
3.2 Kvantni Monte-Carlo

MIHA SRDINŠEK

Pri tej nalogi sem se želel naučiti z Metropolisovim postopkom reševati tudi probleme v kvantni fiziki. Ukvarjal sem se z reševanjem kvantne particijske funkcije, ki se jo zapiše kot

$$Z(\beta) = \text{Tr} \left[e^{-\beta H} \right]. \quad (1)$$

To lahko recimo rešimo tako, da razbijemo eksponentno funkcijo na M kosov in vmes vstavimo identitete. Ko izpostavimo vsoto dobimo

$$Z(\beta) = \sum_{\underline{n}_1, \underline{n}_2, \dots, \underline{n}_M} \langle \underline{n}_1 | e^{-\frac{\beta}{M} H} | \underline{n}_2 \rangle \langle \underline{n}_2 | \dots \langle \underline{n}_M | e^{-\frac{\beta}{M} H} | \underline{n}_1 \rangle, \quad (2)$$

kjer je \underline{n}_i vektor, ki predstavlja večdelčno kvantno stanje, vsota pa je po vseh možnih vrednostih tega vektorja. Če vzamemo dovolj velik M pa lahko celo razvijemo eksponent in tako, pravzaprav računamo kar s Hamiltonjanom. Vsake ta "matrični" element potem obravnamo kot verjento za prehod in desnega v levo stanje, oziroma

$$Z(\beta) = \sum_{\underline{n}_1, \underline{n}_2, \dots, \underline{n}_M} P_{\underline{n}_1, \underline{n}_2} \dots P_{\underline{n}_M, \underline{n}_1}. \quad (3)$$

Pri tem moramo paziti da najdemo takšno bazo, da so v njej matrike P nenegativne. V našem primeru harmonskega ali anharmonskega oscilatorja s potencialom

$$V(x) = \frac{\mu}{2} x^2 + \lambda x^4, \quad (4)$$

lahko poračunamo P s pomočjo razcepa na kinetični in potencialni člen, ki ju znamo izračunati. Na tak način dobimo enačbo

$$P_{\underline{n}_i, \underline{n}_{i+1}} \propto \exp \left(-\frac{mM}{2\hbar\beta} (\underline{n}_{i+1} - \underline{n}_i)^2 - \frac{\beta}{M} V(\underline{n}_i) \right). \quad (5)$$

Algoritem se sedaj glasi:

1. Izžrebaš začetno stanje, kar pomeni, M baznih vektorjev. S temi baznimi vektorji izračunaš začetno akcijo, oz. v našem primeru energijo. Torej argument produkta eksponentov oz. vsoto

$$\sum_{i=0}^M \left(-\frac{mM}{2\hbar\beta} (\underline{n}_{i+1} - \underline{n}_i)^2 - \frac{\beta}{M} V(\underline{n}_i) \right) \quad (6)$$

2. Zdaj izžrebaš naključno rezino i oziroma bolj natančno "identiteto", in zamenjaš bazno stanje z novim, ki je prejšnje plus nek vektor naključnih števil, katerih obsek je določen s parametrom ϵ . V našem primeru je to samo eno število. S to novo bazo izražunaš kako se bo s to potezo spremenila energija oziroma verjetnost da se nahajamo v tem stanju

3. Če je sprememba energije negativna, oziroma verjetnost večja, potem potezo sprejmemo in ponovimo postopek, če je poteza zavrnjena izžrebamo naključno število na intervalu $[0, 1)$ in pogledamo če je manjše od verjetnosti za spremembo energije $e^{-\Delta E}$. Če je manjše potezo sprejmemo in ponovimo algoritem, če je večje potezo zavrnemo in ponovimo algoritem.

