

Univerza v Ljubljani  
Fakulteta za matematiko in fiziko



# Zaključna naloga iz teorije dinamičnih sistemov

Avtor  
**Andrej Kolar - Požun**

5. November, 2018

# Kazalo

<b>1 Naloga</b>	<b>1</b>
<b>2 Priprava biljarda</b>	<b>1</b>
2.1 Analiza členov vrste . . . . .	1
2.2 Primeri biljardov . . . . .	2
<b>3 Gibanje delca</b>	<b>4</b>
3.1 Kratek opis algoritma . . . . .	4
3.2 Simulacija gibanja . . . . .	5
<b>4 Difuzija x koordinate</b>	<b>8</b>
<b>5 Prehod v globalni kaos</b>	<b>11</b>
<b>6 Kriterij Čirikova</b>	<b>13</b>

# 1 Naloga

Študiraj transport točkastih prožnih delcev v biljardu, ki ima obliko hrapavega neskončno dolgega kanala. Ena od sten naj bo ravna  $y = 0$ , druga pa naj bo parametrizirana kot

$$y(x) = a + \sum_{n=1}^{\infty} a_n \cos nx + b_n \sin nx$$

$a_n$  in  $b_n$  naj bodo Gaussovo porazdeljena neodvisna slučajna števila z variacijo  $\langle a_n^2 \rangle = \langle b_n^2 \rangle = \varepsilon^2/n^2$ . Razišči možnost difuzije koordinate vz dolžne, x-komponente hitrosti ali položaja delca v biljardu po dolgem času, t.j. računaj  $\langle (p_x(t) - p_x(0))^2 \rangle$  kot funkcijo časa  $t$ , kjer  $\langle \rangle$  predstavlja povprečenje po velikem številu začetnih pogojev in/ali realizacij hrapavosti, in preveri možnost normalne ali anomalne difuzije ( $2Dt$  ali morda celo  $2Dt^\alpha$ ). Kako je difuzijska konstanta D odvisna od  $\varepsilon$ ? Ali obstaja prag  $\varepsilon_c$  tako, da za  $\varepsilon > \varepsilon_c$  dobimo globalni kaos v faznem prostoru (zlom vseh KAM torusov).  $\varepsilon_c$  morda lahko poskusit oceniti po kriteriju Čirikova?

Tekom naloge bom uporabljal programerski jezik Python skupaj z numpy/scipy knjižnicami.

## 2 Priprava biljarda

### 2.1 Analiza členov vrste

V funkciji, ki opisuje zgornjo mejo je neskončna vrsta. Seštevati neskončno členov ne moremo in bo torej treba vsoto nekje odrezati. V povprečju bodo členi z višjim n manjši, saj bodo konstante  $a_n, b_n$  porazdeljene ožje okoli ničle. Odločiti se moramo, kdaj bo nastopil prvi zanemarljivo majhen člen.

Recimo, da seštejemo le  $N$  členov in zgornjo steno biljarda aproksimiramo z funkcijo.

$$y_N(x) = a + \sum_{n=1}^N a_n \cos nx + b_n \sin nx$$

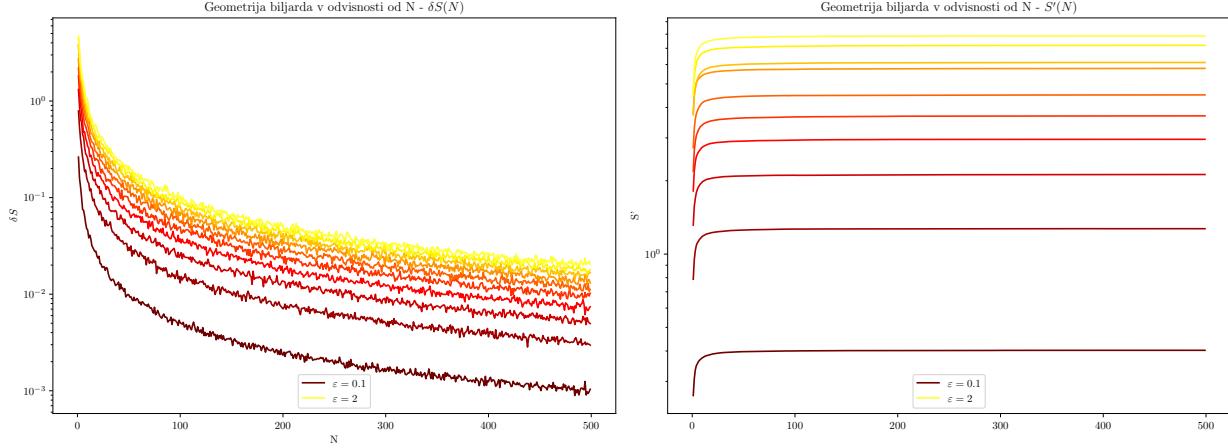
Dodatni prispevek k steni, ki bi ga dobili, če bi v vsoti sešteli še en člen več je

$$\delta y_N(x) = a_{N+1} \cos((N+1)x) + b_{N+1} \sin((N+1)x)$$

Zanima nas, koliko  $\delta y(x)_N$  zares spremeni kanal v odvisnosti od  $N$ . Opazovali bomo količini

$$\begin{aligned}\delta S(N) &= \int_0^{2\pi} |\delta y_N(x)| dx \\ S'(N) &= \int_0^{2\pi} |y_N(x)| dx\end{aligned}$$

Integrala smo aproksimirali z vsoto (trapezna formula).

