika 4-8.) i 90° oko osi y (slika 4-9.). Kod uzorka rotiranog oko x osi može se primijetiti pravilnija distribucija, sličnija normalnoj raspodjeli, dok kod uzorka rotiranog oko y osi postoje dva očekivanja, oko vrijednosti 1,6 i 1,8 za devijaciju 1,0 te 1,7 i 1,8 za devijaciju 10,0.

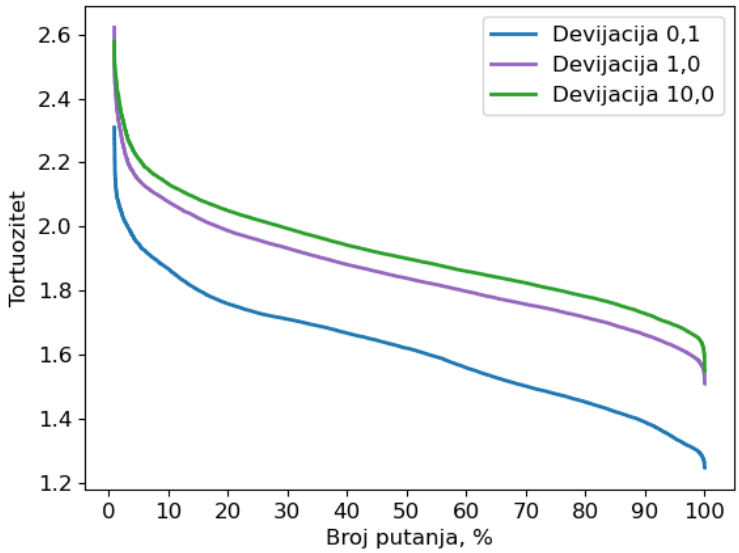


Slika 4-8. Histogram tortuoziteta uzorka rotiranog oko x osi za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)



Slika 4-9. Histogram tortuoziteta uzorka rotiranog oko y osi za devijaciju 0,1 (lijevo), 1,0 (sredina) i 10,0 (desno)

Na slici 4-10. prikazane se raspodjele tortuoziteta u obliku krivulje, gdje je na x osi broj putanja izražen u postocima, a na y osi tortuozitet za devijacije 0,1, 1,0 i 10,0.



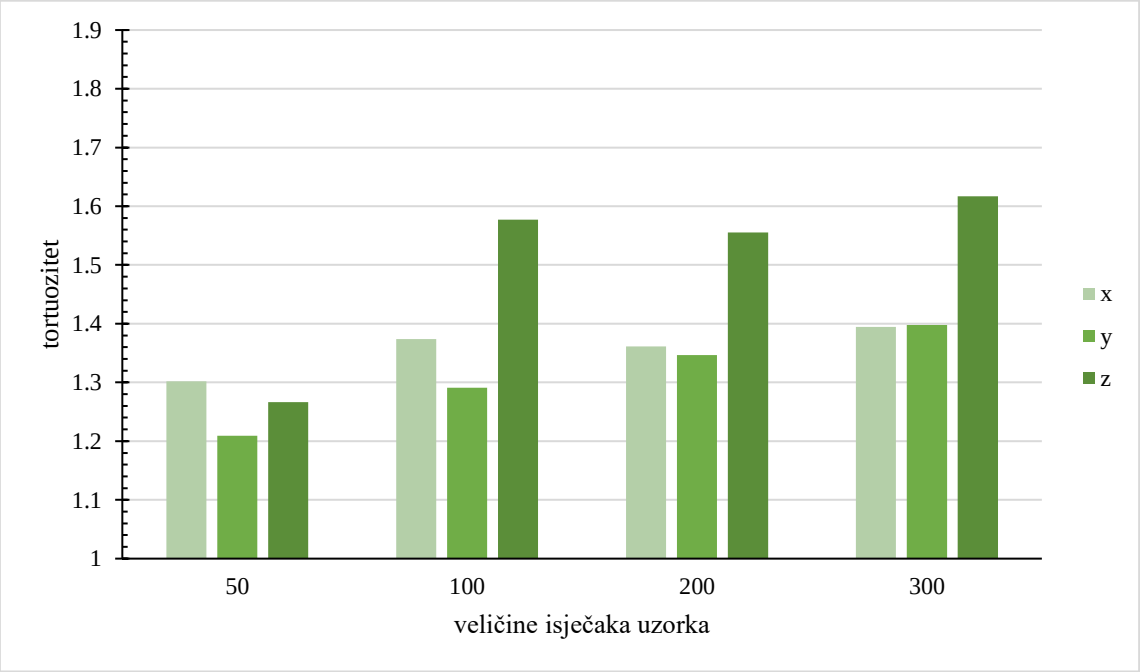
Slika 4-10. Raspodjela tortuoziteta za devijacije 0,1, 1,0 i 10,0

Od uzorka veličine 300 voksela napravljeni su manji uzorci, veličina 50, 100 i 200 voksela. U tablici 4-1 dan je prikaz izračunatih tortuoziteta za isječke uzorka veličine 50, 100, 200 i 300 voksela. Isječci su različitih veličina kako bi se odredila najmanja veličina uzorka iz koje se može izračunati reprezentativni tortuozitet, čime se štedi na vremenu izvršavanja programa. Tortuoziteti su računati za pronađene putanje devijacija 0,1 i 1,0, te duž sve 3 osi. To je postignuto rotacijom uzorka za 90° oko svake osi. Može se primijetiti da je tortuozitet u pravilu najveći duž osi z, i to za obje devijacije, dok je tortuozitet duž osi x i y približno jednak.