Pričakovano vrednost energije pa izračunamo po formuli

$$\frac{\langle H \rangle}{N} = \left\langle \frac{M}{2\beta} - \frac{mM}{2\hbar^2\beta^2N} \sum_{j=1}^M (q_{j+1} - q_j)^2 + \frac{1}{MN} \sum_{j=0}^M V(q_j) \right\rangle, \quad (7)$$

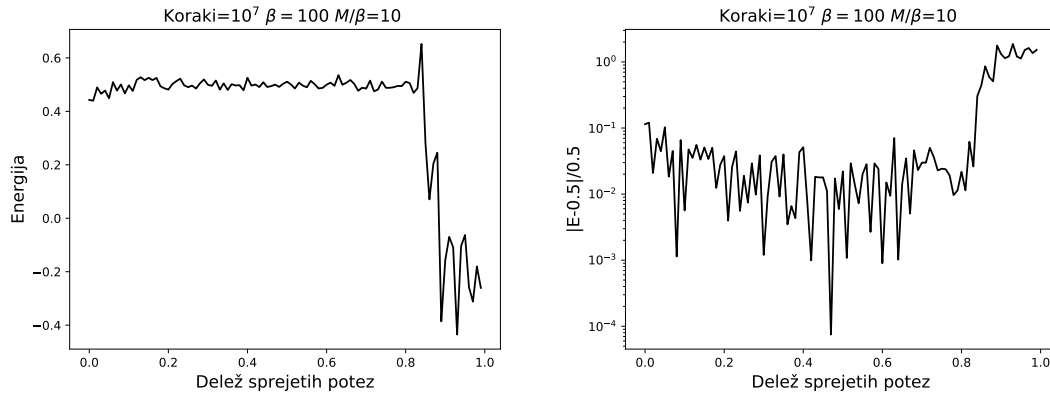
Kar sem jaz izvedel tako, da sem vakih $10M$ korakov izračunal energijo in na koncu poiskal povprečje.

I. TEST ALGORITMA NA HO

Z namenom da preverim če algoritem dobro deluje, sem se lotil harmonskega oscilatorja. Pri tem sem sprva naletel na probleme. Najprej sem po zgoraj opisanem postopku izžrebal začetno stanje, potem pa na vsakem koraku algoritma pri naključnem "časušpremenil bazno stanje, tako da sem mu prištel naključno število. Poskusil sem marsikaj. Najprej sem žbral naključno gavsovsko število s povprečjem nič in varianco ϵ , nato sem žrebal število s povprečjem ϵ in neko varianco (recimo 1), na koncu pa tudi enostavno naključno število na intervalu $x \in (-\epsilon, \epsilon)$. Vse to sem počel pri izbranem β in številu razkosanj M . Pri recimo $\epsilon = 0.1$ in $M = 1000$ in $\beta = 1$ mi je to dajalo dobre rezultate, a je začelo odstopati pri pri ostalih β . Nato sem se odločil da moram poenotiti izbiro razmerja M/β , ker je to pravzaprav tisto kar mi določa, kako dobro velja razcep eksponentne funkcije. S tem sem ukrotil odstopanja do katerih je prihajalo pri velikih β , vseeno pa sem dobil rezultate, ki so bili močno odvisni od izbire ϵ . Na tej točki sem opustil žrebanje gavsovskih števil in žrebal le še števila na intervalu $(-\epsilon, \epsilon)$. Šele tu sem se odločil da moram resneje upoštevati profesorjev komentar, glede razmerja med številom zavrnjenih in sprejetih potez. To sem naredil tako, da sem na vsakem koraku algoritma enostavno izvedel potezo

$$\epsilon(t+1) = \epsilon(t) + \left(\frac{i_t}{t} - r\right) \frac{1}{M_*}, \quad (8)$$

kjer je t število koraka v algoritmu, i_t število sprejetih korakov do koraka t , M_* pa je nek faktor. r predstavlja želeno prepustnost našega algoritma, torej želeno razmerje med sprejetimi in vsemi potezami. Sprva sem za faktor M_* uzbral kar M in dobil končno zelo dobre rezultate pri mnogih izbirah r , ampak le pri $\beta > 1$. To seveda zaradi tega, ker je pri višjih temperaturah M ratal premajhen in se je ϵ spreminjal preveč drastično. Zato sem zapisal še en pogoj, ki je na začetku preveril ali je $\beta \geq 1$, če je večji od ena vzamem $M_* = M$, če je manjši pa neko arbitrarno funkcijo, recimo $M_* = M * 100$ in začetni $\epsilon = \sqrt{\beta}$, kar je tudi arbitrarni in samo zato, ker sem enkrat vmes mislil da je to pomembno, mislim pa da bi moralo delovati tudi brez tega. Po vseh teh začetnih težavah je algoritem začel delovati brezhibno.