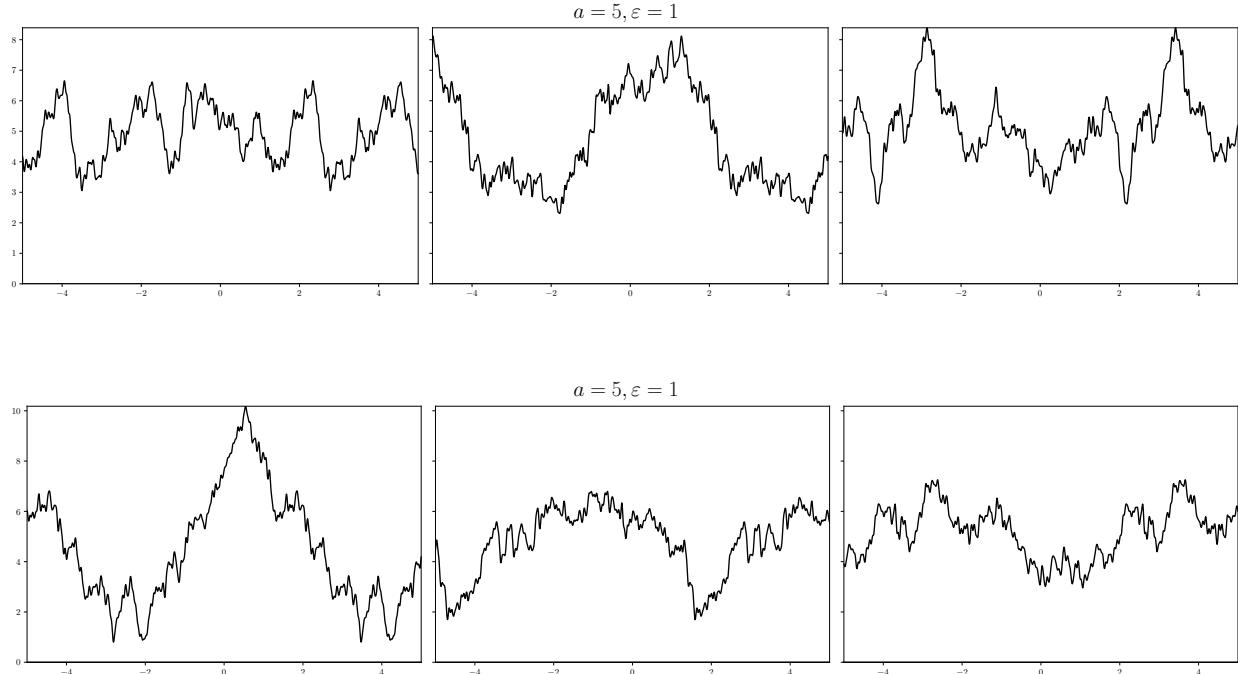


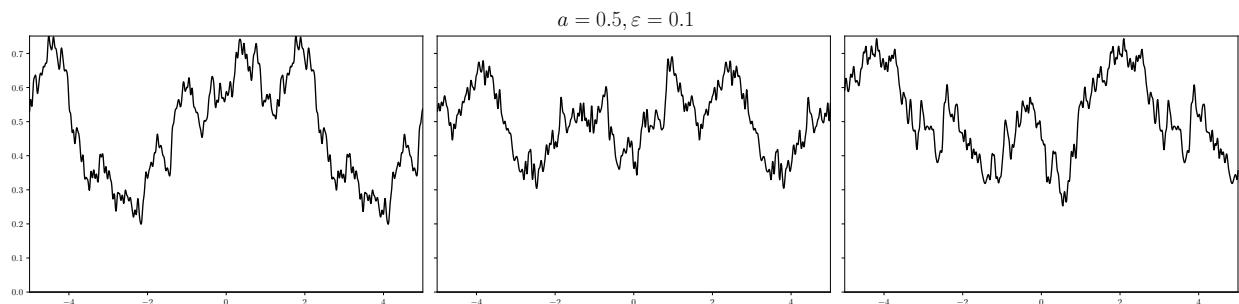
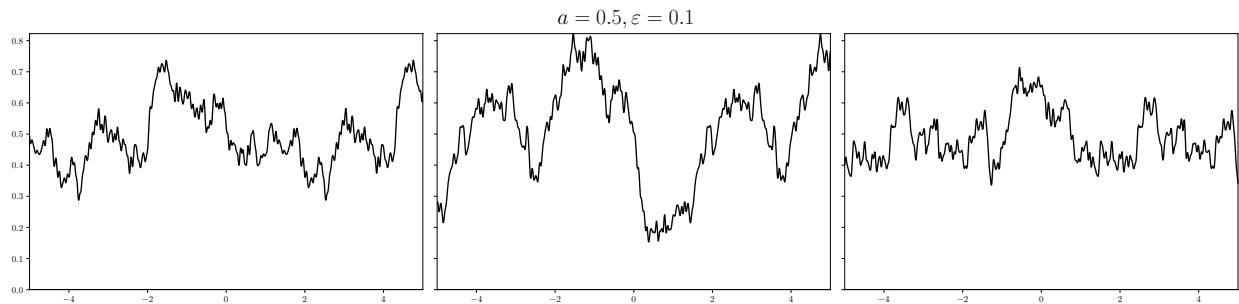
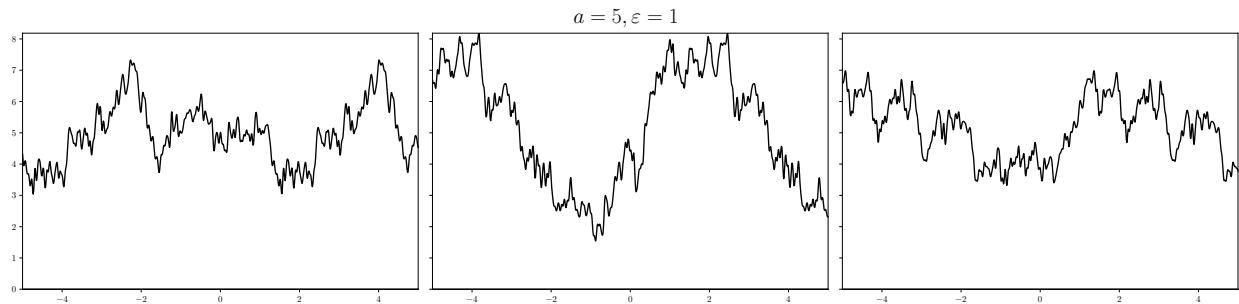
Na grafu so prikazani  $\delta S$  (leva y os) in  $S'$ (desna os) za več vrednosti parametra  $\varepsilon$ . Po tem grafu sodeč ne bomo naredili prevelike napake, če nehamo seštevati pri  $N=100$ (ne glede na  $\varepsilon$ ) kar se dodane ploščine krivulje tiče. Ker pa sumimo, da bo sistem kaotičen, nam lahko tudi majhne spremembe v obliki stene znantno vplivajo na rezultate. Zaenkrat bom simulacije izvajal pri  $N=100$ , ko bom izračunal konkretne količine (npr. difuzijsko konstanto), pa se bom vrnil k odvisnosti od  $N$  in preveril, ali smo prezgodaj odrezali vsoto.

V priloženi animaciji Stena.mp4 je prikazano spremenjanje oblike stene, ko seštevamo vedno več členov. Tu tudi vidimo, da se stena veliko bolj spreminja, ko je  $N$  še krepko pod 100.

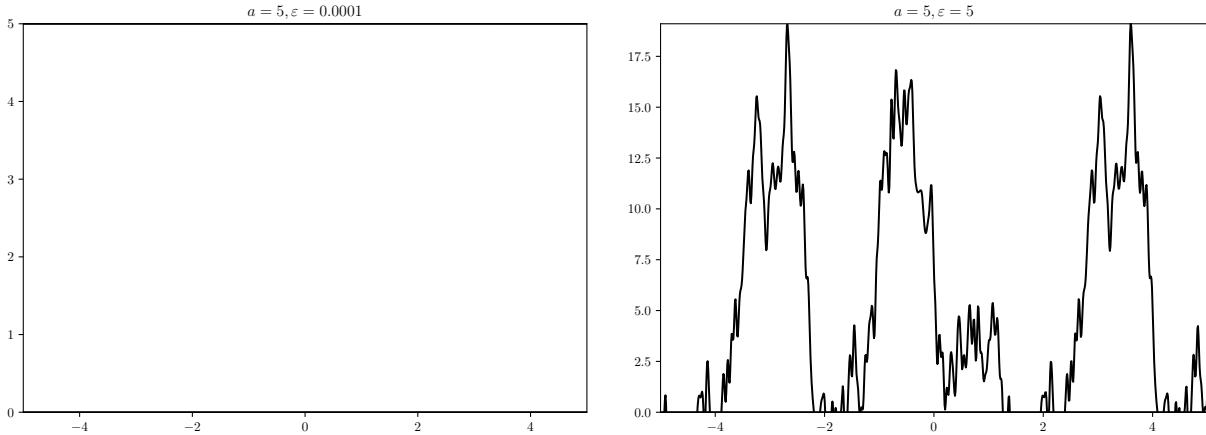
## 2.2 Primeri biljardov

Poglejmo si kako tipično izgleda naš biljard:





Opazimo, da ima zid z različnim epsilonom še vedno enako obliko, le da je reskaliran.



Poglejmo še limitna primera: Na levi je epsilon veliko manjši napram a, kar generira skoraj raven kanal. Na desnem grafu je epsilon primerjiv z a, kar povzroči, da so členi v vrsti dovolj visoki, da se kanal zapre in nimamo več neskončno dolgega biljarda.

Očitno je tudi, da je biljard periodičen (Torej lahko na naš problem gledamo kot končen biljard s periodičnimi robnimi pogoji). Namreč v vsoti nastopajo členi oblike  $\sin nx, \cos nx$ , katerih perioda je  $2\pi/n$ , kar za vsako naravno število n deli  $2\pi$  kar je torej tudi celotna perioda zgornje stene biljarda.

### 3 Gibanje delca

#### 3.1 Kratek opis algoritma

Simulacija gibanja delca deluje po naslednji psevdokodi:

```

r, v = zacetniPogoji()
t = 0
while t < KončniČas do
    y = zgornjaStena(r)
    if trk() then
        rotirajHitrost()
    end if
    r = r + vΔt
    t = t + Δt
end while
```