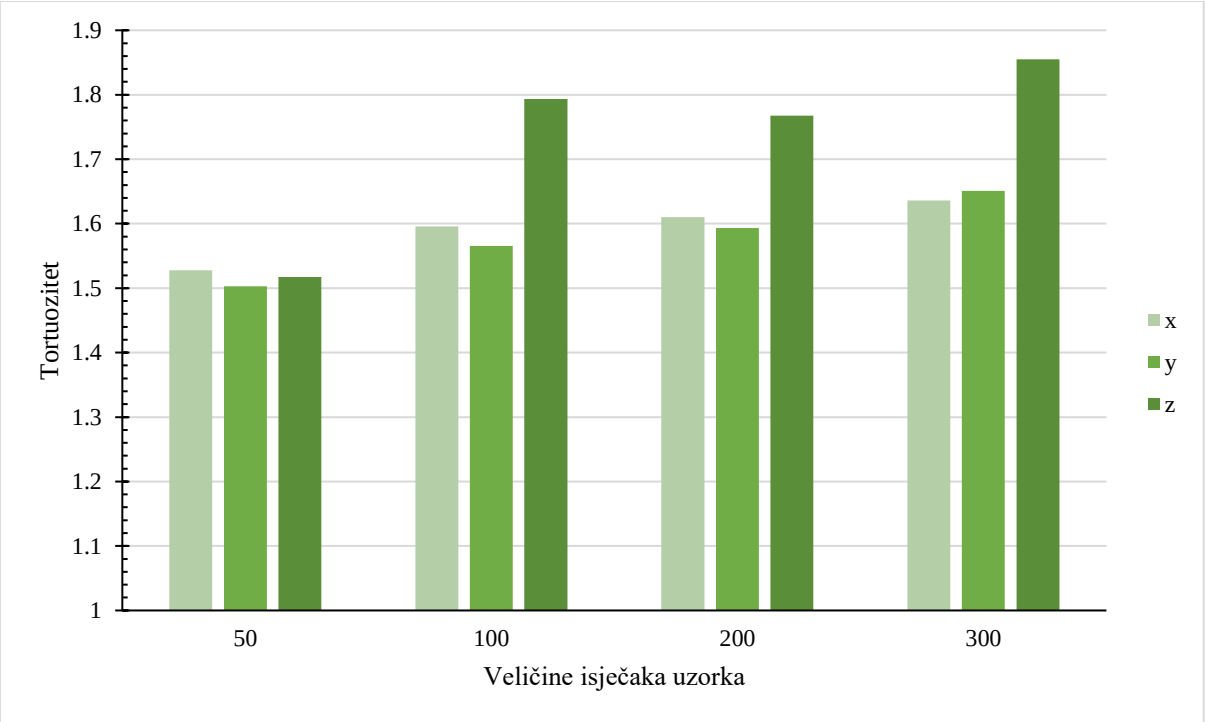
Tablica 4-1. Izračunati tortuozitet za uzorke veličine 50, 100, 200 i 300 voksela

	#50		#100		#200		#300	
	0,1	1,0	0,1	1,0	0,1	1,0	0,1	1,0
duž z osi	1,267	1,517	1,577	1,793	1,555	1,768	1,617	1,855
duž x osi	1,302	1,528	1,374	1,596	1,361	1,610	1,395	1,636
duž y osi	1,209	1,503	1,291	1,566	1,346	1,594	1,398	1,651

Na slici 4-11. prikazani su podaci iz gornje tablice u obliku dijagrama za devijaciju 0,1, gdje su na apscisi podaci grupirani prema veličinama isječaka te prikazani za pojedinu os, dok je na ordinati izračunati tortuozitet. Na slici 4-12. prikazano je isto samo za devijaciju 1,0.



Slika 4-11. Podaci o izračunatim tortuozitetima za pojedini isječak za devijaciju 0,1



Slika 4-12. Podaci o izračunatim tortuozitetima za pojedini isječak za devijaciju 1,0

5. DISKUSIJA REZULTATA

Berea pješčenjak dostupan online i analiziran u literaturi, uzet je za testiranje algoritma i koda koji proračunava koeficijent zakrivljenosti pora, tj. tortuozitet. Za razliku od publiciranih radova, u ovoj analizi se smjer povezanosti pora rotirao te je tortuozitet izračunat za putanje po z osi, a zatim po x i y osi. Na slikama 4-9 i 4-10 vidljivo je kako je tortuozitet najveći po z osi, dok je po x i y osi tortuozitet podjednak.

Analiza tortuoziteta je rađena na RGN fakultetu pomoću:

1. vlastitog (*in-house*) na RGN-u izrađenog koda za geometrijsko određivanje tortuoziteta (u svakom sloju traženi su težišta pornih površina te računate udaljenosti do težišta u idućem sloju);
2. ImageJ dodatka koji također koristi geometrijska svojstva digitalne slike (ImageJ Tortuosity Plugin, Roque and Costa, 2020), zatim su preuzeti rezultati dobiveni;
3. komercijalnim softverom koji računa metodom centralne linije kanalića pora (engl. *centerline tortuosity*, Huang et al., 2019.) te
4. korištenjem Pytrax python modula, koji također koristi svojevrsni algoritam slučajnog hoda (Tranter et al., 2019).

Usporedni rezultati pokazani su u tablici:

Tablica 5-1. Usporedni rezultati tortuoziteta dobivenih različitim metodama

metoda	**tortuozitet**
vlastiti kod – morfologija i geometrijska svojstva	1,651
ImageJ plugin – morfologija i geometrijska svojstva	1,617
LBM – metoda centralne linije	2,621
Pytrax – metoda slučajnog hoda	4,414
rezultat iz ovog rada (z-smjer putanje, σ=1,0)	1,617

Iz rezultata je razvidno kako pomoću tri metode rezultati ispadaju dosta slično. Zanimljivo je kako je pytrax algoritam najsličniji algoritmu iz ovog rada, ali daje dosta različite rezultate. Međutim, iz publikacije vezane uz pytrax modul (Tranter et al., 2019), kao i iz same dokumentacije modula, nije razvidna statistika (standardne devijacije) na temelju koje se

radi simulacija. Također, pytrax modul ne pruža mogućnost dijagonalnog kretanja, nego samo ortogonalnog te su prijelazi između pojedinih slojeva neprirodnih putanja (s ortogonalnim skretanjem koje rezultira većim ukupnim računatim duljinama). Komercijalni softver, koji u svojim proračunima koristi LBM metodu, a u slučaju računanja tortuoziteta pronalazi središnju liniju kanalića pora daje dosta različite rezultate, međutim za diskusiju je koliko bi u stvarnosti čestica fluida ili prolaz el. struje pronalazili ne geometrijom pornih kanala najkraći mogući put, nego onaj kroz sredinu pornog kanala. Vlastiti *in-house* kod i ImageJ modul koriste geometrijsko težište svake pore te se računa udaljenost geometrijskih težišta između dva proslojka. Manjkavost ove metode je fizikalna nekonzistentnost, do koje može doći ukoliko je presjek pore takav da se njegovo težište nalazi izvan samog presjeka.