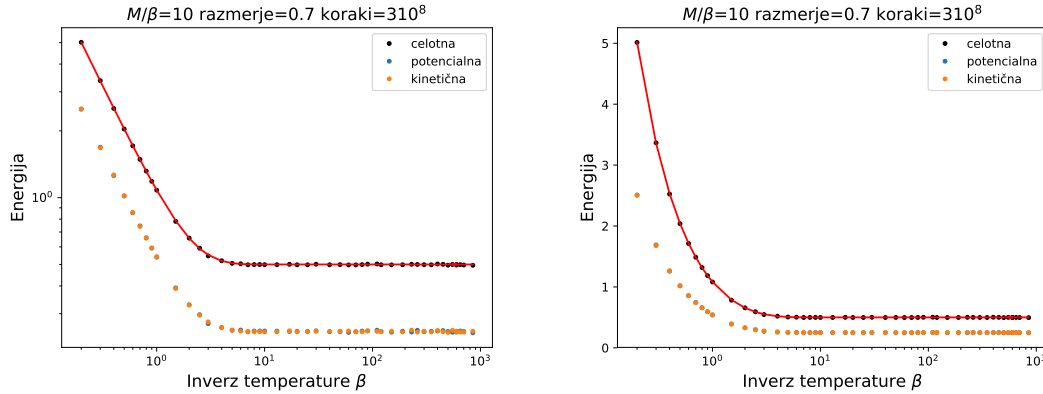
Slika 1: Sliki prikazujeta kako veliko je odstopanje od analitične rešitve pri različnih razmerjih med prejetimi in svemi potezami. Na desni vidimo relativno napako, na levi pa enostavno vrednost energije.

Algoritem sem nato uglasil tako, da sem na primeru harmonskega oscilatorja, kjer poznam analitični rezultat

$$E_0 = \frac{1}{2} \sqrt{\frac{\mu}{m}} \quad (9)$$

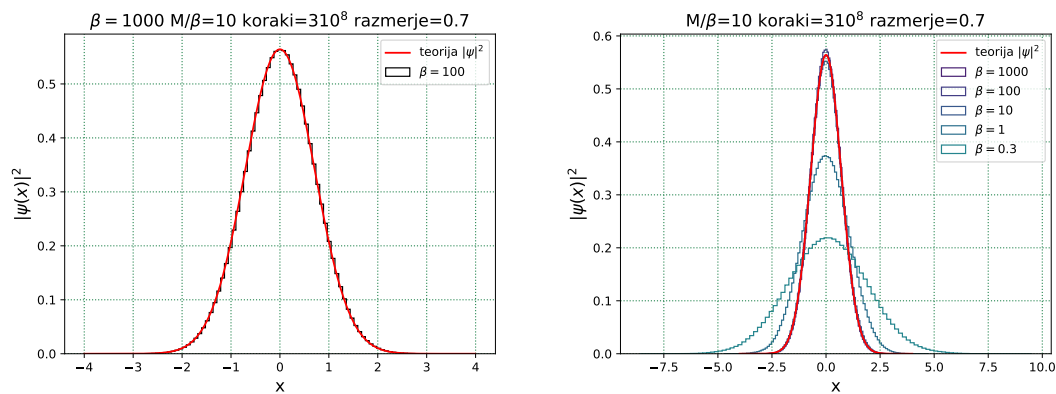
izrisal slike prikazani na 1. Na slikah se lepo vidi, da je moramo res paziti na razmerje, ampak da hkrati ni spet tako zelo riskantno. Jaz sem delal z vrednostmi med 0.5 in 0.7 in dobival dobre

rezultate. Iz teh in še nekaterih drugih podobnih slik ki sem jih narisal, pa bi lahko lahko rekel, da je moj algoritem verjetno najbolj natančen pri $r = 0.5$. Na ta način sem si že pogledal limito $\beta \rightarrow \infty$, ampak zelo nenazorno, zato sem izrisal še 2.



