MATEMATIČNO-FIZIKALNI PRAKTIKUM 2021/22

2. naloga: Naključni sprehodi

Gregor Žunič, 28193076

Naključni sprehodi so vrsta gibanja, pri katerem v velikem številu korakov napredujemo iz izhodišča v neko končno lego, tako da se parametri vsakega naslednjega koraka sproti naključno določajo. Običajni zgled je Brownovo gibanje (difuzija) drobnih delcev barvila po mirujoči homogeni tekočini, kjer je spočetka barvilo zbrano v izhodišču. "Težišče" barvila $\langle x(t)\rangle$ v povprečju ostane v izhodišču, razen če v tekočini vzpostavimo kako anizotropijo (na primer v dveh razsežnostih z vsiljeno rotacijo). "Razmazanost" po dolgem času je sorazmerna s časom,

$$\sigma^2(t) \equiv \langle x^2(t)\rangle - \langle x(t)\rangle^2 = 2Dt\,.$$

Sorazmernostni koeficient je običajna difuzijska konstanta, priča smo normalni difuziji. Ta rezultat izhaja iz centralnega limitnega teorema (CLT), ki izraža, da je rezultantna porazdelitev končnih leg pri difuziji porazdeljena normalno (Gauss), če so le povprečni časi med koraki in povprečni kvadrati dolžin korakov končni.

Zanimiveje je opazovati naključne sprehode, pri katerih dovolimo nadpovprečno dolge korake. Verjetnostno gostoto porazdelitve po dolžinah posameznih korakov parametrizirajmo v potenčni obliki

$$p(l) \propto l^{-\mu}\,, \tag{1}$$

kjer naj bo $1 < \mu < 3$. Tedaj postane drugi moment porazdelitve

$$\langle l^2\rangle = \int l^2 p(l)\,\mathrm{d}l$$

neskončen. Govorimo o anomalni difuziji, prisotni pri celi družini kinematičnih distribucij dolžin poti z "debelimi repi".

Ustrezno sliko naključnega gibanja, povezanega s temi dolgimi koraki, lahko interpretiramo na dva načina:

- Lévyjev pobeg oz. polet (*flight*), implicira, da vsak korak iz porazdelitve (1) traja enako dolgo, medtem ko se hitrost gibanja med koraki (divje) spreminja.
- Lévyjev sprehod (*walk*), ki interpretira korak iz porazdelitve (1) kot gibanje s konstantno hitrostjo in tako koraki trajajo različno dolgo časa (dolžina koraka je sorazmerna s časom).

Slednja intepretacija bolj ustreza fizikalni sliki naključnega gibanja delca skozi snov, medtem ko se prva interpretacija uporablja v drugačnih aplikacijah.

Vse naloge lahko obravnavaš za obe interpretaciji, pobegov in sprehodov. V prvem primeru (pobeg, flight) je pretečeni čas direktno sorazmeren s številom korakov, v drugem primeru (sprehod, walk) pa je pretečeni čas sorazmeren z vsoto dolžine korakov.

Pri anomalni difuziji razmazanost (varianca) velike množice končnih leg naključnih Lévyjevih **sprehodov (walks)** narašča z drugačno potenco časa. Velja $\sigma^2(t) \sim t^\gamma$, kjer je

$$\begin{aligned}
1 < \mu < 2\,, &\qquad \gamma = 2 && \text{(balistični režim)}\,,\\
2 < \mu < 3\,, &\qquad \gamma = 4 - \mu && \text{(super-difuzivni režim)}\,,\\
\mu > 3\,, &\qquad \gamma = 1 && \text{(normalna difuzija)}\,.
\end{aligned}$$

Za $\mu = 2$ pričakujemo $\sigma^2(t) \sim t^2/\ln t$, za $\mu = 3$ pa $\sigma^2(t) \sim t \ln t$ (glej na primer [1] in druge reference prav tam).
Slika je nekoliko drugačna pri opazovanju naključnih Lévyjevih **poletov (flights)**. Spet vzamemo zvezo $\sigma^2(t) \sim t^\gamma$ in dobimo odvisnosti

$$1 < \mu < 3\,, \qquad \gamma = \frac{2}{\mu - 1} \qquad \text{(super-difuzivni režim)}\,,$$
$$\mu > 3\,, \qquad \gamma = 1 \qquad \text{(normalna difuzija)}\,.$$

Pri $\mu = 2$ očitno pričakujemo $\sigma^2(t) \sim t^2$, torej balistični režim.