6. ZAKLJUČAK

Tortuozitet, kao koeficijent zakrivljenosti pora, ima mnogo definicija. Jedna od najjednostavnijih definicija je omjer najkraćeg puta povezanih točaka i udaljenosti ravne linije između početne i krajnje točke. Takav tortuozitet se naziva geometrijski tortuozitet, te je on karakteristika mikrostrukture pora i određen je isključivo geometrijskim i morfološkim obilježjima poroznog medija. Metode određivanja geometrijskog tortuoziteta se svode na algoritme traženja najkraćeg puta, primjerice Dijkstrin algoritam.

S druge strane, fizički tortuozitet je karakteristika protoka fluida unutar mikrostrukture, te se određuje numeričkim simulacijama protoka fluida. Mogu se razaznati četiri tipa fizičkog tortuoziteta: hidraulički, električni, difuzni i toplinski tortuozitet. Neke od numeričkih metoda određivanja fizičkih tortuoziteta su metoda konačnih volumena, metoda konačnih elemenata, metoda konačnih razlika i Lattice-Boltzmannova metoda.

Gore navedene metode su kompleksne te zahtijevaju skupi softver i snažna računala. Zbog toga je, u sklopu ovog rada, napravljen stohastički algoritam koji metodom slučajne putanje pronalazi moguće putanje čestice fluida od početnog do zadnjeg sloja razmatranog digitalnog uzorka te računa njihov tortuozitet.

Na promatranom uzorku 300 × 300 × 300 voksela provedeni su proračuni tortuoziteta za devijacije 1,0 i 0,1, te je postupak ponovljen na manjim isječcima uzorka, veličine 50, 100 i 200 voksela kako bi se odredio elementarni volumen. Tortuozitet je računat kroz sve tri prostorne osi. Iz dobivenih podataka o tortuozitetima, vidljivo je da je tortuozitet u smjeru z osi, za sve veličine uzorka, veći nego tortuozitet u smjeru x i y osi. Jedan od mogućih razloga za to je anizotropnost pornog prostora, tj. postojanje horizontalnog tortuoziteta u smjeru x i y osi te vertikalnog tortuoziteta u smjeru z osi. Kako se radi o poznatom Berea pješčenjaku koji služi kao referenca za ovakve analize, može se zaključiti da je smjer z ekvivalentan smjeru vertikalne propusnosti, dok su smjerovi x i y ekvivalentni horizontalnoj orijentaciji uzorka te je na taj način objašnjiva slična vrijednost tortuoziteta u x i y smjeru, i manja od one u z smjeru.

Iako uzorak veličine 50 voksela rezultira najbržim izvršenjem koda, izračunati tortuoziteti previše odstupaju od tortuoziteta većih uzoraka. Slično vrijedi i za veličinu od 100 voksela.

Uzorci veličine 200 i 300 voksela daju približno slične vrijednosti tortuoziteta, zbog čega se uzorak veličine 300 voksela može uzeti kao reprezentativan elementarni volumen.

Kako se rezultati poklapaju s onima dobivenim pomoću dvije druge razmatrane metode postupak se pokazao pogodan za brzu procjenu tortuoziteta.

Pokazavši da je algoritam perspektivan, postaje razvidno i koji su elementi poželjni za buduće istraživanje i analize:

1. Provjera rezultata (i reprezentativnih elementarnih volumena) različitim metodama na više uzoraka;
2. Analiza na većem rasponu statističkih parametara (uključujući različite pristupe dodjeljivanja vjerojatnosti dijagonalnim pomacima);
3. Izrada vlastitog koda za računanje centralne linije (računanjem duljine centralne linije može se odrediti tortuozitet);
4. Provjera različitih strategija uračunavanja duljine slijepih pora.

Prednost koncepta koji je implementiran za računanje tortuoziteta uzorka je u jednostavnosti metode, mogućnosti definiranja različite difuznosti protjecanja (standardnih devijacija), te realističnom opisu putanje (mogući su dijagonalni smjerovi putanja u 3D prostoru, nije moguće uzeti u obzir koordinate koje su izvan pornog prostora), iz čega se može utvrditi da je potvrđena hipoteza rada.

7. LITERATURA

1. BODLA, K. K., MURTHY, J. Y., GARIMELLA, S. V., 2012. Direct simulation of Thermal Transport Through Sintered Wick Microstructures. *Journal of Heat Transfer*, 134 (1).
2. CARMAN, P. C., 1937. Fluid flow through granular beds. *Transactions of the Institution of Chemical Engineers*, 50, str. 150-166.
3. CHERKASSKY, B. V., GOLDBERG, A. V., RADZIK, T., 1996. Shortest paths algorithms: Theory and experimental evaluation. *Mathematical Programming*, 73, str. 129-174.
4. CLENNELL, M. B., 1997. Tortuosity: a guide through the maze. *Geological Society*, 122, str. 299-344.
5. COOPER, S. J., EASTWOOD, D. S., GELB, J., DAMBLANC, G., BRETT, D. J. L., BRADLEY, R. S., WITHERS P. J., Lee, P. D., MARQUIS, A. J., BRANDON, N. P., SHEARING, P. R., 2014. Image based modelling of microstructural heterogeneity in $LiFePO_4$ electrodes for Li-ion batteries. *Journal of Power Sources*, 247, str. 1033-1039.
6. DA SILVA, M. T. Q. S., CARDOSO, M. R., VERONESE, C. M. P., MAZER, W., 2022. Tortuosity: A brief review. *Materials Today: Proceedings*, 58(4), str. 1344-1349.
7. DELAY, F., ACKERER, P., DANQUIGNY, C., 2005. Simulating solute transport in porous or fractured formations using random walk particle tracking: A review. *Vadose Zone Journal* 4(2), str. 360-379.
8. DONG, H., 2008. *Micro-CT imaging and pore network extraction*. Doctoral dissertation, Department of Earth Science and Engineering, Imperial College London.
9. GARROUCH, A. A., ALI, L., QASEM, F., 2001. Using diffusion and Electrical Measurements to Assess Tortuosity of Porous Media. *Industrial & Engineering Chemistry Research*, 40 (20), str. 4363-4369.
10. GOSTOVIC, D., SMITH, J. R., KUNDINGER, D. P., JONES, K. S., WACHSMAN, E. D., 2007. Three-Dimensional Reconstruction of Porous LSCF Cathodes. *Electrochemical and Solid-State Letters*, 10 (12), str. 214-217.
11. HARRIS, C.R., MILLMAN, K.J., VAN DER WALT, S.J., 2020. Array programming with NumPy. *Nature* 585, 357–362.