Slika 2: Sliki prikazujeta isti rezultat: kako se energija spreminja z β . Leva slika ima y os v logaritmski skali, desna pa nima, to je vsa razlika. Rdeča krivulja prikazuje analitično napoved.

Na sliki 2 sem izrisal še kinetično in potencialno energijo, ki kot vidimo znašata ravno polovico celotne energije. Rdeča krivulja, pa prikazuje analitično napoved. Vse skupaj torej deluje prepričljivo in lahko bi že rekel da očitno algoritem deluje pravilno. Vseeno, pa sem želel prej preveriti še kako izgleda valovna funkcija, ki jo napove ta algoritem. To sem izračunal tako, da sem hkrati z energijo shranil še vse položaje in na koncu narisal histogram, ki sem ga moral potem še primerno normirati. To vidimo na slikah 3. Zanimalo me je tudi kako se rešitev spreminja s temperaturo in očitno se valovna funkcija počasi razleze. Pri tem sem izračunal tudi pričakovani položaj in večinoma sem dobil 0 do šeste decimalke natančno. Tak primer lahko vidite na koncu na sliki 9 levo zgoraj.



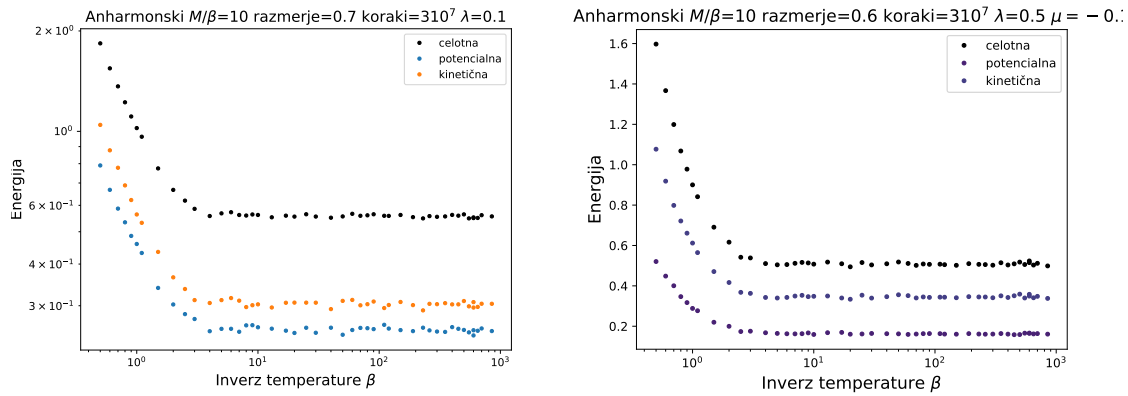
Slika 3: Sliki prikazujeta izračunano verjetnostno gostoto osnovnega stanja pri izbrani temperaturi.

II. ANHARMONSKI OSCILATOR

Sedaj uporabim novi potencial, ki izgleda kot

$$V(x) = \frac{\mu}{2}x^2 + \lambda x^4, \quad (10)$$

in z njim delam dalje. Že prej smo imeli opravka s parametrom μ . Pri tem potencialu je meni predvsem zanimivo, kadar izberemo $\mu < 0$, in potencial namesto enega dobi dva minimuma. Ko algoritem zaide v enega izmed minimumov iz njega težko kar tako pobegne, zato se na slikah 7 dogaja to, da je verjetnostna gostota večja na eni kot na drugi strani. To vemo iz preostalih nalog, da ni pravilna rešitev, in da bi morale z velikim številom korakov priti do izenačenja.



Slika 4: Sliki prikazujeta energijo v odvisnosti od inverzne temperature za dva primera anharmonskega oscilatorja.