Statistični komentar: v primerih, ko je drugi moment porazdelitve neskončen, bo tudi račun razmazanosti končnih leg x_n v obliki

$$\sigma^2 = \frac{1}{N-1} \sum_{n=1}^{N} (x_n - \langle x \rangle)^2 \tag{2}$$

divergiral oziroma bo imel ob ponovnih zagonih naključnega sprehoda močno raztresene vrednosti. Pomagaš si lahko na več načinov. širino porazdelitve končnih leg lahko oceniš tako, da prilagajaš Gaussovo krivuljo zgolj centralnega dela porazdelitve, tako da s prilagajanjem ne zajameš štrlečih (ne-Gaussovskih) repov. Lahko tudi neposredno računaš vsoto (2), a vanjo vključiš samo "razumne" člene (izpusti na primer nekaj odstotkov najmanjših in nekaj odstotkov največjih). Tretja možnost je, da definiramo novo vrsto variance

$$\sigma/N^p$$

in poiščemo tako potenco p, da ta spremenljivka konvergira za velike N (oz. velike t). še ena možnost je, da vzameš kako robustno mero za množico vrednosti X_i, na primer MAD, "median absolute deviation"

$$\mathrm{MAD} \equiv \mathrm{median}_i \left(|X_i - \mathrm{median}_j X_j| \right)\,.$$

Z njo merimo povprečje absolutne vrednosti deviacije na način, ki je zelo malo občutljiv na oddaljene vrednosti v repih porazdelitve, saj te vrednosti na račun mediane bistveno manj vplivajo kot na račun običajne povprečne vrednosti.

Naloga: Napravi računalniško simulacijo dvorazsežne naključne hoje za **polete in sprehode**. Začni vedno v izhodišču ($x = y = 0$), nato pa določi naslednjo lego tako, da naključno izbereš smer koraka in statistično neodvisno od te izbire še njegovo dolžino, torej

$$\begin{aligned} x &\leftarrow x + l\cos\varphi\,, \\ y &\leftarrow y + l\sin\varphi\,, \end{aligned}$$

kjer je φ enakomerno naključno porazdeljen po intervalu $[0, 2\pi]$, dolžina koraka l pa naj bo porazdeljena v skladu s potenčno obliko (Enačba 1). Dolžine l_i je v tem primeru potrebno generirati po verjetnostni porazdelitvi w(l)∼ p(l) (Enačba 1). Za izračun algoritma je osnova naslednja formula:

$$\int_a^l w(t)\,dt = \rho \cdot \int_a^b w(t)\,dt, \tag{3}$$

ki jo je potrebno rešiti in iz nje izraziti spremenljivko l. Tu je ρ (psevdo-)naključno število na intervalu $[0, 1]$ ter je $[a, b]$ relevantni interval vzorčenja. Za nekatere porazdelitve je izračun preprost, npr $w(t) = \frac{1}{\tau} e^{-\frac{t}{\tau}}$ nam da kar:

$$l = -\tau \ln(1 - \rho). \tag{4}$$

Dodatno pomoč za pretvorbo med verjetnostnimi porazdelitvami najdeš v gradivu v spletni učilnici ter na spletu.

Opomba: Korakaš lahko tudi v kartezičnem sistemu,

$$\begin{aligned} x &\leftarrow x + f_x(\rho_1)\,, \\ y &\leftarrow y + f_y(\rho_2)\,, \end{aligned}$$

kjer sta ρ_1 in ρ_2 naključni števili na intervalu $[0,1]$ ali $[-1/2,1/2]$, funkciji $f_{x,y}$ pa morata na koncu podati ustrezno porazdelitev po dolžinah poti glede na potenčno obliko (1), kjer lahko določiš zvezo med porazdelitvami po končnih legah x ali y ter po $l = \sqrt{x^2+y^2}$ z uporabo ustreznega Jacobijevega faktorja.

V vsakem primeru nariši nekaj značilnih slik sprehodov za 10, 100, 1000 in 10000 korakov. Iz velikega števila sprehodov z velikim številom korakov nato poskusi določiti eksponent γ za nekaj izbranih parametrov μ oziroma funkcij $f(x)$ v posameznih primerih ter presodi, za kakšno vrsto difuzije gre.

Dodatna naloga: Naključno spreminjaj še čas, ko delec pred naslednjim korakom miruje (s tako dodatno prostostno stopnjo poskušamo modelirati tako imenovani "sticking time" ali "trapping time" pri anomalni difuziji elektronov v amorfnih snoveh). Ustrezna verjetnostna gostota naj ima potenčno odvisnost

$$p(t) \propto t^{-\nu}\,,$$

kjer $1 < \nu < 2$. Je ta odvisnost razklopljena od porazdelitve osnovnega naključnega sprehoda po dolžinah (oziroma časih) posameznih korakov?