12. HUANG, H., LU, X., SUKOP, M., 2019. Numerical study of lattice Boltzmann methods for a convection–diffusion equation coupled with Navier–Stokes equations. *J. Phys*. A 2011;44(5).

13. JINLONG, F., HYWEL, R. T., CHENFENG, L., 2020. Tortuosity of porous media: Image analysis and physical simulation. *Earth-Science Reviews*, 212.

14. KARANIKOLA, V., CORRAL, A. F., JIANG, H., SÁEZ, A. E., ELA, W. P., ARNOLD, R. G., 2017. Effects of membrane structure and operational variables on membrane distillation performance. *Journal of Membrane Science*, 524, str. 87-96.

15. KEHRWALD, D., SHEARING, P. R., BRANDON, N. P., SINHA P. K., HARRIS, S. J., 2011. Local tortuosity inhomogeneities in a lithium battery composite electrode. *Journal of The Electrochemical Society*.

16. KINZELBACH, W., 1988. The random walk method in pollutant transport simulation. Groundwater flow and quality modelling, *Springer*, str. 227-245.

17. KOLITCHEFF, S., JOLIMAITRE, E., HUGON, A., VERSTRAETE, J., CARRETTE, P. L., TAYAKOUT-FAYOLLE, M., 2017. Tortuosity of mesoporous alumina catalyst supports: Influence of the pore network organization. *Microporous and Mesoporous Materials*, 248, str. 91-98.

18. KOZENY, J., 1927. Über kapillare leitung des wassers im boden. Wien.

19. KRÜGER, T., KUSUMAATMAJA, H., KUZMIN, A., SHARDT, O., SILVA, G., VIGGEN, E. M., 2017. *The Lattice Boltzmann Method*. Switzerland: Springer Cham.

20. LINDQUIST, W. B., LEE, S. M., COKER, D. A., JONES, K. W., SPANNE, P., 1996. Medial axis analysis of void structure in three-dimensional tomographic image of porous media. *Journal of Geophysical Research: Solid Earth*, 101 (B4), str. 8297-8310.

21. MALEK, K., COPPENS, M. O., 2003. Knudsen self- and Fickian diffusion in rough nanoporous media. *The Journal of Chemical Physics*, 119 (5), str 2801-2811.

22. MOUKALLED, F., MANGANI, L., DARWISH, M., 2016. *The Finite Volume Method in Computational Fluid Dynamics*. Switzerland: Springer International Publishing.

23. NEMATI, R., SHAHROUZI, J. R., ALIZADEH, R., 2020. A stochastic approach for predicting tortuosity in porous media via pore network modeling. *Computer and Geotechnics*, 120.

24. NOETINGER, B., ESTEBENET, T., 2000. Up-scaling of double porosity fractured media using continuous-time random walks methods. *Transport in Porous Media* 39(3), str. 315-337.

25. NOETINGER, B., ESTEBENET, T., LANDEREAU, P., 2001a. A direct determination of the transient exchange term of fractured media using a continuous time random walk method. *Transport in Porous Media* 44(3), str. 539-557.

26. NOETINGER, B., ESTEBENET, T., QUINTARD, M., 2001b. Up scaling of fractured media: Equivalence between the large scale averaging theory and the continuous time random walk method. *Transport in Porous Media* 43(3), str. 581-596.

27. ØREN, P.-E. and BAKKE, S., 2002. Process Based Reconstruction of Sandstones and Prediction of Transport Properties. *Transport in Porous Media*, 46, str. 311-343.

28. ØREN, P.-E. and BAKKE, S., 2003. Reconstruction of Berea Sandstone and Pore-Scale Modeling of Wettability Effects. *Journal of Petroleum Science and Engineering*, 39, str. 177-199.

29. ROQUE, W. L. and COSTA, R. R., 2020. A plugin for computing the pore/grain network tortuosity of a porous medium from 2D/3D MicroCT image. *Applied Computing and Geosciences*, 5, 100019.

30. SHANTI, N. O., CHAN, V. W. L., STOCK, S. R., DE CARLO, F., THORNTON, K., FABER, K. T., 2014. X-ray micro-computed tomography and tortuosity calculations of percolating pore networks. *Acta Materialia*, 71, str. 126-135.

31. SETHIAN, J. A., 1996. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences of the United States of America*, 93 (4), str. 1591-1595.

32. TRANTER, T. G., KOK, M. D., LAM, M., & GOSTICK, J. T., 2019. Pytrax: A simple and efficient random walk implementation for calculating the directional tortuosity of images. *SoftwareX*, 10, 100277.

33. VULIN, D., 2022. *Skripta za kolegij - Višefazni protok u poroznoj stijeni*. Zagreb: Rudarsko-geološko-naftni fakultet

34. ZIMMERMANN, S., KOUMOUTSAKOS, P., KINZELBACH, W., 2001. Simulation of pollutant transport using a particle method. *Journal of Computational Physics* 173(1), str. 322-347.