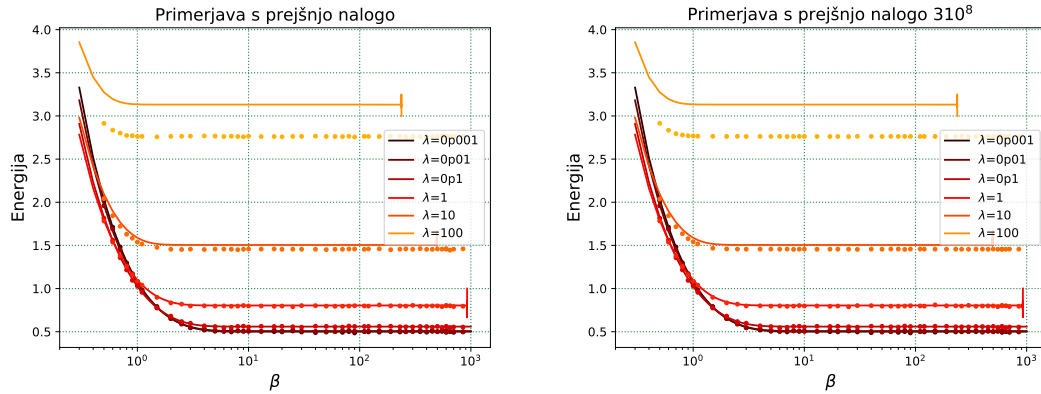
Da testiramo, če je algoritem konsistenten z rezultati iz druge naloge pri našem predmetu, in s tem stestiramo, če algoritem pravilno deluje, izračunamo nekaj energij pri $\beta = 100$ in jih primerjamo s tistimi iz druge naloge. Rezultate vidimo v spodnji tabeli.

λ	E_0 2.nal.	E_0 zdaj
0.001	0.5007	0.503047
0.01	0.507	0.504633
0.1	0.56	0.55724
1	0.804	0.798599
10	1.5	1.455449
100	3.13	2.76034

Dobim zelo lepo ujemanje z metodo razvoja po končni bazi. Ampak nam rešitve odstopajo s tem ko zvišujemo λ . Pri tem sam nisem preveč siguren katera rešitev je dejansko pravilna. Metoda, ki smo jo uprabljali v drugi nalogi je če prav razumem pertrubativna, med tem ko je ta naša nepertrubativna in bi morala ohranjati natančnost z zviševanjem λ . Amapk tudi o tem nisem preveč prepričan in nenazadnje pričakoval bi, da je metoda iz druge naloge vseeno zelo natančna pri energijah osnovnega stanja. Vse to je malce nenavadno zato sem se odločil, da poračunam še pričakovano vrednost energije s pomočjo lastnih energij izračunanih pri drugi nalogi. Samo uporabil sem energije, ki sem jih že prejšnjih poračunal in shranil v datoteke in izrisal

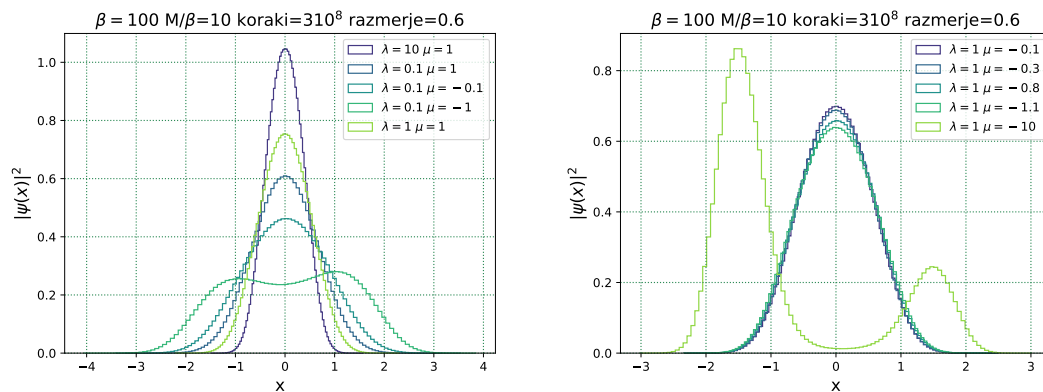
$$\langle E \rangle = \frac{\sum_{n=0}^{n_{\max}} E_n e^{-\beta E_n}}{\sum_{n=0}^{n_{\max}} e^{-\beta E_n}}. \quad (11)$$

Uporabil sem torej okoli 800 najnižjih energij, rezultat pa vidimo na slikah 5. Ostanemo pri istih problemih kot na začetku. Kolikor natančni smo dobimo isti rezultat z metropolisovim algoritmom. Vprašanje je torej, kateri rezultat je pravilen?