1 Reševanje

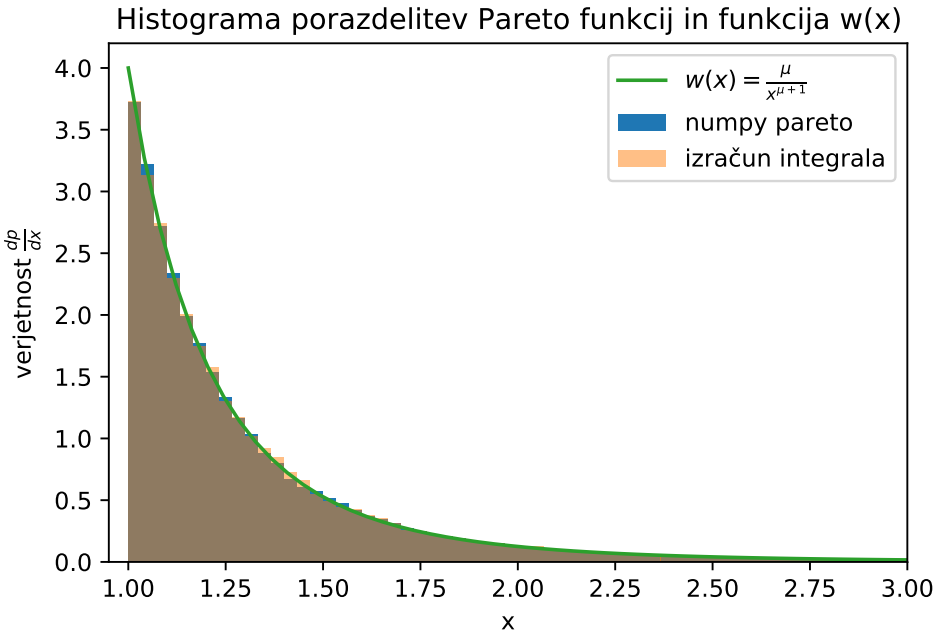
1.1 Generirajoče funkcije

Če hočemo dobiti generirajoče funkcije za naše dolžine l moramo najprej rešiti integral 3. Za poljubno integrabilno funkcijo w, ki gre v neskončnosti proti 0 (to potrebujemo, ker potem lahko zgornjo mejo b pošljemo proti ∞), se izraz preoblikuje v obliko

$$l = W^{-}(W(a)(1-\rho))$$

To lahko naredimo, ker je $W(\infty) = 0$. Odvisno od porazdelitve w, lahko tudi izberemo tak a, da se izračun bistveno pokrajša.

Za nalogo najbolj relevantna porazdelitev je $w(l) = l^{-\mu}$. Imenujemo jo tudi Paretova porazdelitev. Da bi izračunali svojo, moramo samo izračunati integral in inverz integrala funkcije w. Za primerjavo oziroma preverbo našega rezultata lahko izračunamo nekaj naključnih števil po porazdelitvi in iz njih narišemo histogram (graf 1). Seveda se histograma zelo dobro ujemata kar nakazuje na to, da sem pravilno izračunal Pareto porazdelitev. Od zdaj naprej bom kar uporabljal vgrajeno Pareto funkcijo iz knjižnice `Numpy`, ker za veliko število generiranih števil deluje občutno hitreje (C implementacija je hitrejša).



Slika 1: Primerjava histogramov moje implementacije in numpy funkcije z dejansko porazdelitveno funkcijo. Vidimo, da se vse tri funkcije zelo dobro ujemajo na celotnem območju. Graf je narisan z $\mu = 4$.

1.2 Ostale funkcije

1.3 Implementacija poletov in sprehodov

Kot že piše v navodilu, je precej lažje za implementacijo če se premaknemo v polarne koordinate ter za vsak korak generiramo poljuben kot in dolžino koraka. Iz tega vidika, je implementacija za polete lažja, ker je vsak korak zgolj generacija dveh spremenljivk in prištevek k trenutnim koordinatam.

Pri sprehodih je malce več dela, če nas zanima konstanten čas. Moja implementacija verjetno ni najbolj učinkovita direktno za najbolj preprost primer, vendar je zelo uporabna pri dodatni nalogi,

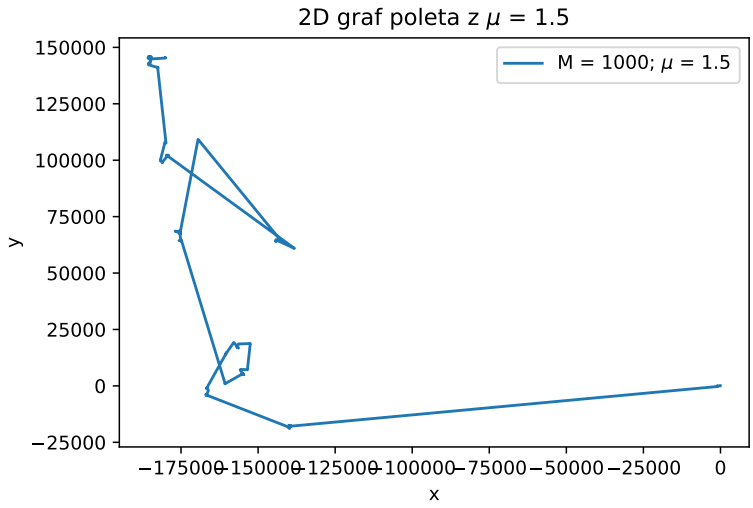
ker je potrebno dodati zgolj eno vrstico kode. Pri sprehodih moramo izbrati neko smiselno (ne preveč veliko) hitrost, ki pa vbistvu ni zares pomembna, ker nas tako ali tako zanima zgolj produkt vt_1, kjer je t_1 čas enega koraka. Velikost vt_1 ni pomembna, ker njena povečava pomeni zgolj razteg parametrov v enačbi. Izbral sem $vt_1 = 1,5$.

Najlažji pristop je sicer, da samo linearno interpoliramo premik med L_i ter L_{i+1}. To velja za nek m, kjer je $L_i < m \cdot vt_1 < L_{i+1}$.