8. PRILOG

-KOD

```
import itertools as it
import math
import matplotlib.pyplot as plt
from matplotlib import colors
from matplotlib.ticker import PercentFormatter
import numpy as np
import random as rnd
from scipy import ndimage as nd
from scipy.ndimage import label
import time
from tqdm import tqdm # loading bar
from pyevtk.hl import gridToVTK

class Matrica():

    def __init__(self, dim, mx, my, sx, sy):
        """
        Parameters
        ----------
        dim : integer
            Dimenzije matrice.
        mx : float
            Očekivanje (mean) normalne distribucije po x osi.
        my : float
            Očekivanje (mean) normalne distribucije po y osi.
        sx : float
            Devijacija normalne distribucije po x osi.
        sy : float
            Devijacija normalne distribucije po y osi.
        Returns
        -------
        Tau.
        """

        # npzImgDir = 'C:\\Users\\Robert\\Desktop\\Fakultet\\Diplomski\\tort\\NPZ\\' # loadanje direktorija

        self.dim = dim
        self.mx = mx
```

```
39.         self.my = my
40.         self.sx = sx
41.         self.sy = sy
42.         return
43.
44.     def set_ImgDir(self, path):
45.         self.ImgDir = path
46.         return
47.
48.     def load_Image(self, npzImgDir = None):
49.         m = np.zeros([self.dim, self.dim, self.dim], np.int16)
50.         if npzImgDir == None: npzImgDir = self.ImgDir
51.         for i in tqdm(range(1, self.dim+1)):
52.             s = np.load(npzImgDir + str(i-1) +
    '.input.npz')['arr_0'] # ucitava matricu iz filea
53.             m[i-1, :, :] = s[:self.dim, :self.dim] # sprema 2D
    matricu u 3D matricu pomičući se za brojač
54.         self.m = m
55.         return
56.
57.     def labelPores(self, myImage, pattern = 2):
58.
59.         if pattern == 1:
60.             pattern_3D = np.array([
61.               [[1, 1, 1],
62.               [1, 1, 1],
63.               [1, 1, 1]],
64.               [[1, 1, 1],
65.               [1, 1, 1],
66.               [1, 1, 1]],
67.               [[1, 1, 1],
68.               [1, 1, 1],
69.               [1, 1, 1]]])
70.
71.         if pattern == 2:
72.             pattern_3D = np.array([
73.               [[0, 1, 0],
74.                [1, 1, 1],
75.                [0, 1, 0]],
76.               [[0, 1, 0],
77.                [1, 1, 1],
78.                [0, 1, 0]],
79.               [[0, 1, 0],
80.                [1, 1, 1],
81.                [0, 1, 0]]])
82.
```

```
        labeledArray, numFeatures = label(myImage>0, structure =
pattern_3D)

        return (labeledArray, numFeatures)

    def connectedPores(self, myArray):
        connected_pores = myArray.copy()
        test_elements = np.unique(connected_pores[-
1].flatten()).copy()
        connected_pores =
connected_pores*(1*np.isin(connected_pores, test_elements))
        test_elements =
np.unique(connected_pores[0].flatten()).copy()
        connected_pores =
connected_pores*(1*np.isin(connected_pores, test_elements))

        return (connected_pores)

    def koordinate(self, lcp):
        """
        Parameters
        ----------
              lcp : numpy array
                  3D matrica nula i jedinica gdje jedinice
predstavljaju porni prostor.

              Returns
              -------
              x : numpy array
                  Lista x koordinata svih pora predstavljenih
jedinicom u matrici.
              y : numpy array
                  Lista y koordinata svih pora predstavljenih
jedinicom u matrici.
              z : numpy array
                  Lista z koordinata svih pora predstavljenih
jedinicom u matrici.
              """
              coords = np.where(lcp == 1)
              x, y, z = coords[2], coords[1], coords[0]

              return (x, y, z)

         def tau_iz_tezista(self, dim, lcp):

              """
              racuna teziste za svaki slice (sveukupno)
```

```
121.             i na temelju takvog tezista udaljenosti izmedju
   tezista susjednih sliceova
122.
123.             Parameters
124.             ----------
125.             dim : integer
126.                 Dimenzije matrice.
127.             lcp : numpy array
128.                 3D matrica nula i jedinica gdje jedinice
   predstavljaju porni prostor.
129.
130.             Returns
131.             -------
132.             tau : float
133.                 Tortuozitet izračunat iz težišta pora.
134.             """
135.
136.             self.tezista = []
137.             udaljenost, udaljenost_suma = 0, 0
138.
139.             for i in range(dim):
140.                 self.tezista.append(nd.center_of_mass(lcp[i]))
141.
142.             for i in range(dim-1):
143.                 udaljenost = math.sqrt((self.tezista[i][1] -
   self.tezista[i+1][1]) ** 2 + (self.tezista[i][0] -
   self.tezista[i+1][0]) ** 2 + (0 - 1) ** 2)
144.                 udaljenost_suma += udaljenost
145.
146.             tau = udaljenost_suma / dim
147.
148.             return tau
149.
150.         def normal2D(self, mx, my, sx, sy):
151.
152.             """
153.             Parameters
154.             ----------
155.             mx : float
156.                 Očekivanje (mean) normalne distribucije po x
   osi.
157.             my : float
158.                 Očekivanje (mean) normalne distribucije po y
   osi.
159.             sx : float
160.                 Devijacija normalne distribucije po x osi.
161.             sy : float
```

```
                Devijacija normalne distribucije po y osi.

            Returns
            -------
            distr: numpy array
                Vrijednosti distribucije za 3x3 matricu.

            """

            x = np.array([1, 0, 1])
            y = np.array([1, 0, 1])

            x, y = np.meshgrid(x, y)

            self.distr = 1. / (2. * np.pi * sx * sy) * np.exp(-((x - mx)**2. / (2. * sx**2.) + (y - my)**2. / (2. * sy**2.)))    # Formula za normalnu distribuciju

            return self.distr

        def provjera(self, z):

            """
            provjerava ostvarive putanje

            Parameters
            ----------
            z : integer
                Slice u z osi.

            Returns
            -------
            mogucnosti : array
                Mogući pomaci u x, y smjeru.

            """

            mogucnosti = []     # Ovdje se spremaju mogućnosti kretanja

            for i in range(len(self.koordinate)):   # Za sve moguće pomake provjeri
                if (self.point[1]+self.koordinate[i][1]) >= 0 and (self.point[2]+self.koordinate[i][0]) >= 0:   # Ako je unutar dimenzija max dimenzija matrice
                    if (self.point[1]+self.koordinate[i][1]) < self.dim and (self.point[2]+self.koordinate[i][0]) < self.dim:   # Ako je unutar min dimenzija matrice
```

```
                        if
self.kopija[z][self.point[1]+self.koordinate[i][1]][self.point[2]
+self.koordinate[i][0]] == 1:   # Ako je točka neposjećena
                            mogucnosti.append(i+1)  # Dodaj ju
kao mogući smjer kretanja

            return mogucnosti

        def hod(self, mogucnosti, z, mx, my, sx, sy):

            """
            Parameters
            ----------
            mogucnosti : array
                Mogući pomaci u x, y smjeru.
            z : integer
                Slice u z osi.
            mx : float
                Očekivanje (mean) normalne distribucije po x
osi.
            my : float
                Očekivanje (mean) normalne distribucije po y
osi.
            sx : float
                Devijacija normalne distribucije po x osi.
            sy : float
                Devijacija normalne distribucije po y osi.

            Returns
            -------
            self.point : tuple
                Koordinate točke koja je odabrana kao sljedeća.

            """

            self.sanse = self.normal2D(mx, my, sx,
sy).flatten()   # Normalna distribucija unutar 3x3 grida,
pretvorena u 1D listu
            self.novi = [self.sanse[mogucnosti[i]-1] for i in
range(len(mogucnosti))]   # Dodaj samo one elemente distribucije
na koje se može pomaknuti u 1D listu
            self.posto = [self.novi[i]/sum(self.novi) for i in
range(len(self.novi))]   # Normalizacija za računanje postotnog
udjela
            self.rand = np.random.choice(mogucnosti,
p=self.posto)   # Random smjer kretanja izabran iz mogućih
smjerova s postotcima određenim normalnom distribucijom