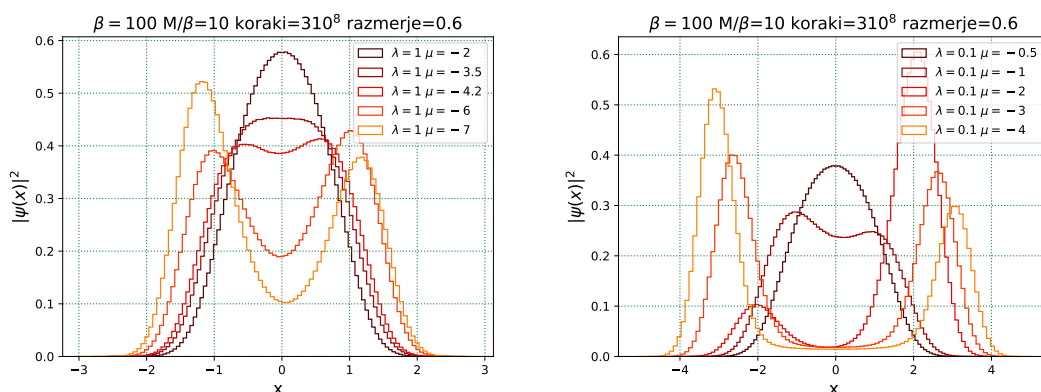


Slika 5: Sliki prikazujeta odvisnost energije od β . Sta zelo natrpani zato naj pojasnim: Gladka črta prikazuje rezultat, ki ga dobimo z pomočjo najnižjih 800 lastnih energij, ki smo jih zračunali pri drugi nalogi tega predmeta. Točke pa predstavljajo rezultat, ki ga dobimo z metropolisovim algoritmom. Na prvi pogled sta idejnični, a v resnici leva predstavlja rezultat po 310^7 korakih algoritma, desna pa po 310^8 korakih. Vidimo, da se rezultat z večjim številom korakov ne spremeni. Črtice na koncu krivulj so točke, kjer enačba z energijami iz druge naloge odpoje, domnevam da zaradi deljenja z ničlo.

Lahko pa sedaj izkoristimo trenutek in si pogledamo verjetnostne gostote za različne potenciale. Predvsem me je zanimalo kaj se zgodi, če pri isti temperaturi večamo in večamo bariero med obema minimuma potenciala, do katerih pride pri negativni μ . Na sliki 6 levo vidimo kar vse nametano, samo za občutek, kakšne rešitve obstajajo. Na sliki desno pa vidimo potem kako se rešitev spreminja, če pri konstantni λ znižujemo μ . Isto potem lahko opazujemo tudi na slikah 7.

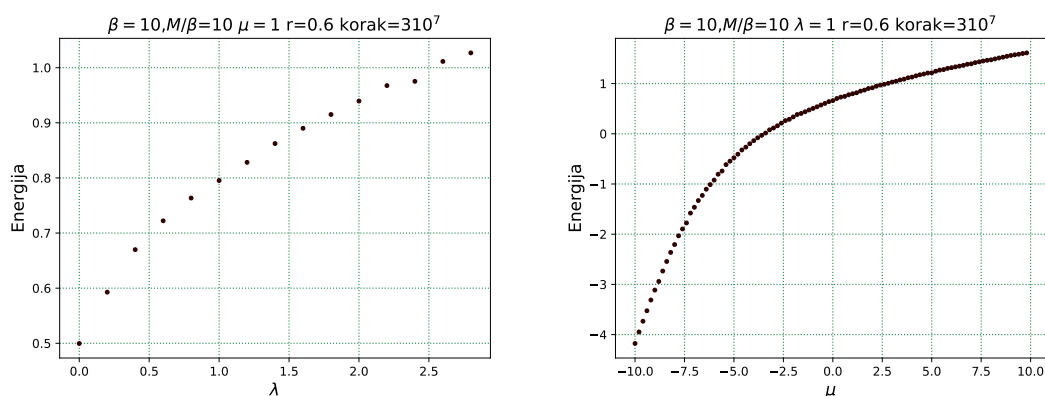


Slika 6: Sliki prikazujeta verjetnostno gostoto osnovnega stanja pri različnih potencialih.