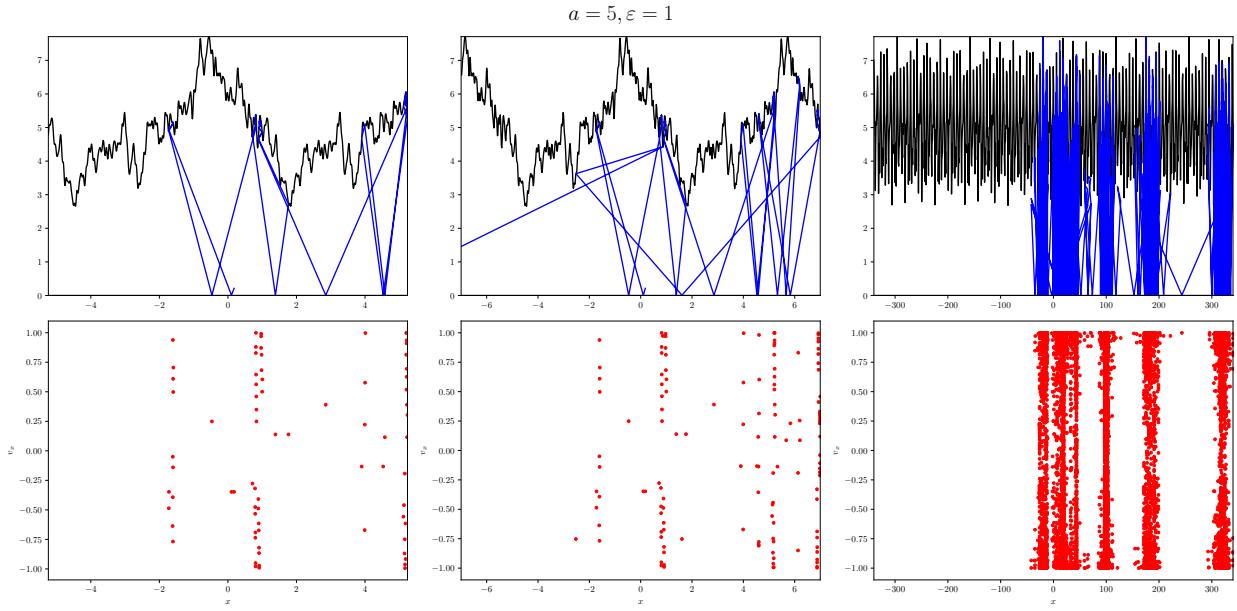
- Funkcija `zacetniPogoji()` generira naključne zacetne pogoje delca in parametre zgornje stene. Vrne vektorja `r` in `v`, ki predstavlja pozicijo in hitrost delca.
- Funkcija `zgornjaStena` poda `y` koordinato stene pri izbranem `x`.
- Funkcija `trk()` na vsakem koraku preveri, ali je prišlo do trka delca z eno izmed sten biljarda. Izberemo si določen  $\epsilon$  (Vedno sem izbral take, da je enak dolžini koraka - 0.01) in preverim, ali je delec v `y` smeri od stene oddaljen za manj kot  $\epsilon$ . Če je, se je zgodil trk. Težavo predstavljajo skoraj navpični deli zgornje stene biljarda, pri katerih zelo težko zaznamo trk po tem postopku. Ena izmed rešitev je zmanjševanje časovnega koraka, ki ni praktična, saj simulacija potem teče predolgo. Namesto tega raje na vsakem koraku v funkciji `trk()` preverim še, če se trenutno nahajam nad zgornjo steno biljarda. Če se, grem korak nazaj in s petimi iteracijami bisekcije poiščem natančno mesto trka.
- Funkcija `rotirajHitrost()` loči dva primera: Če je mesto trka na spodnji steni biljarda, enostavno spremenim predznak `y` komponente hitrosti. Če je na zgornji steni, pa poračunam kot med vektorjem hitrosti in normalo na zgornjo steno na mestu trka. Naklon premice vzporedne z vektorjem hitrosti

namreč imam, tangentno in normalno premico na mesto trka pa dobim iz funkcije zgornje stene  $y(x)_N$ , katere odvod je preprost, saj v njej nastopajo vsote kotnih funkcij. Odbojni zakon pravi, da morata biti vpadni in izpadni kot enaka, torej vektor hitrosti enostavno rotiram za  $2\pi - 2^*$  kot med hitrostjo in normalo.

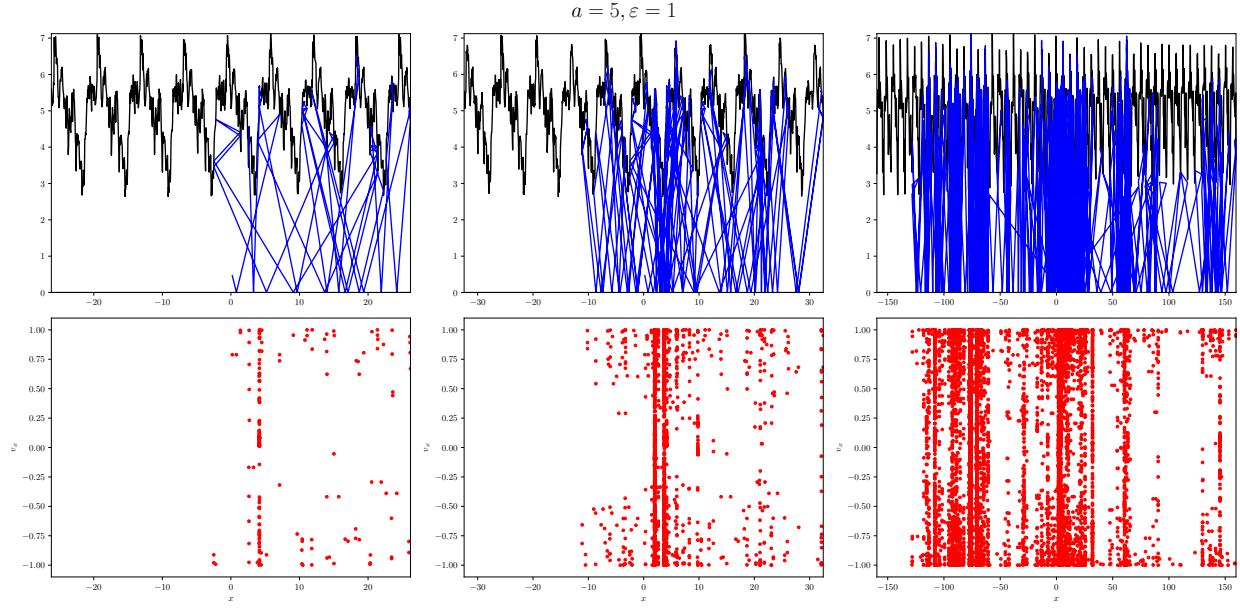
### 3.2 Simulacija gibanja

Najprej na hitro pomislimo kaj pričakujemo. Za majhne epsilonove, je zgornja stena skoraj ravna in bo gibanje dokaj regularno. Pri večanjem epsilona ima zgornja stena vedno bolj divjo obliko, gibanje zato postaja vedno bolj nepredvidljivo.

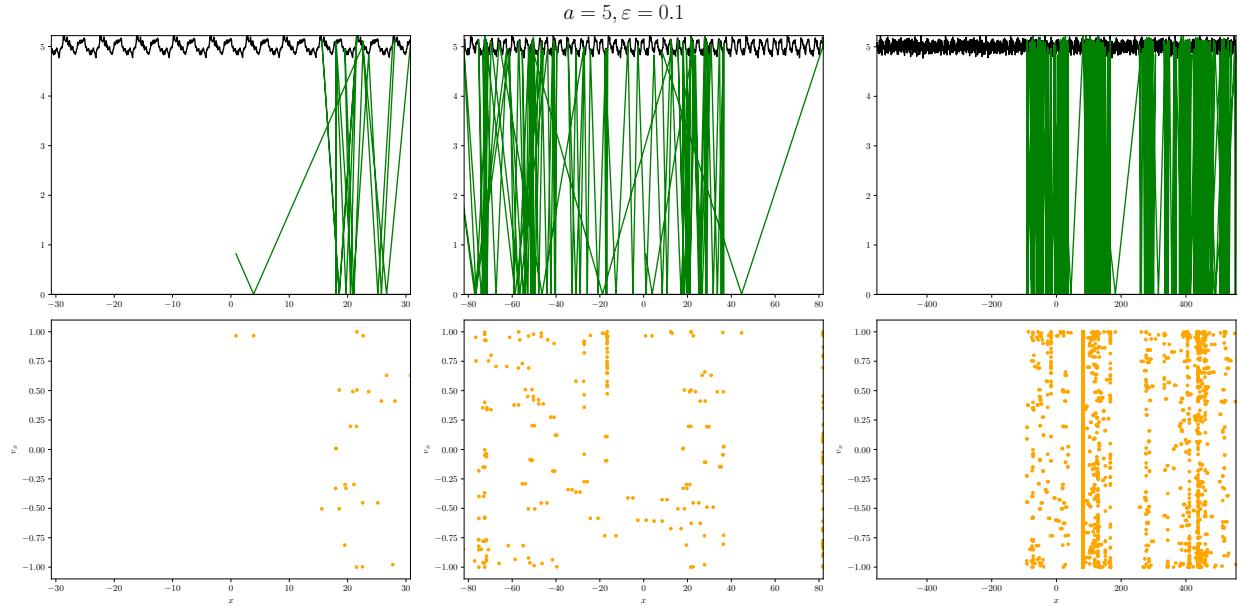
Nekaj tipičnih trajektorij delca pri različnih vrednostih parametra  $\varepsilon$ :