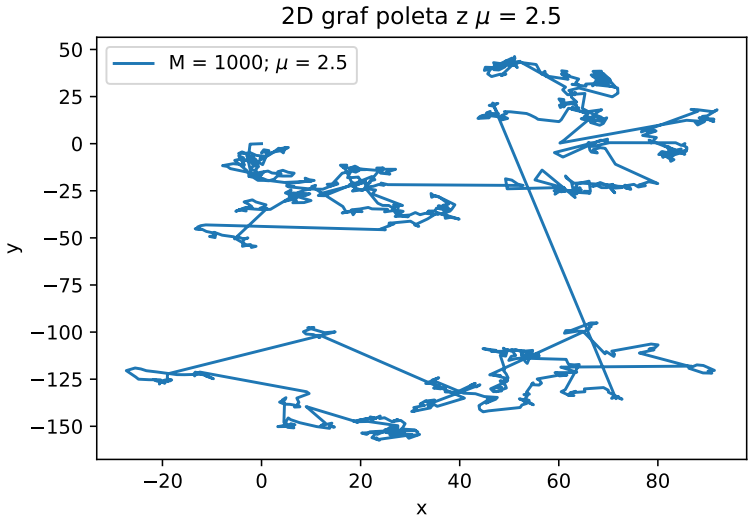
Moja implementacija je malce drugačna. Določim razdaljo vt_1, ki mora (delec) prepotovati v enem koraku. Nato se premikam po poti, dokler ne prepotuje vt_1. Koncept ni dosti bolj zakompliciran, vendar je treba upoštevati vse različne scenarije dolžin originalno generiranih (za polete) poti in njihovih kombinacij. Tako v principu delec v enem koraku prepotuje enako razdaljo, vendar vmes (lahko ni pa nujno) večkrat spremeni smer potovanja.

1.4 2D grafi premikanja

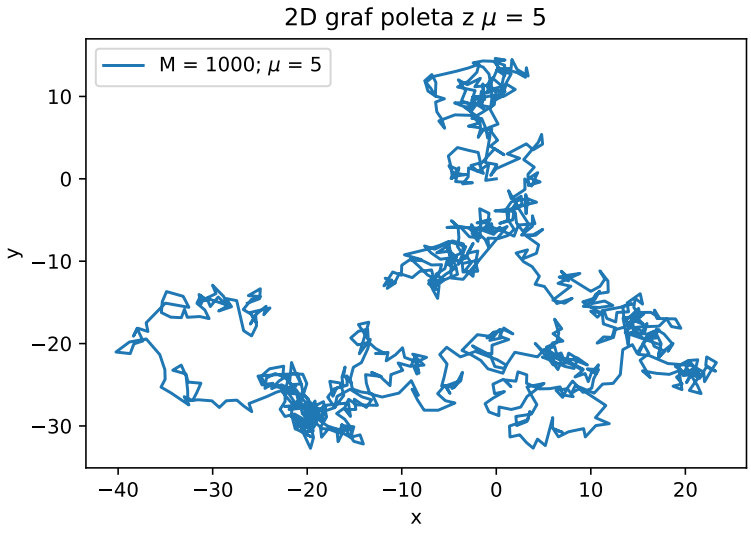
Sedaj lahko narišemo nekaj značilnih grafov premikanja za različne μ. Za primer 2D vizualizacije je v pricipu vseeno če narišemo polet ali sprehod. Krivulja po kateri potujeta je enaka, samo časovna parametrizacija je drugačna.



Slika 2: Graf poleta z $\mu = 1,5$. Vidimo, da pri majhnih vrednostnih čisto prevladujejo izredno dolge razdalje, malih pa se skoraj ne opazi. To je logično, saj verjetnostna gostota w ni dovolj blizu nič za velike razdalje. To pomeni, da bodo v vzorcu tudi zelo velike dolžine posameznih korakov.



Slika 3: Graf poleta z $\mu = 2,5$. Pri tem grafu, se že opazi, da razlika med najdaljšimi in najkrajšimi dolžinami ni več tako velika.



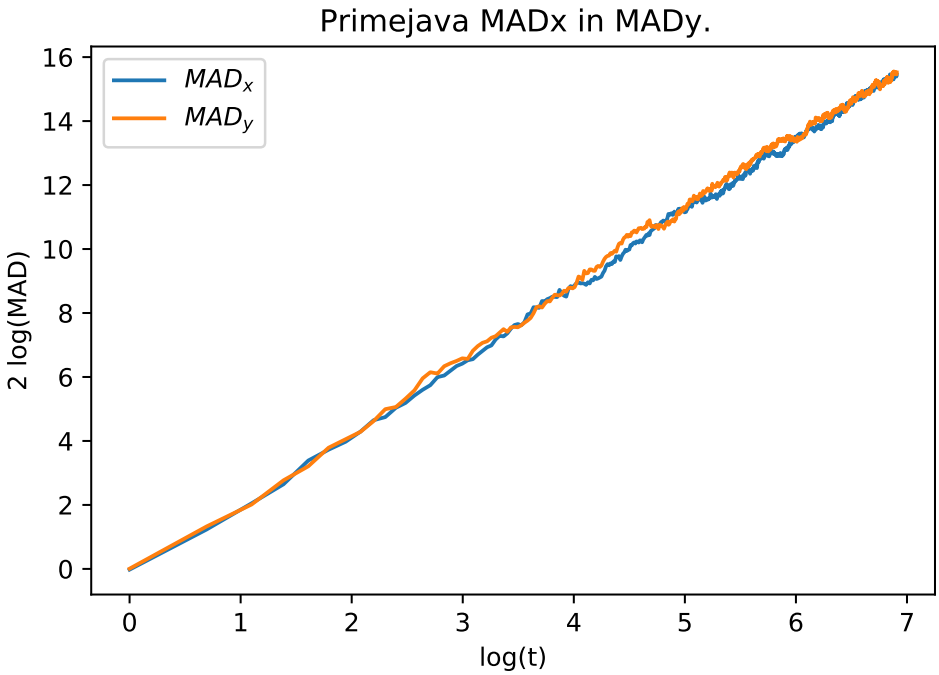
Slika 4: Graf poleta z $\mu = 5$. Pri večjih vrednostih μ grafi precej izgledajo precej bolj enakomerno porazdeljeni (ker je vzorec dolžin bolj zgneten v manjše vrednosti) Tudi skala premikanja se precej zmanjša.