```

```
237.             for i in range(len(self.koordinate)):   # Za sve
   moguće pomake provjeri
238.                 if self.rand == (i+1):  # Ako je smjer kretanja
   (i+1)
239.                     self.point = (z,
   self.point[1]+self.koordinate[i][1],
   self.point[2]+self.koordinate[i][0])  # Nova točka
240.                     self.kopija[self.point[0]][self.point[1]][s
   elf.point[2]] = 3  # Koordinate nove točke označene kao posjećene
241.
242.             return self.point
243.
244.         def stoh(self):
245.
246.             """
247.
248.             Returns
249.             -------
250.             srednji_tau : float
251.                 Srednji tortuozitet putanja iz svih točaka
   nultog slicea izračunat stohastičkim hodom kroz pore.
252.
253.             """
254.
255.             self.sliceZ = np.zeros(len(np.where(self.z ==
   0)[0]), dtype=int)     # z koordinate svih točaka iz prvog
   (nultog) slicea
256.             self.sliceY =
   self.y[:len(self.sliceZ)]                                # y
   koordinate svih točaka iz prvog (nultog) slicea
257.             self.sliceX =
   self.x[:len(self.sliceZ)]                                # x
   koordinate svih točaka iz prvog (nultog) slicea
258.             self.test_tocke = []
259.             for i in range(len(self.sliceZ)):
260.                 self.test_tocke.append((self.sliceZ[i],
   self.sliceY[i], self.sliceX[i]))
261.
262.             self.svetocke = []     # Točke koje tvore putanju
   (koordinate putanje "jedne cestice" od z = 0 do z = n)
263.             timerpocetak = time.perf_counter()     #
   Inicijalizacija timera
264.             self.koordinate = [(-1, -1), (0, -1), (1, -1), (-1,
   0), (0, 0), (1, 0), (-1, 1), (0, 1), (1, 1)]    # Koordinate svih
   mogućih pomaka
265.             self.svitau = []    # Tau za svaku putanju se
   sprema unutar liste
266.
267.             for j in range(len(self.sliceZ)):
268.                 timerrute1 = time.perf_counter()
```

```
269.                 self.kopija =
    self.labeledConnectedPores.copy()      # Kopija lcp koja će
    služiti za označavanje već posjećenih pora
270.                 self.tocke = []     # Točke koje tvore putanju
    (koordinate putanje "jedne cestice" od z = 0 do z = n)
271.                 self.point = (self.sliceZ[j], self.sliceY[j],
    self.sliceX[j])    # Odabir početne točke
272.                 self.tocke.append(self.point)    # Spremanje
    početne točke u listu odabranih točaka
273.                 self.zpomak = 1   # Inicijalizacija pomaka po z
    osi
274.
275.
276.                 while self.zpomak != self.dim:  # Dok pomak po z
    osi ne dođe do dimenzija matrice
277.                     self.indeksi_pomaka =
    self.provjera(self.zpomak)    # Računanje mogućih točaka u
    sljedećem sliceu
278.
279.                     if len(self.indeksi_pomaka) != 0:   # Ako
    lista mogućih pomaka nije prazna
280.                         self.point =
    self.hod(self.indeksi_pomaka, self.zpomak, self.mx, self.my,
    self.sx, self.sy)     # Odaberi novu točku
281.                         self.tocke.append(self.point)   # Dodaj
    novu točku u listu odabranih točaka
282.                         self.zpomak+=1 # Pomakni se na sljedeći
    slice
283.                         # print('Mogući pomaci: {} | Odabran
    pomak: {} | Nova točka: {}'.format(self.indeksi_pomaka,
    self.rand, self.point))
284.
285.                     else:
286.                         self.strane =
    self.provjera(self.zpomak-1)  # Ako je lista mogućih pomaka
    prazna, pogledaj oko sebe gdje se može pomaknut u istom sliceu
287.                         if len(self.strane) != 0:   # Ako se
    može pomaknut oko sebe u istom sliceu
288.                             self.point = self.hod(self.strane,
    self.zpomak-1, self.mx, self.my, self.sx, self.sy)   # Pomakni se
    na točku oko sebe
289.                             self.tocke.pop()    # Ukloni
    posljednju točku iz liste odabranih točaka
290.                             self.tocke.append(self.point)    #
    Dodaj novu točku u listu odabranih točaka (kako bi bila samo
    jedna točka po sliceu)
291.                             # print('Nova bočna točka:
    {}'.format(self.point))
292.                         else:
293.                             if len(self.tocke) != 1:
```

```
                                    self.zpomak-=1 # Ako se ne može pomaknut ni naprijed ni oko sebe, vrati se na prošli slice
                                    self.tocke.pop()     # Ukloni točku iz posljednjeg slicea
                                    self.point= self.tocke[-1]  # Nova točka je zadnja odabrana
                                    # print('Obrisano | Nova točka: {}'.format(self.point))
                                else:
                                    self.kopija = self.labeledConnectedPores.copy()     # U malom broju slučaja (kada postoji pomak za samo jedan
                                                                                        # slice prije nego dođe do potrebe za vraćanjem) ako točka
                                    # skrene u stranu gdje nema puta može doći do zapinjanja
                                    # točke pa je potrebno očistiti oznake posjećenih ćelija
                                    # da proba opet naći put
                                    # print('Kopija se očistila')

                    self.svetocke.append(self.tocke)    # Točke jedne putanje se dodaju u listu svih putanja
                    self.zapis(self.tocke)  # Točke putanje se bilježe u .txt file
                    timerrute2 = time.perf_counter()
                    udaljenost_suma = 0

                    for i in range(len(self.tocke)-1):
                        udaljenost = math.sqrt((self.tocke[i][1] - self.tocke[i+1][1]) ** 2 + (self.tocke[i][2] - self.tocke[i+1][2]) ** 2 + (0 - 1) ** 2)
                        udaljenost_suma += udaljenost

                    self.svitau.append(udaljenost_suma / self.dim)  # Tau pojedine putanje se sprema u listu svih tau
                    print('{}/{} | {}'.format(j+1, len(self.sliceZ), (timerrute2-timerrute1)))

                timerkraj = time.perf_counter()
                print('Vrijeme do pronalaska svih ruta: {} sekundi'.format(timerkraj-timerpocetak))

                self.srednji_tau = sum(self.svitau) / len(self.svitau)  # Srednji tau dobiven aritmetičkom sredinom tau-a svih putanja
```

```
323.             return self.srednji_tau
324.
325.         def zapis(self, tocke):
326.
327.             """
328.
329.             Parameters
330.             ----------
331.             tocke : array
332.                 Lista svih točaka jedne putanje.
333.
334.             Returns
335.             -------
336.             .txt datoteka u koju je u svaki red zapisana
    putanja pojedine čestice od ulaza u jezgru do izlaska.
337.
338.             """
339.
340.             with open('koordinate_' + str(self.dim) + '.txt',
    'a') as f:
341.                 f.write(str(tocke))
342.                 f.write('\n')
343.
344.             return
345.
346.         def citanje(dim, putanja = None):
347.
348.             """
349.
350.             Parameters
351.             ----------
352.             dim : integer
353.                 Dimenzije matrice.
354.             putanja: string