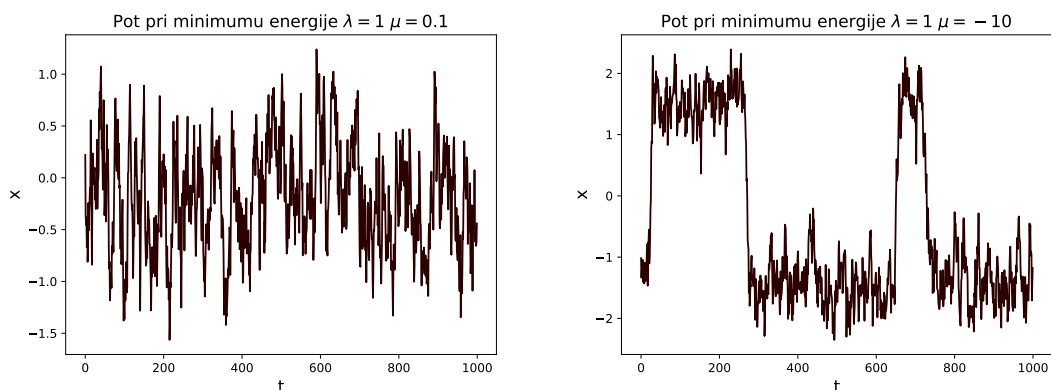
Slika 7: Slike prikazujeta verjetnostno gostoto osnovnega stanja pri različnih potencialih.

Zdaj ko smo si ogledali kakšen učinek ima spreminjanje potenciala na kvadrat valovne funkcije, si lahko ogledamo kako se s spreminjanjem parametrov λ in μ spreminja energija. to lahko vidimo na slikah 8. Za λ je rezultat precej intuitiven, krivulje pa ne znam razložiti, ampak v logaritemski skali izgleda premica. Res pa je, da sem izračunal malo točk. Za μ je rezultat dosti bolj zanimiv, saj ima nekakšen prevoj, ko energija postane negativna. Nisem prepričan ali je to pravilna rešitev, ampak ne vidim razloga zakaj bi o njej dvomil. Lahko bi si pogledal tudi negativen λ , ampak mislim da bi bilo to precej nesmiselno, saj sistem nikoli ne bo prispel do minimuma.



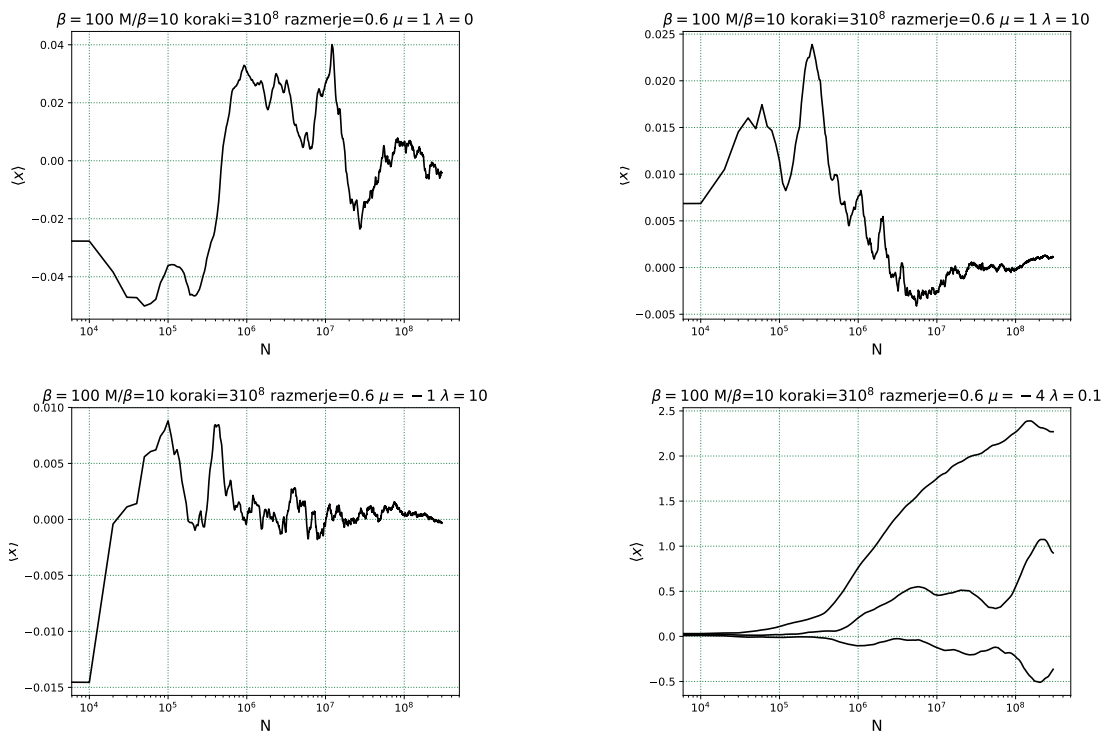
Slika 8: Na slikah vidimo kako se energija osnovnega stanja spreminja s potencialom. Enega od parametrov fiksiramo in drugega variramo. Na levi sliki variramo λ , na desni pa μ .