1.5 Vzporedni poskusi in izračun drugega momenta

Ker je drugi moment porazdelitve $l^{-\mu}$ neskončen, bo vsota za izračun sigme slabo konvergirala. Zato je precej boljša alternativa MAD, ki ni toliko občutljiv na zelo oddaljene vrednosti. Vendar kako slab je zares izračun σ po definiciji?

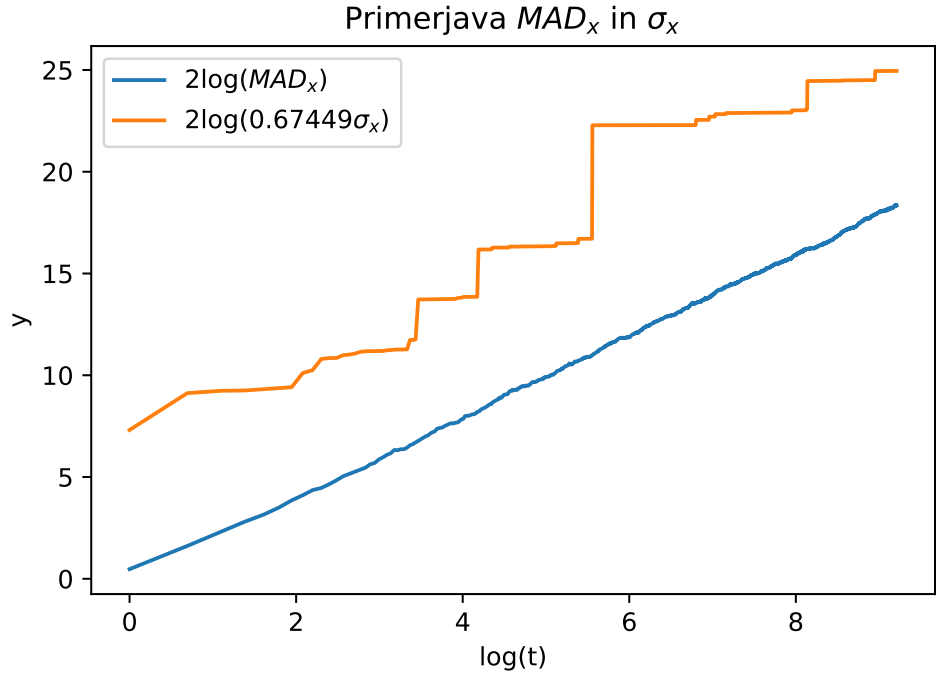
V obeh primerih poskus izvedemo tako, da vzporedno izvajamo 1000 poskusov in ob vsakem času izračunamo MAD ali σ na vseh poskusih.

Primerjavo lahko naredimo zgolj v eni dimenziji ali pa ju kar kvadratno seštejem (to storim pri izračunih), ker sta σ_x in σ_y za velike čase skoraj enaka. To lahko argumentiram s tem, da smer gibanja za vsak korak poljubna. To prikazuje graf 5.



Slika 5: Vidimo, da sta si MAD_x in MAD_y zelo blizu. Graf je narisan za polet, $\mu = 3$.

Iz tega znanja, lahko sedaj narišem graf 6, ki primerja izračun drugega momenta po definiciji (**??**) ter z MAD. Iz grafa je očitno, da je MAD superiorna metoda izračuna drugega momenta.



Slika 6: Vidimo, da sta si MAD_x in σ_x zelo različni, poleg tega, pa vrednost σ_x močno skače. Graf je narisan za polet, $\mu = 3$.

