

2. naloga: Naključni sprehodi

Samo Krejan, 28231092

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 rezultatna 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

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

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

$$\langle l^2 \rangle = \int l^2 p(l) dl$$

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 obravnavas 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

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

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

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

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

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

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”

$$\text{MAD} \equiv \text{median}_i (|X_i - \text{median}_j X_j|) .$$

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.

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

2 Simulacija sprehodov in letov

Predno se lotimo računanja dejanskih količin, bi radi dobili občutek, kako se naključni sprehodi in poleti sploh obnašajo.

Naključne korake, sem generiral kot je opisano v navodilih naloge. Tu sem si pomagal z *numpy* vgrajeno Paretovo porazdelitvijo:

$$p(x) = \frac{\alpha x_m^\alpha}{x^{\alpha+1}}$$

kjer je x_m minimalni možni izid, ki sem ga za simulacije vedno nastavil na 1, α pa je ekvivalentna $\mu - 1$.

Uporabi vgrajenega Pareta bi se lahko izognil na sledeč način; najprej zapišemo kumulativno porazdelitev Pareta:

$$P(x) = 1 - \left(\frac{x_m}{x}\right)^\alpha$$

za tem lahko uporabimo inverzno transformacijsko metodo. Če najprej generiramo števila y iz uniforme porazdelitve na intervalu $[0, 1]$ lahko izrazimo inverzno transformacijo:

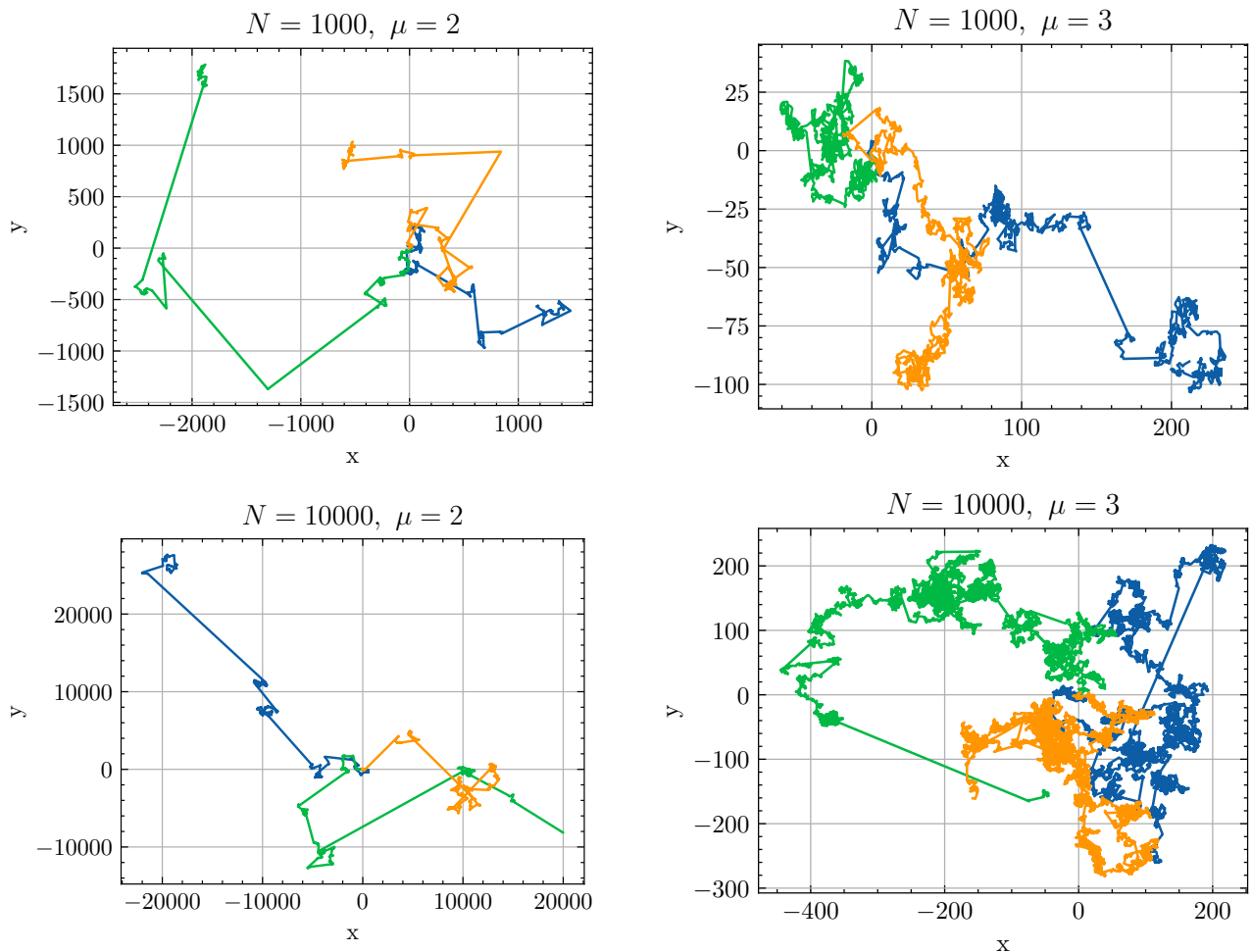
$$x = x_m \left(\frac{1}{1-y}\right)^{1/\alpha}$$