Na zgornjih 3 slikah je prikazan isti primer v več časovnih skalah. Skrajno desno so prikazani trki do 500000 korakov, levo od tega je 10x manj trkov, najbolj levo pa še 10x manj. Na spodnjih slikah so prikazani pripadajoči fazni portreti, kjer še nisem upošteval periodičnosti kanala

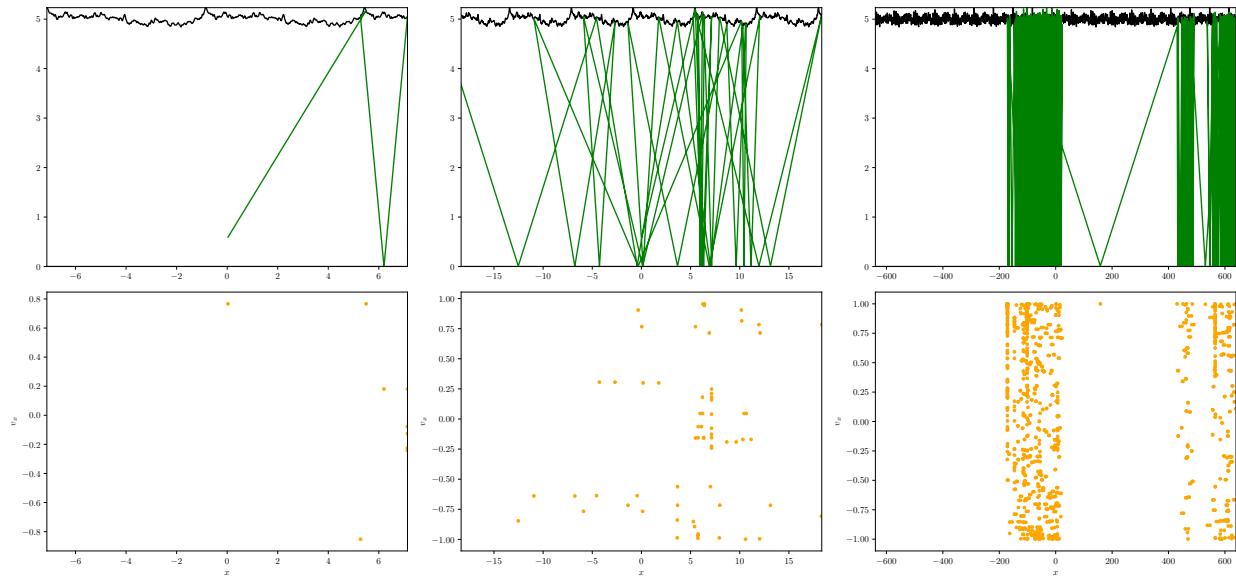


Še ena realizacija simulacije

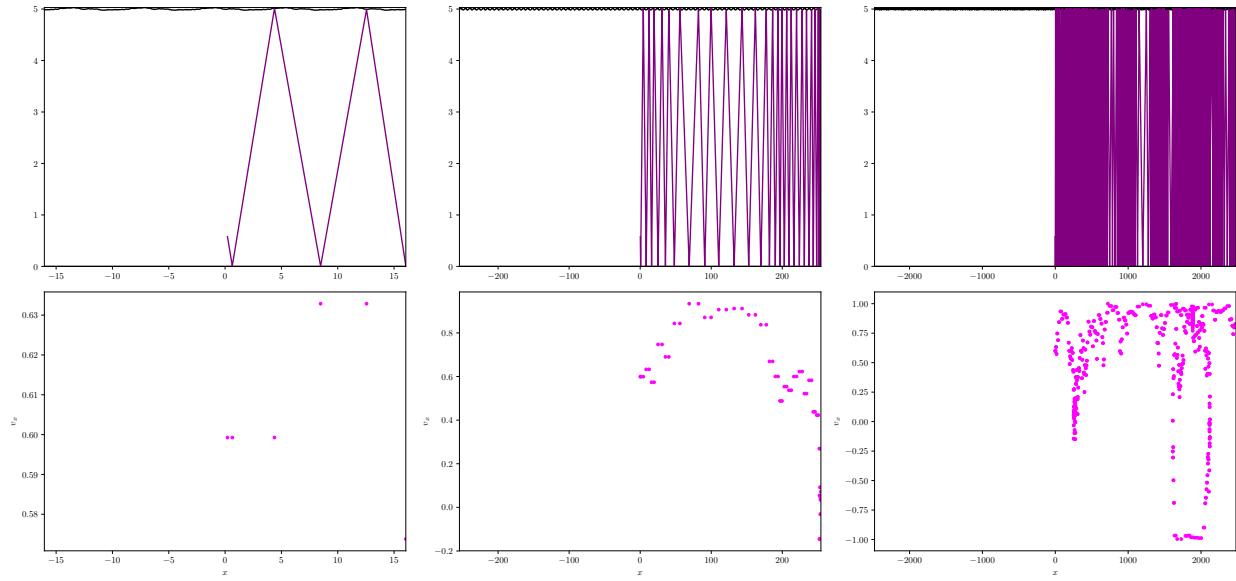


Pri manjšem epsilonu pridemo v povprečju dlje

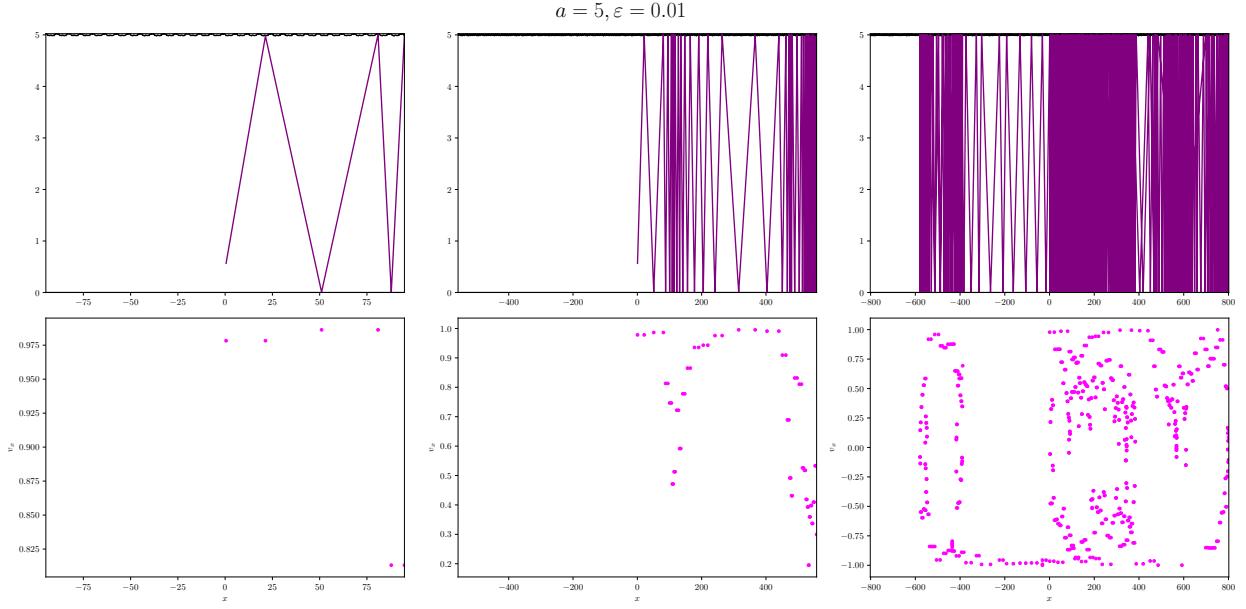
$$a = 5, \varepsilon = 0.1$$



$$a = 5, \varepsilon = 0.01$$



Fazni portret že začenja spremenjati obliko. Vemo, da je za povsem raven kanal fazni portret le horizontalna linija (konstantna hitrost v x smeri).



Še več primerov je prikazano na animacijah GibanjeX.mp4 , kjer sem simulacijo poganjal do  $t = 250$ . Evolucijo več bližnjih začetnih pogojev hkrati lahko spremjam v animacijah GibanjeVecX.mp4, kar nam pri večjih epsilon da slutiti, da imamo res kaotično obnašanje (kar bomo kasneje podrobnejše preiskali). Na animacijah na trenutke zgleda, kot da delec močno pospeši. To je le posledica tega, da je x skala veliko večja kot y.