355.                 Putanja do .txt datoteke, po defaultu je
    formata koordinate_dimenzijematrice.txt, npr.
    'koordinate_300.txt'.
356.
357.             Returns
358.             -------
359.             ucitano : array
360.                 Točke svih izračunatih putanja učitane iz .txt
    filea.
361.
362.             """
363.
364.             ucitano = []
```

```
365.             makni = '[]()\n,'
366.
367.             if putanja == None:
368.                 putanja = 'koordinate_' + str(dim) + '.txt'
369.
370.             with open(putanja, 'r') as f:
371.                 for i in f:
372.                     ucitano.append(i)
373.
374.                 for i in range(len(ucitano)):
375.                     for j in makni:
376.                         ucitano[i] = ucitano[i].replace(j, '')
377.
378.                     ucitano[i] = ucitano[i].split()
379.                     ucitano[i] = [int(i) for i in ucitano[i]]
380.                     ucitano[i] = [tuple(ucitano[i][j:j+3]) for j
    in range(0, len(ucitano[i]), 3)]
381.
382.             return ucitano
383.
384.         def svi_tort(putanje):
385.
386.             """
387.
388.             Parameters
389.             ----------
390.             putanje : array
391.                 Lista putanja iz .txt filea.
392.
393.             Returns
394.             -------
395.             svi_tau : array
396.                 Lista tortuoziteta za svaku putanju.
397.
398.             """
399.             svi_tau = []
400.
401.             for i in range(len(putanje)):
402.                 udaljenost_suma = 0
403.                 for j in range(len(putanje[0])-1):
404.                     udaljenost = math.sqrt((putanje[i][j][1] -
    putanje[i][j+1][1]) ** 2 + (putanje[i][j][2] - putanje[i][j+1][2])
    ** 2 + (0 - 1) ** 2)
405.                     udaljenost_suma += udaljenost
406.                 svi_tau.append(udaljenost_suma / 300)
407.
408.             return svi_tau
```

```
409.
410.         def avg_tort(putanje):
411.
412.             """
413.
414.             Parameters
415.             ----------
416.             putanje : array
417.                 Lista putanja iz .txt filea.
418.
419.             Returns
420.             -------
421.             tau_avg : float
422.                 Prosječni tortuozitet uzorka.
423.
424.             """
425.             svi_tau = Matrica.svi_tort(putanje)
426.             tau_avg = sum(svi_tau) / len(svi_tau)
427.
428.             return tau_avg
429.
430.     class Vizualizacija():
431.
432.         def povezane_tocke(dim, tocke):
433.             """
434.             Parameters
435.             ----------
436.             dim : integer
437.                 Dimenzije matrice.
438.             tocke : numpy array
439.                 Lista parova z, y, x koordinata.
440.
441.             Returns
442.             -------
443.             Graf povezanih točaka.
444.
445.             """
446.
447.             %matplotlib qt
448.
449.             plt.rcParams["figure.figsize"] = [10, 10]
450.             plt.rcParams["figure.autolayout"] = True
451.
452.             ax = plt.axes(projection='3d')
453.             plt.xlim([0, dim])
454.             plt.ylim([0, dim])
```

```
            ax.set_zlim([0, dim])
            plt.locator_params(nbins=10)
            ax.view_init(None, 225)

            ax.set_xlabel('$Z$', fontsize=20)
            ax.set_ylabel('$X$', fontsize=20)
            ax.set_zlabel('$Y$', fontsize=20)

            for i in range(len(tocke)):
                x, y, z = [], [], []
                for j in range(len(tocke[i])):
                    x.append(tocke[i][j][1])
                    y.append(tocke[i][j][2])
                    z.append(tocke[i][j][0])

                plt.plot(z, y, x, 'mediumblue', linestyle="-",
marker = 'None')


            plt.show()

            return

        def paraview (dim, lcp):

            """

            Parameters
            ----------
            dim : integer
                Dimenzije  matrice.
            lcp : numpy array
                3D matrica nula i jedinica gdje jedinice
predstavljaju porni prostor.

            Returns
            -------
            .vtr file za učitavanje u ParaView.

            """

            x = np.arange(0, dim+1)
            y = np.arange(0, dim+1)
            z = np.arange(0, dim+1)

            gridToVTK(str(dim), x, y, z, cellData = {str(dim):
lcp})
```

```
499.
500.             return
501.
502.         def pore(dim, lcp):
503.
504.             """
505.
506.             Parameters
507.             ----------
508.             dim : integer
509.                 Dimenzije  matrice.
510.             lcp : array
511.                 3D matrica nula i jedinica gdje jedinice
    predstavljaju porni prostor.
512.
513.             Returns
514.             -------
515.             3D graf pornog prostora.
516.
517.             """
518.             # %matplotlib qt
519.
520.             plt.rcParams["figure.figsize"] = [5, 5] # veličina
    prozora s grafom
521.             plt.rcParams["figure.autolayout"] = True
522.             fig = plt.figure(figsize=plt.figaspect(0.5)*1.5)
523.             ax = fig.add_subplot(111, projection='3d')
524.
525.             coords = np.where(lcp == 1)
526.             x = coords[2]
527.             y = coords[1]
528.             z = coords[0]
529.
530.             # ax.scatter3D(z, y, x, c = (x+y+z),
    cmap=plt.get_cmap('bone'), alpha = 1)
531.             ax.voxels(filled = lcp, facecolors = 'tab:blue',
    edgecolors = None)
532.
533.             plt.xlim([0, dim])  # duljina x osi
534.             plt.ylim([0, dim])  # duljina y osi
535.             ax.set_zlim([0, dim])   # duljina z osi
536.             plt.locator_params(nbins=10)    # "gustoća" osi
537.             ax.view_init(None, 225)
538.             ax.set_xlabel('$Z$', fontsize=20)
539.             ax.set_ylabel('$Y$', fontsize=20)
540.             ax.set_zlabel('$X$', fontsize=20)
541.             plt.show()
```

```
542.
543.             return
544.
545.         def histogram(tau):
546.
547.             """
548.
549.             Parameters
550.             ----------
551.             tau : array
552.                 Lista svih tortuoziteta za sve putanje.
553.
554.             Returns
555.             -------
556.             Crta histogram tortuoziteta.
557.
558.             """
559.
560.             fig, ax = plt.subplots(1, 1)
561.             ax.hist(tau, bins=20, color = 'olivedrab',
    edgecolor = 'black')
562.             ax.set_ylabel('Broj putanja')
563.             ax.set_xlabel('Tortuozitet')
564.
565.             return
566.
567.         def krivulja(tau):
568.
569.             """
570.
571.             Parameters
572.             ----------
573.             tau : array
574.                 Lista svih tortuoziteta za sve putanje.
575.
576.             Returns
577.             -------
578.             Crta krivulju tortuoziteta.
579.
580.             """
581.
582.             tau_sort = np.sort(tau)[::-1]
583.             x = np.linspace(1, 100, 11905)
584.
585.             fig, ax = plt.subplots(1, 1)
586.             ax.step(x, tau_sort, linewidth = 2, color =
    'olivedrab')
```

```
587.             ax.set_ylabel('Tortuozitet')
588.             ax.set_xlabel('Broj putanja, %')
589.             plt.xticks(np.linspace(0, 100, 11))
590.
591.             return
```