ki sledi Paretovi porazdelitvi, lahko pa tudi direktno izrazimo porazdelitev dolžin kot:

$$l = (1-y)^{-\frac{1}{\mu-1}}$$

Na koncu se nisem odločil za implementacijo tega v pythonu, saj sem stremel k čim bolj časovno optimiziranemu algoritmu in prepričan sem, da je *numpy* implementacija precej bolj optimizirana kot katerakoli koda, ki jo bom kadarkoli napisal.

S tem problemom razčiščenim se sedaj lahko lotimo risati nekaj različnih sprehodov, da dobimo nekaj občutka za situacijo. Narišemo si sprehode za 10, 100, 1000 in 10 000 korakov s parametrom $\mu = 2$ in $\mu = 3$. Pri majhnih številih še ne opazimo kakih znatnih razlik, pri večjih dveh pa opazimo znatne razlike.



Slika 1: 3 različni sprehodi za različno število korakov in različen parameter μ

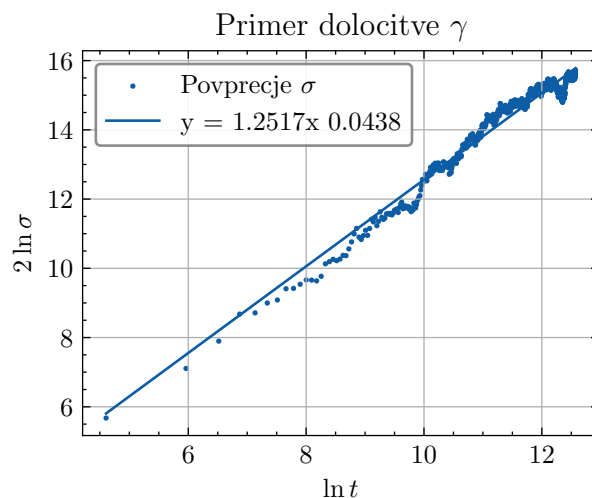
Iz slike 1 takoj opazimo, da parameter μ močno vpliva na obnašanje sprehodov, že sama skala na grafih mora biti dva velikostna reda večja pri $\mu = 2$ da cel sprehod sploh lahko predstavimo. Kasneje bomo videli, da je temu tako ker sta $\mu = 2$ in $\mu = 3$ v različnih režimih, tako kot smo to napovedali že v uvodu.

2.1 Določanje režimov

V katerem režimu se nahajamo nam bo povedal faktor γ opisan v uvodu. Le tega določimo z linearno regresijo:

$$2 * \ln \sigma = \gamma \ln t + c$$

Za izvajanje regresije sem uporabil *numpy.polyfit* funkcijo, ki poleg parametrov *fita* določi tudi njihovo napako.