## 4 Difuzija x koordinate

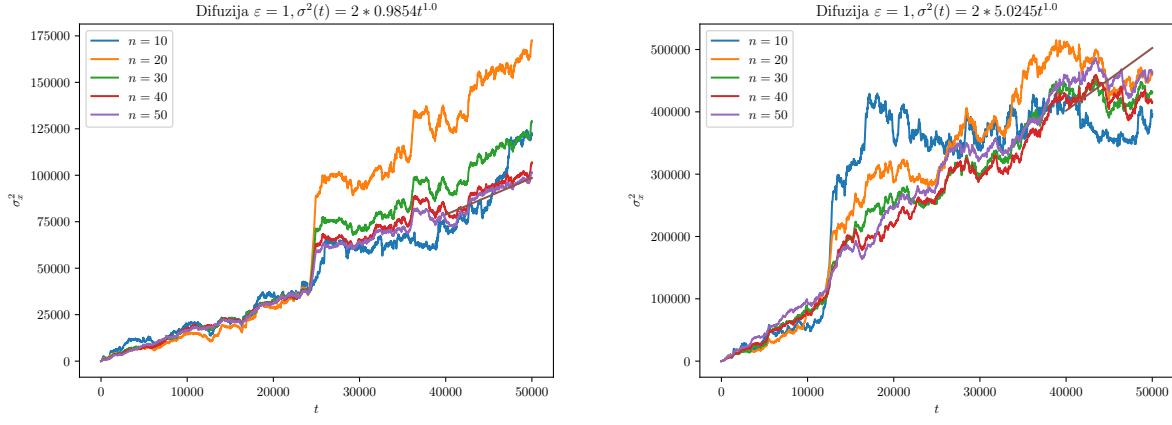
Preveriti želimo naslednje:

$$\lim_{t \rightarrow \infty} \langle (x(t) - x(0))^2 \rangle = 2Dt^\alpha$$

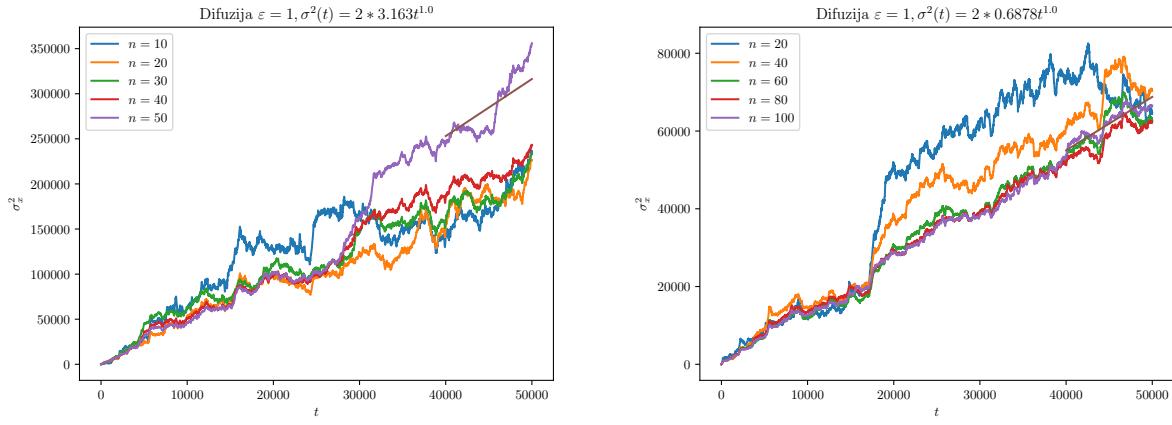
Kjer povprečujemo po veliko različnih začetnih pogojih delca in/ali realizacij biljarda. Zaenkrat se bom omejil na fiksen biljard, povprečeval pa bom le po različnih začetnih pogojih.

Ponovno smo naleteli na problem, katerega smo imeli že pri pripravi biljarda: pri katerem času bomo dovolj dobro aproksimirali  $t \rightarrow \infty$  in po koliko začetnih pogojih moramo povprečiti, da bo naše povprečje dovolj dobro aproksimiralo pravo povprečje po vseh možnih začetnih pogojih? Odgovora na to nimamo, zato kar poižkusimo nekje začeti: Levo stran zgornje enačbe bom risal do subjektivno dolgega časa in povprečil po različnem številu začetnih pogojev. Upam, da bom na neki točki videl, da je obnašanje prišlo iz začetnega v nek asimptotski režim, ter da so se deviacije izpovprečile v neko krivuljo na kateri bo šum čim manj viden.

S scipyjevo funkcijo curvefit, ki uporablja nelinearno metodo najmanjših kvadratov, je na spodnjih grafih z rjavo barvo narisan izračunani najboljši fit.



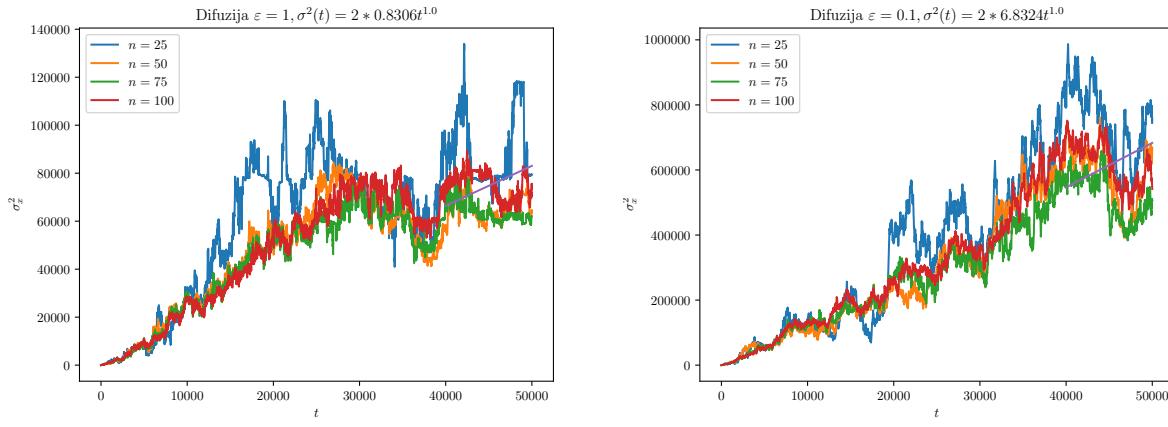
n tukaj predstavlja število različnih začetnih pogojev, po katerih sem povprečeval. Na levi sliki izgleda, kot da bi v nek asimptotski režim prišli pri približno 4 miljona korakov (kar ustreza času 40000) na desni pa se obnašanje ravno nekje proti koncu simulacije malce spremeni in se ”najboljši fit” kar razlikuje od dejanske slike.



Še dva primera. Na levi je moteče to, da je graf za n=50 krepko drugačen od ostalih

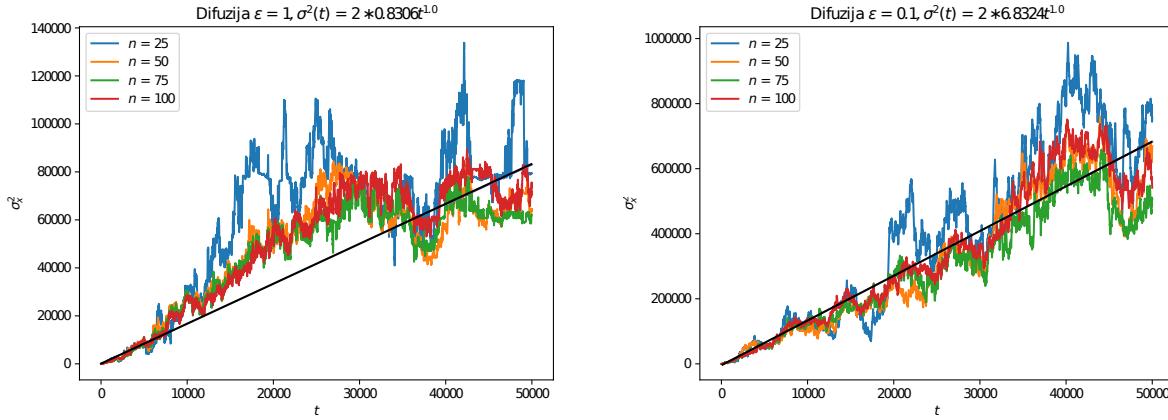
V zgornjih grafih opazimo zelo strme skoke v določeno smer in se zdi, kot bi prišli v nek nov režim v našem sistemu. Vendar so preprost razlog za skoke (redkejši) primeri takšnih odobjev od stene, da je hitrost skoraj izključno v x smeri in delec prepotuje veliko razdaljo preden se zopet odbije. Taki outlierji nam kvarijo obliko grafa.