Zdaj pa bi rad potrdil, še mojo trditev, da dobivamo takšno nesimetrično rešitev na slikah 7 zgolj zato, ker algoritem težko pobegne iz minimuma in zato dolgo časa ostane v enem preden zamenja v drugega. Potrditev tega zares vidimo na slikah 9 in tudi 10.



Slika 9: Siki prikazujeta bazna stanja ob zaključku algoritma, po 310^8 korakih. Lepo se vidi, da so ob razbitju potenciala na dva minimuma, tudi bazna stanja razbita. Iz tega postane jasno, zakaj imamo različne vrhove pri verjetnostni gostoti.

Tu se je treba spomniti, da prikazano nima nobene veze s pričakovano vrednostjo. Kar vidimo na slikah so zgolj bazna stanja v zadnjem trenutku algoritma. Zanimivo pa bi si bilo pogledati tudi kako se tekom algoritma spreminja pričakovana vrednost kraja. To vidimo na sklopu slik 10. Načeloma lahko opazimo, da je algoritem zanesljiv, in da se povprečna vrednost nahaja okoli ničle, ko postane na sliki zgoraj densno potencial bolj strm se po pričakovanjih algoritem ničle drži še bolje. Kar me je presenetilo je slika spodaj levo, kjer je pričakovana vrednost tudi zelo blizu ničle, malce nenavaden rezultat pa daje slika spodaj desno kjer vidimo, da se očitno zgodi to, da se sistem kar ujame v enega izmed minimumov.



Slika 10: Slike prikazujejo kako se povprečna vrednost položaja spreminja s koraki algoritma. Na sliki spodaj desno sem prikazal več krivulj, saj je očitno algoritem tam precej nezanesljiv.

Poskusimo še s trikom, za katerega sem izvedel v članku [3]. V članku trdijo, da lahko s pomočjo računanja korelacijskih funkcij določimo razmak med osnovnim in prvim energijskim stanjem po formuli

$$E_1 = -\frac{M}{\beta} \ln \left(\frac{\langle x(0)x(0) \rangle}{\langle x(0)x(1) \rangle} \right) + E_0. \quad (12)$$

Sam se nisem preveč poglobljal v to, sem pa preveril, če drži. To sem na hitro spisal in dobil kar dobro ujemanje z drugo nalogo, kar lahko vidimo v sponji tabeli.

λ	E_0 2.nal.	E_0 zdaj	E_1 2.nal.	E_1 zdaj
0.001	0.5007	0.503047	1.504	1.531
0.01	0.507	0.504633	1.536	1.4925
0.1	0.56	0.55724	1.7695	1.789
1	0.804	0.798599	2.738	2.740
10	1.5	1.455449	5.322	5.257
100	3.13	2.76034	11.187	10.359

LITERATURA

- [1] <https://www-zeuthen.desy.de/students/2014/reports/RodgersRaes.pdf>
- [2] Westbroek, Dürr, Vvedensky, User's guide to Monte Carlo methods for evaluating path integrals
- [3] <http://cyber.sci-hub.tw/MTAuMTAxNi8wMDAzLTQ5MTYoODEpOTAwNzQtOQ==/10.1016>