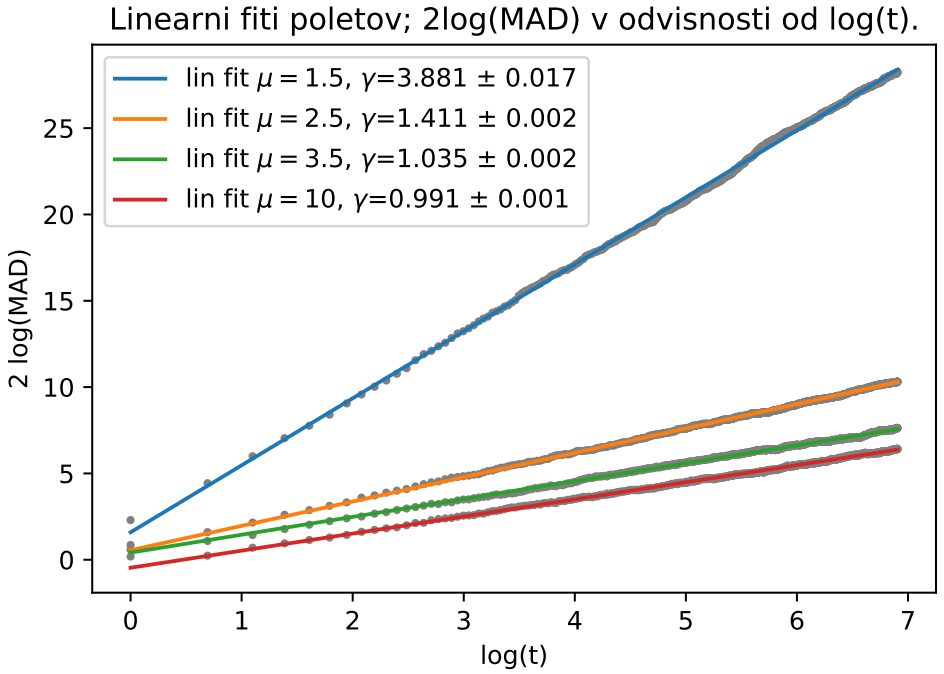
1.6 Linearni fit in izračun γ

Sedaj, ko imamo vso orodje, lahko izračunamo γ za poljubne μ za polete in sprehode.

Ko narišemo točke vrednosti MADx za posamezne korake, jih lahko povežemo z premico, če logaritmiramo obe osi. Torej na graf zares narišemo $2\log(MAD)$ na ordinato in $\log(t)$ na absciso, kjer je t zares kar enak številu korakov.

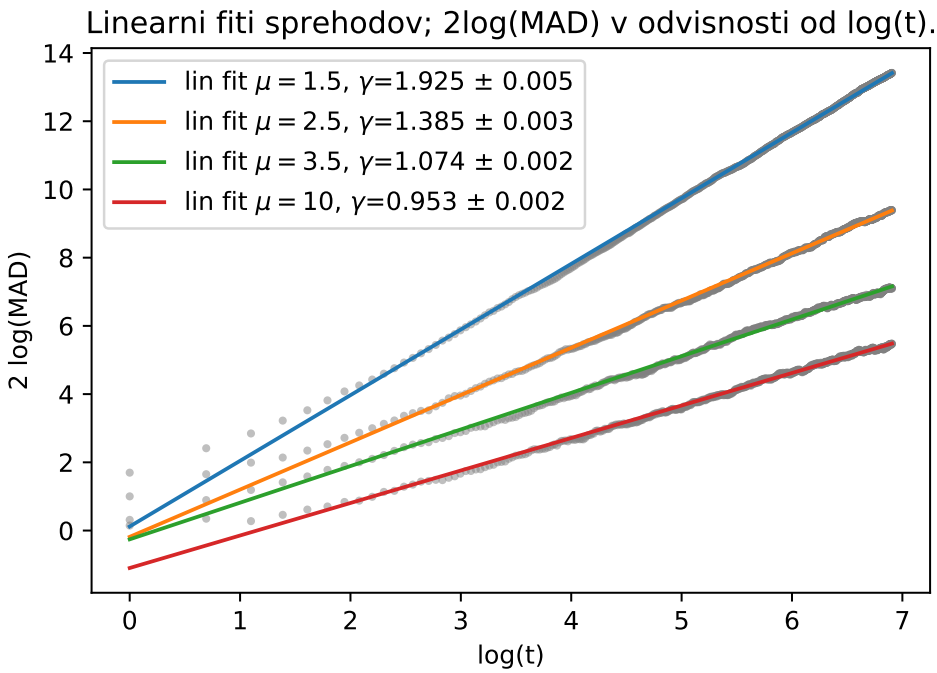
Za izračun premic uporabim linearno regresijo s pomočjo funkcije `curve_fit` iz knjižnice `Scipy`. Metoda nam že sama izračuna napako, oziroma še bolj natančno korelacijsko matriko, iz katere lahko izračunamo absolutne napake, če vzamemo diagonalne elemente in jih korenimo, kar nam da relativno napako, nato pa jih pomnožimo z vrednostjo funkcije, da dobimo absolutno napako.

Najprej lahko narišemo graf 8 za polete. Če bi vzporedno delal več meritev, bi bile napake za σ še manjše. Opazimo lahko tudi to, da so koeficienta linearne regresije γ zelo natančni oziroma je napaka relativno majhna.



Slika 7: Linearne regresije na vrednostih MAD(t) za polete. Opazimo, da so meritve precej linearne.