Težave se bom poizksuil znebiti tako, da bom namesto povprečenja po začetnih pogojih raje gledal mediane, ki so manj občutljiva na takšne posebne primere. S tem dobimo naslednje grafe:



Zdi se, da sedaj vrednosti veliko bolj oscilirajo okoli nekega trenda. Izgleda pa, da se ta manj sunkovito spreminja kot prej. Na desni je narisan še primer  $\varepsilon = 0.1$

V zgornjih grafih bi pravilni naklon (in torej difuzijsko konstanto) lahko zadeli tudi, če bi premico prilagajali pri majhnih časih (Če premico podaljšamo do izhodišča se prilega tudi preostanku grafa). Poglejmo si kako to izgleda na zgornjih grafih, na katerih ni nekih velikih skokov:



Res zgleda, da povprečevanje okoli časa 10000, da isti naklon kot čakanje do časa 50000.

Na naslednjih simulacijah bom zato privzel, da lahko gledam čas le do 10000 in dobim zadovoljiv rezultat. Za večjo natančnost bom celotno simulacijo še večkrat (petkrat) realiziral in pridobljene difuzijske konstante še enkrat povprečil. Na ta način dobimo naslednje rezultate:

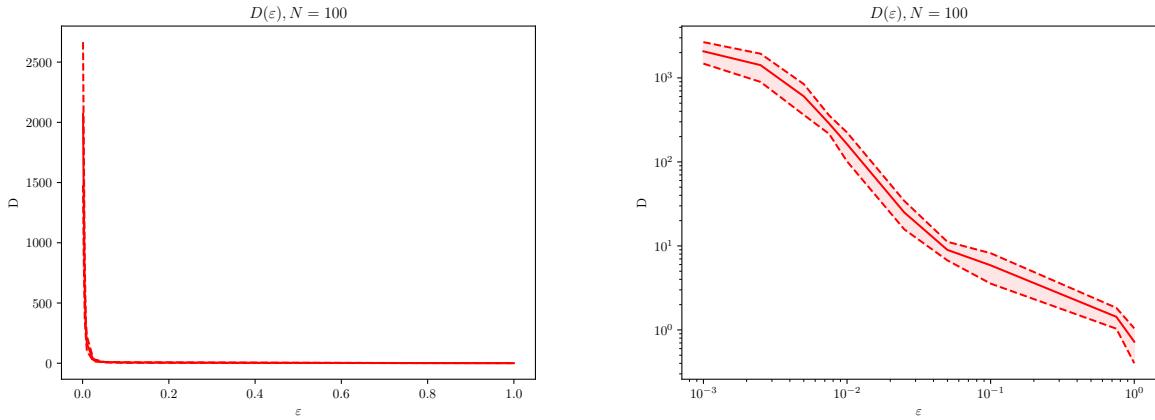
	$D(\varepsilon = 1)$	$D(\varepsilon = 0.1)$	$D(\varepsilon = 0.01)$
Povprečevanje po zač. pogojih	$1.32 \pm 0.93$	$5.91 \pm 2.09$	$174.9 \pm 20.4$
Povprečevanje po realizacijah kanala	$0.12 \pm 0.36$	$4.96 \pm 1.77$	$114 \pm 41$
Povprečevanje po obojem	$0.57 \pm 0.13$	$4.23 \pm 1.18$	$148.9 \pm 28.8$

Pri zgornjih rezultatih se nisem spuščal v nikakršno napredno statistiko. Absolutne napake so preprosto največja odstopanja povprečne vrednosti difuzijskih konstant od tistih 5, katerih sem povprečeval. Metoda linearnih kvadratov, pa je za napake parametrov vrnila zanemarljivo majhne številke (reda velikosti  $10^{-5}$ ). Očitno se difuzijska konstanta z epsilonom manjša, kar se povprečevanja po različnih stvareh tiče pa imamo dokaj velike razlike, čeprav so vse paroma v okviru napake. Ne bom se spuščal v iskanje vzroka teh razlik (verjetno so določeni načini povprečenja bolj občutljivi na določene posebne primere / outlierje kot ostali), raje se bom osredotočil na veliko bolj očitno odvisnost difuzijske konstante od  $\varepsilon$ .

Vrnimo se za trenutek k našem vprašanju iz prvega poglavja. Ali  $N = 100$  dovolj dobro opiše neskončno vsoto, ki definira našo zgornjo steno? Tako se difuzijska konstanta spreminja, če za iste začetne pogoje seštejemo več členov:

	$D(N = 50)$	$D(N = 100)$	$D(N = 200)$
$\varepsilon = 0.01$	310	118	115.5
$\varepsilon = 0.1$	9.13	3.48	3.11
$\varepsilon = 1$	$0.57 \pm 0.13$	$4.23 \pm 1.18$	$148.9 \pm 28.8$

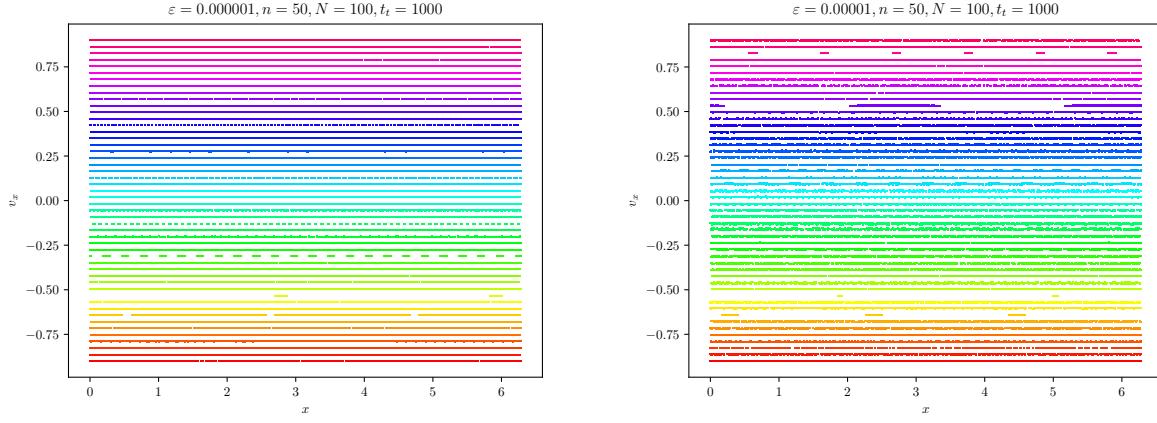
Po zgornjih podatkih sodeč, smo s seštevanjem le 100 členov neko napako sicer naredili (reda velikosti procent), vendar je ta v primerjavi s seštevanjem npr. le 50 členov zanemarljivo majhna. Zaradi tega skupaj z dejstvom, da bi simulacija za  $N = 200$  postala zelo počasna, bom ostal pri  $N = 100$ .



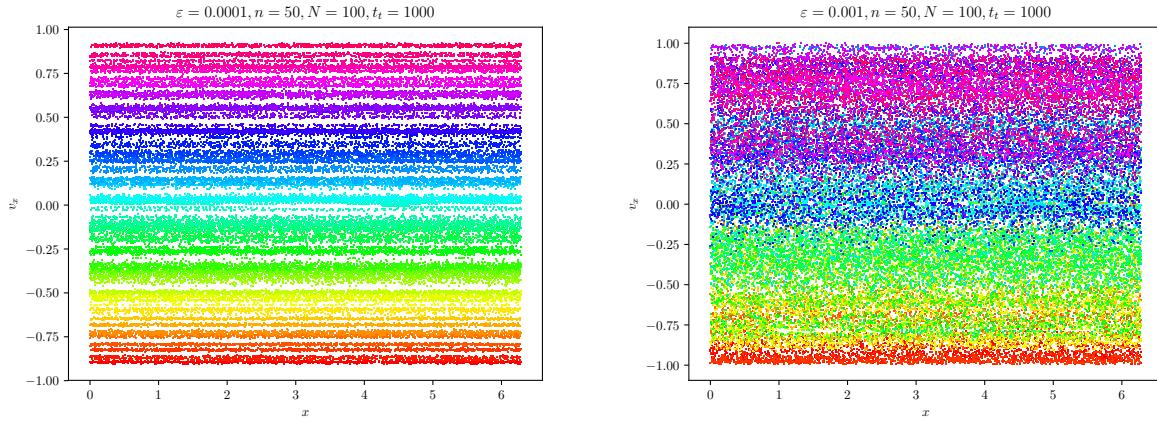
Končna odvisnost difuzijske konstante. S polno črto je prikazana povprečna vrednost difuzijske konstante po 5 simulacijah, s črtkano pa skrajni vrednosti, pridobljeni iz povprečne vrednosti in absolutne napake. Na levi je graf v navadni skali na desni pa na log-log skali.