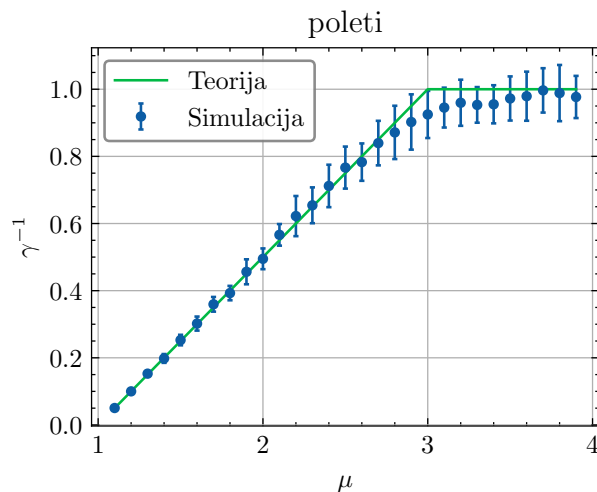
Slika 2: Faktor γ določimo z linearno regresijo. Na sliki je primer regresije za sprehod s parametrom $\mu = 2.5$, z 10 000 koraki in 100 različnimi sprehodi

Regresijo ponovimo za nekaj različnih μ tako za naključne sprehode kot tudi za naključne polete in na ta način dobimo grafe odvisnosti γ od μ . Najprej to naredimo za naključne polete, kjer najprej teoretično odvisnost opisano v odvodu lineariziramo in tako na graf 3 namesto γ rišemo γ^{-1}

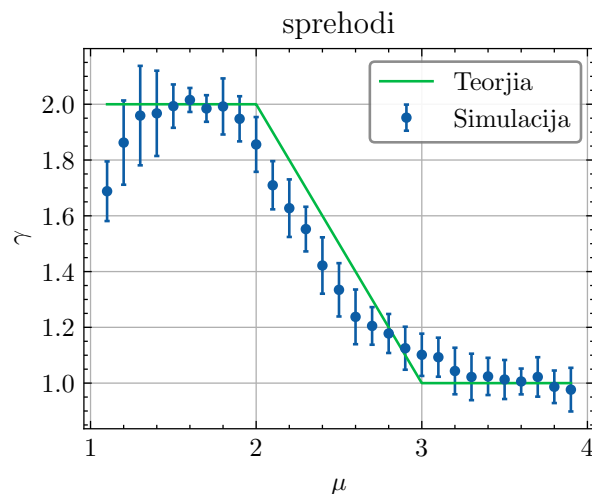
Že pri poletih vidimo, da teorija deluje, saj se praktično vsi podatki simulacij skladajo s teorijo v okviru napake. Kljub temu lahko opazimo trend, da okoli točke $\mu = 3$ simulacije najbolj odstopajo. Čeprav nam to morda ni všeč je lahko pojasniljivo. Tam namreč pride do faznega prehoda med super difuzivnim režimom ($\mu < 3$) in normalno difuzijo ($\mu > 3$). Ker simulacija lahko poteka zgolj s končnim številom sprehodov s končnimi koraki, bo seveda ta prehod manj oster.

Vajo ponovimo tudi z naključnimi sprehodi. Te so nekoliko bolj kompleksni za generirati in obdelati, saj časovni intervali med posameznimi koraki niso konstantni, pač pa so sorazmerni z dolžino posameznega koraka. Ker si seveda želimo primerjati mnogo različnih sprehodov ob enakomerno razmaknjenih časih, moramo pozicije ob teh v naprej določenih časih interpolirati iz časa pred in po določenem času. Na srečo za to že obstaja funkcija *np.interp*, saj če bi interpolacije implementirali v pythonu bi nam samo ta operacija vzela cel dan.

Ko pa je ta problem razrešen, je zgodba enaka kot pri naključnih poletih, s to izjemo, da nam ni treba obračati γ faktorja. Odvisnost vidimo na grafu 4



Slika 3: γ^{-1} v odvisnosti od μ (20 vzorcev z 100 sprehodi z 10 000 koraki). Skoraj vse simulacije se v intervalu napake skladajo s teoretičnimi napovedmi.



Slika 4: γ v odvisnosti od μ (20 vzorcev z 100 sprehodi z 10 000 koraki). Simulacije so skoraj v okviru napake. Največje odstopanje opazimo pri μ blizu 1 in v režimu $2 < \mu < 3$

Tudi pri sprehodih opazimo, da teorija deluje precej dobro. Imamo sicer nekoliko manjše ujemanje med simulacijami in teorijo, a vseeno lahko precej dobro ločimo med tremi režimi: pri $\mu < 2$ (balistični difuziji) je ujemanje zelo dobro razen pri μ blizu 1. Ugibam, da imamo na tisti točki nekaj problemov z generacijo naključnih dolžin korakov.

Literatura

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