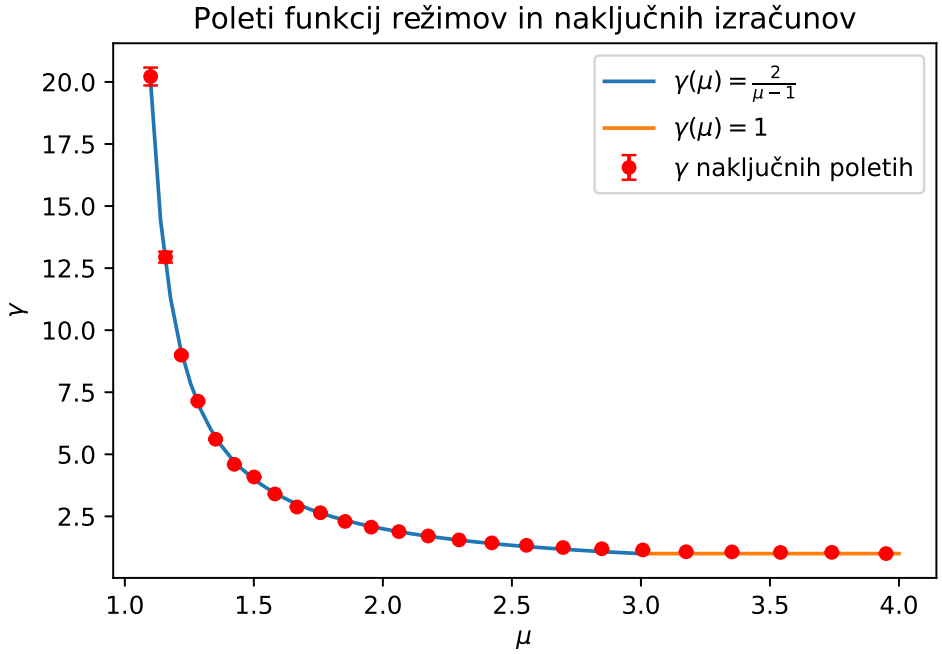
Zdaj še narišimo graf z enakimi parametri za sprehod. Takoj lahko opazimo, da so nakloni oziroma koeficienta linearne regresije precej manjši, kot pri poletih. To je seveda lahko razložiti iz same definicije sprehoda. Namreč pri poletih lahko v zelo majhnem času delec prepotuje zelo veliko razdaljo, pri sprehodih pa smo omejeni z maksimalno hitrostjo. Iz tega lahko tudi predvidimo, da bo γ pri majhnih vrednostih pri poletih divergiral, pri sprehodih, pa bo lepo konvergiral proti neki določeni vrednosti (izkaže se da 2).



Slika 8: Linearne regresije na vrednostih MAD(t) za sprehode.

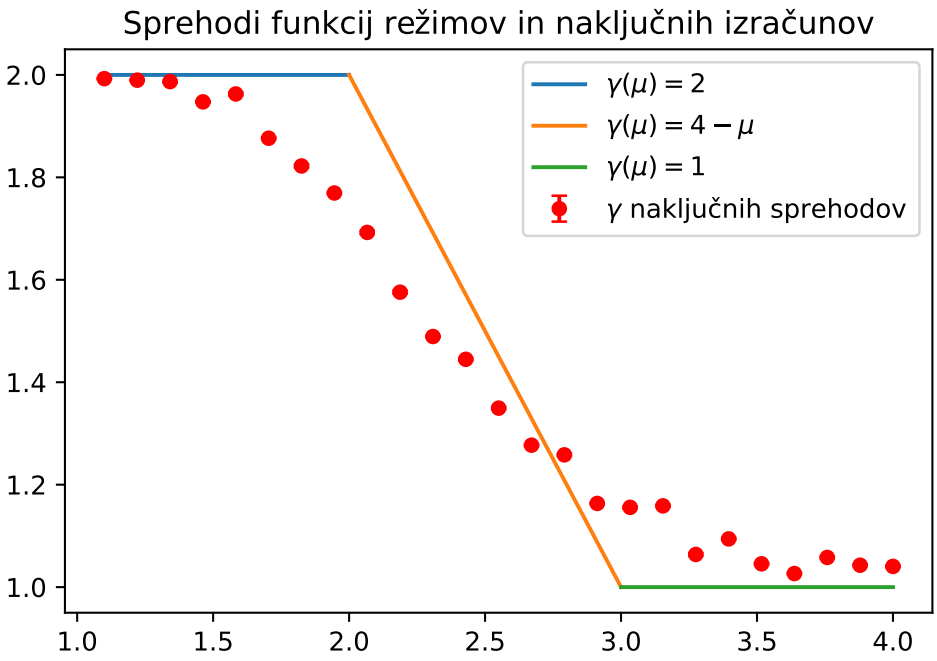
1.7 Primerjava koeficientov z modelom

Ko izračunamo koeficiente γ za različne μ, lahko te vrednosti primerjamo z opisom na prvi strani.



Slika 9: Primerjava koeficientov γ s teoretično vrednostjo, za različna režime. Do $\mu < 3$ velja super-difuzni način, kasneje pa normalna difuzija.

Če komentiramo graf 9, je precej očitno, da se meritve super prilegajo modelu.