## 5 Prehod v globalni kaos

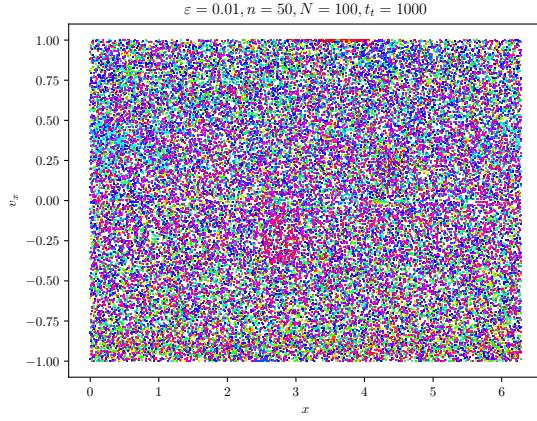
Ostane nam le še iz faznega portreta oceniti, kdaj preidemo v globalni kaos. V fazni portret bom risal točke  $(x, v_x)$  na Poincarejevi sečni ploskvi, ki bo stena biljarda(zgornja in spodnja). V prejšnjih slikah faznih portrefov (ki se sicer bolj logično ujemajo z sliko trajektorij) ni videti nobenih nam dobro znanih eliptičnih in hiperboličnih točk. Spomnimo pa se, da lahko na problem gledamo kot na končni biljard s periodičnimi robnimi pogoji. To bom upošteval in z operacijo modulo  $2\pi$  fazni prostor v x smeri preslikal na interval  $[0, 2\pi]$ .



V naslovu pomeni n število različnih začetnih pogojev pri katerih sem pognal simulacijo (vsak je predstavljen s svojo barvo),  $t_t$  pa je število trkov preden simulacijo končam. Za tako majhen  $\varepsilon$  kaotičnega obnašanja še ni, večinoma ostaja smer x komponente hitrosti enaka in se delec premika. Pri malce večjem  $\varepsilon$  se začne komponenta hitrosti komaj opazno spremenjati.



Na levi opazimo, da se smer hitrosti še malce bolj spreminja, a vseeno ostajamo na neki svoji eliptični orbiti. Na desni sliki pa se orbite počasi začnejo med sabo mešati in prehajamo v kaotični režim.



Pri  $\varepsilon = 0.01$  pa smo že krepko v kaotičnem režimu

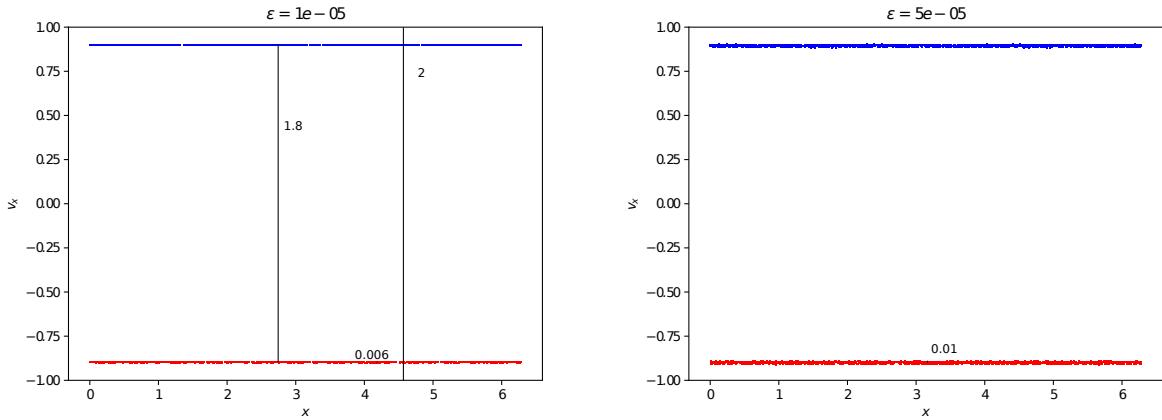
V animaciji Portret.mp4 sem podrobneje prečesal fazne portrete na intervalu  $\varepsilon \in [0.001, 0.01]$  z enakimi ostali parametri. Glede na to animacijo, bi ocenil, da v globalni kaos preidemo pri približno  $\varepsilon_c = 0.004$ . Gledal sem, kdaj na faznem portretu izginejo vodoravni "pasovi" podobne barve in se barve med seboj čisto pomešajo.

## 6 Kriterij Čirikova

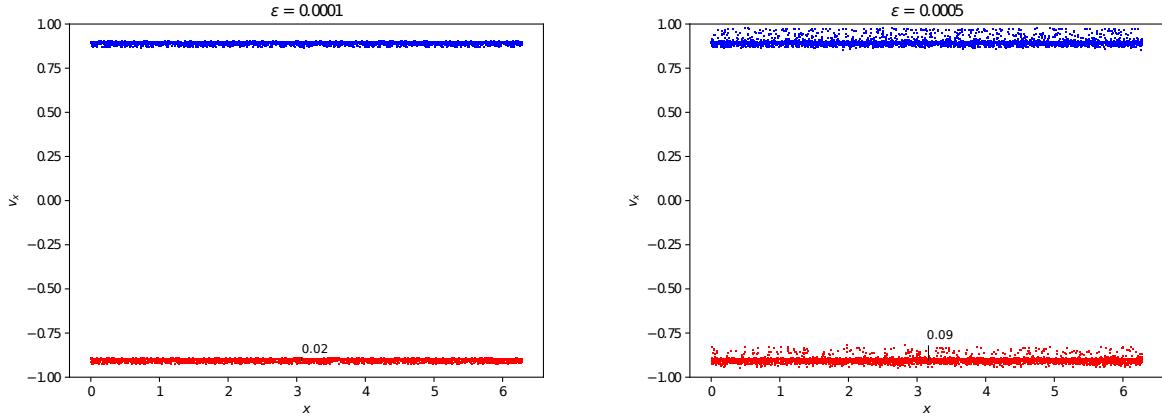
$\varepsilon_c$  bom poizkusil določiti še s kriterijem Čirikova, ki pravi, da v kaotični režim preidemo, ko se začnejo resonance prekrivati, torej, ko velja da je  $2\delta$  enak razmaku med resonancami  $\Delta$ . Na oku bom ocenil  $\delta$  pri več  $\varepsilon$  in rezultate ekstrapoliral do Čirikovega pogoja, kar nam bo dalo oceno za  $\varepsilon_c$ .

Gledal bom oddaljeni resonanci, med seboj zamaknjeni za  $v_x = 1.8$ , pri prekrivanju teh dveh bomo namreč zagotovo že v globalnem kaosu, saj se bodo prekrivale tudi vse tiste vmes. Da naključnost čim manj vpliva na rezultate, sem pri vsakem  $\varepsilon$  začel z enakimi začetnimi pogoji, koeficiente v steni pa preprosto reskaliral po naslednji formuli, ki velja za normalno porazdelitev  $N(0, \sigma_1) = \sigma_1 N(0, 1)$

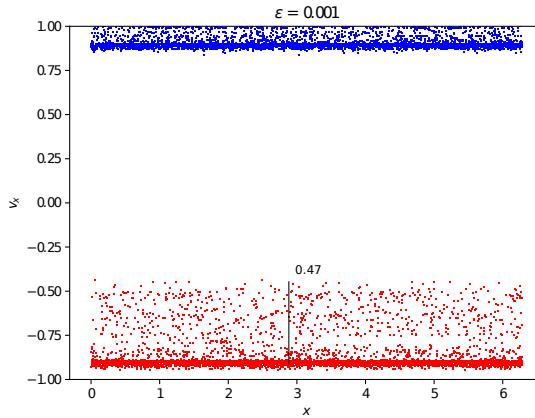
Na spodnjih grafih sem risal le te resonance in jih s programom za vektorsko grafiko Inkscape nato označil in izmeril.



Na levi sem poleg širine resonance označil še razmak med resonancama in velikost celotne y osi.