IZJAVA

Izjavljujem da sam ovaj rad izradio samostalno na temelju znanja stečenih na Rudarsko-geološko-naftnom fakultetu služeći se navedenom literaturom.

Robert Šimić

KLASA: 602-01/22-01/54
URBROJ: 251-70-12-22-2
U Zagrebu, 05.09.2022.

Robert Šimić, student

RJEŠENJE O ODOBRENJU TEME

Na temelju vašeg zahtjeva primljenog pod KLASOM 602-01/22-01/54, URBROJ: 251-70-12-22-1 od 19.04.2022. priopćujemo vam temu diplomskog rada koja glasi:

STOHASTIČKA PROCJENA TORTUOZITETA DIGITALNOG UZORKA STIJENE

Za mentora ovog diplomskog rada imenuje se u smislu Pravilnika o izradi i obrani diplomskog rada Prof.dr.sc. Domagoj Vulin nastavnik Rudarsko-geološko-naftnog-fakulteta Sveučilišta u Zagrebu.

Mentor:

(potpis)

Prof.dr.sc. Domagoj Vulin

(titula, ime i prezime)

Predsjednik povjerenstva za završne i diplomske ispite:

(potpis)

Izv.prof.dr.sc. Luka Perković

(titula, ime i prezime)

Prodekan za nastavu i studente:

(potpis)

Izv.prof.dr.sc. Borivoje Pašić

(titula, ime i prezime)