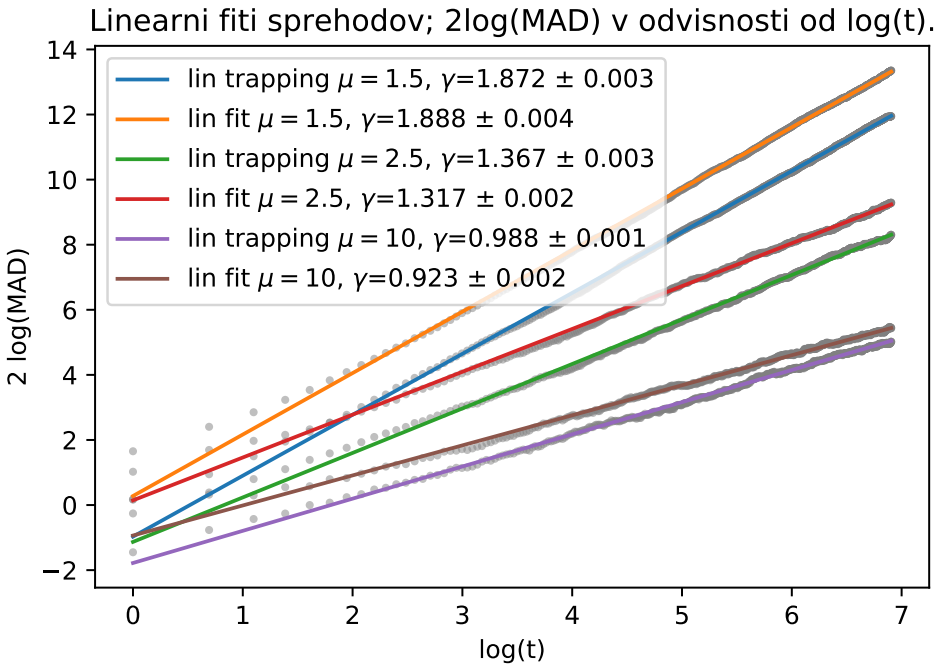


Slika 10: Primerjava koeficientov γ s teoretično vrednostjo, za različna režime. Do $\mu < 2$ velja balističnih režim, do $\mu < 3$ velja super-difuzivni režim, kasneje pa smo v območju normalne difuzije.

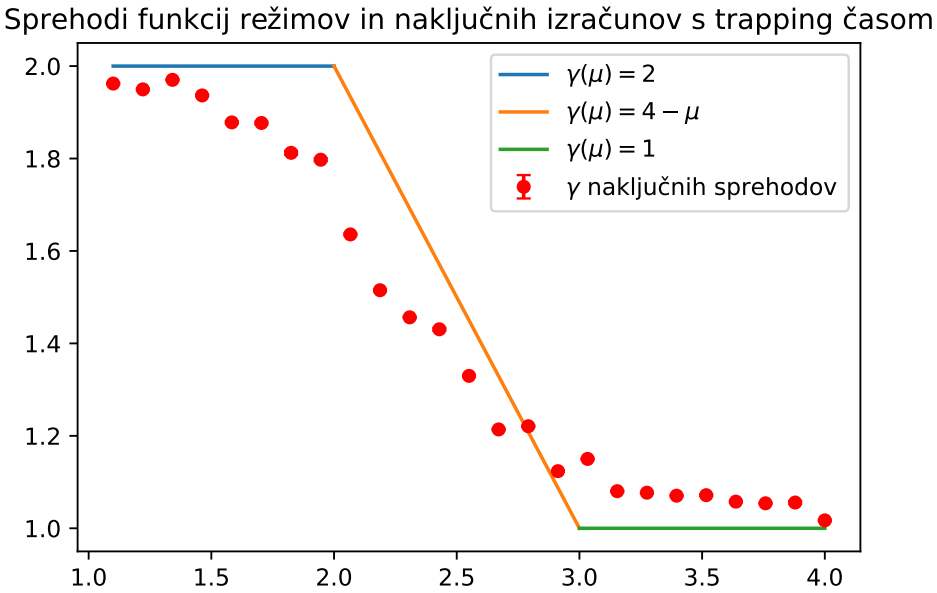
Kot po pričakovanjih, vrednost γ konvergira proti 2. Splošna oblika je načeloma korektna, saj je na začetku in na koncu območij obnašanje konstantno, vmes pa imamo območje linearnega padanja. Ne prilega se modelu do potankosti.

1.8 "Trapping time"

Za izračun dodatne naloge, je vse kar morem storiti v moji kodi, dodati navadni kodi pri sprehodu to, da se vt_1 naključno odšteje vrednost po Paretovi porazdelitvi. Kar dobimo, da zelo podobno navadnemu sprehodu s tem, da je širjenje malenkost počasnejše.



Slika 11: Vidimo, da je model s trapping časom malenkost zelo podoben navadnemu modelu sprehoda, kar nakazuje, da lahko čase mirovanja skoraj zanemarimo.



Slika 12: Vidimo, da je tudi v tem primeru primerjava z modelom zelo blizu, s tem da so vse vrednosti γ premaknjene malenkost navzdol.

2 Komentar

Metode, ki sem jih uporabil so privedle do precej smiselnih rezultatov. Iz tega lahko sklepam, da so metode dokaj pravilne. Pomembna se mi zdi tudi ugotovitev, da je σ zares tako nepredvidljiva v primerjavi z MAD.

Literatura

[1] E. R. Weeks, J. S. Urbach, H. L. Swinney, Physica D **97** (1996) 291.