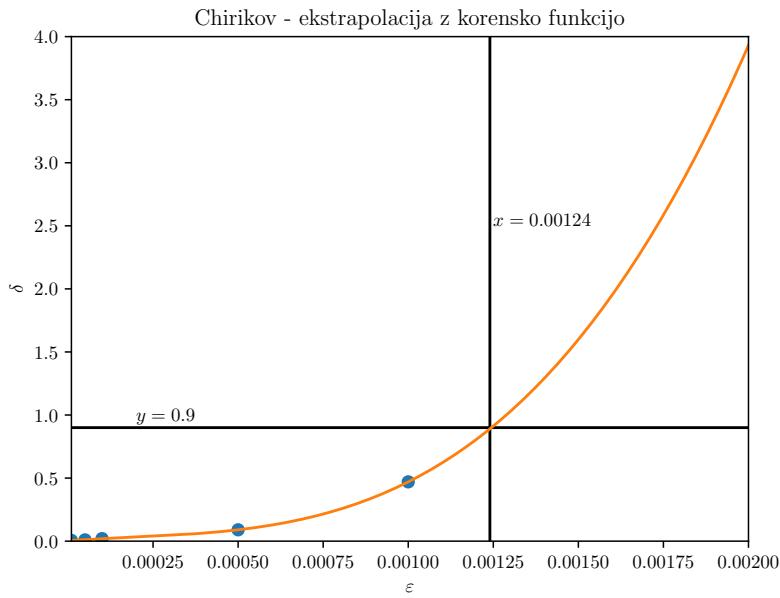


Na desni vidimo, da se resonanci ne širita ena proti drugi, kar se zdi, da bo morda dalo napačen rezultat, če rečem, da se resonanci prekrivata ko bo dvakratnik širine spodnje enak razliki med resonancami. Vseeno bom tako računal in pogledal ali se rezultat zelo razlikuje od tega iz prejšnjega poglavja.

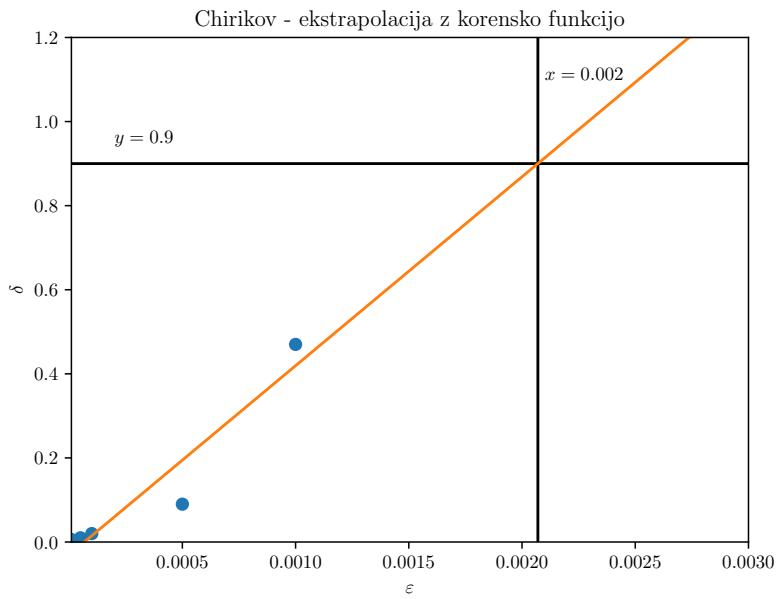


Počasi postaja vprašljivo, če to sploh lahko še smatramo za "povečevanje eliptične orbite", saj prehajamo v kaotični režim. S tem se ne bom preveč ubadal, saj nas navsezadnje zanima le ocena  $\varepsilon_c$ .

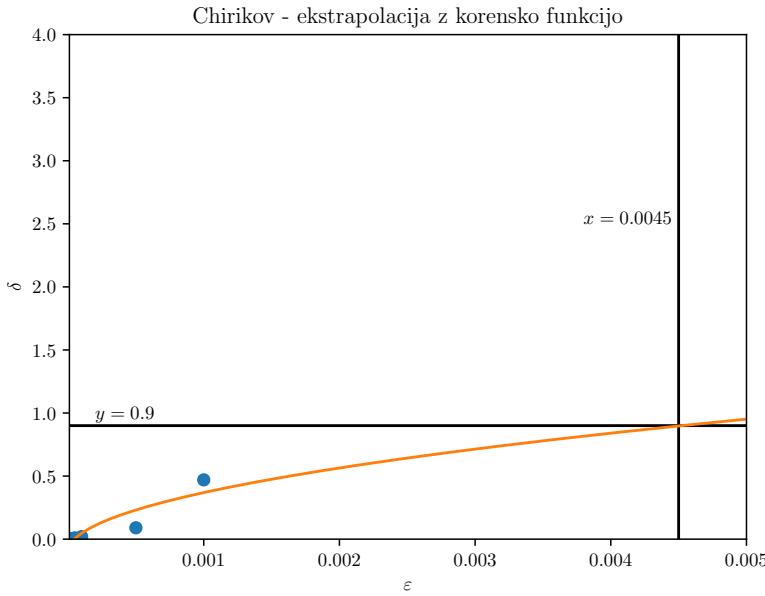
Sedaj je potrebno narediti še ekstrapolacijo do točke  $2\delta = \Delta$ . Oblika funkcije, katero bi točkam prilagajal ni očitna in sem jih zato preizkusil več:



Najprej sem na zgornje meritve naivno prilagajal funkcijo, ki se je zdela, da se bo z vsemi točkami najbolje ujela - polinom 3. stopnje. Takšna ekstrapolacija da  $\varepsilon_c = 0.00124$ , kar je približno trikrat manj od prejšnjega rezultata.



Polinom 3. stopnje je morda prehuda, saj vemo, da lahko z dovolj veliko stopnjo polinom fittamo na skoraj poljubno razporeditev točk. Tukaj sem se zato omejil na bolj enostavno funkcijo in sicer linearino. Takšna ekstrapolacija da  $\varepsilon_c = 0.002$ , kar je višje kot prej, a vseeno dvakrat nižje od ocenjenega  $\varepsilon_c$  v prejšnjem poglavju.



Poizkusil sem še s korensko funkcijo, kar je pogosta odvisnost pri kriteriju Čirikova. Takšna ekstrapolacija da  $\varepsilon_c = 0.0045$ , kar je višje kot predviden kritični parameter v prejšnjem poglavju, a se vidno najslabše ujema.

V spodnji tabeli sem podal še parametre, ki jih je moj fit (Funkciji polyfit in curvefit v scipy-ju) vrnil ter vrednost pri statistiki pogosto uporabljenega koeficienta  $R^2$ , ki nekako pove kako dobro se podatki prilagajo našemu modelu.  $R^2 = 1$  pomeni popoln fit.

	$f(\varepsilon)$	$\varepsilon_c$	$R^2$
Polinom 3. st.	$6 \cdot 10^8 x^3 - 324155 x^2 + 183x$	0.00124	0.99
Linearna	$449.3x - 0.03$	0.002	0.90
Korenska	$14.9\sqrt{x} - 0.1$	0.0045	0.78

Tako vidimo, da nam mora polinomski fit močno nabiti proste parametre, da dobi svojo natančnost. Zdi se, da torej to ni prava oblika funkcije. Poglejmo še linearno, ki ima manjšo napako, vendar tudi manjši kritični parameter  $\varepsilon_c$  in sicer 2 krat manjši od ocenjenega v prejšnjem poglavju. Ponavadi je ravno obratno in da Čirikov večji kritični parameter od resničnega. Razlogov za tako neujemanje v mojem primeru je lahko več: Lahko, da sem premalo natančno pomeril širine orbit ali pa napačno ekstrapoliral - oblika primerne funkcije bi lahko bila drugačna kot ena izmed preizkušenih. Še ena možnost je, da sem v prejšnjem poglavju prek gledanja faznih portretov napačno ocenil nastop kaosa. Morda se barve dodobra pomešajo že prej, vendar zaradi količine pik to ni bilo lepo vidno.

Korenski fit da bolj pričakovano vrednost kritičnega parametra, vendar je tudi napaka fita precej večja. Čeprav se korenska odvisnost v kriteriju Čirikova pogosto pojavlja, nimamo nekega prepričljivega razloga, da je korenski fit res ta pravi in torej ne moremo zagotovo zaključiti, katera funkcija nam da "najbolj pravi"  $\varepsilon_c$ .

Se pa oba načina skladata vsaj v redu velikosti kritične vrednosti epsilona in sicer  $10^{-3}$ . Ker je bila naloga  $\varepsilon_c$  zgolj oceniti, bom pri tej oceni tudi končal.