

Univerza v Ljubljani
Fakulteta *za matematiko in fiziko*



VRM
Kvantni Monte Carlo

Andrej Kolar - Požun

25. 4. 2019

Algoritem

Klasični metropolisov algoritem želimo uporabiti za izračun opazljivk v kvantnih sistemih. Obravnavamo Hamiltonjan kanonične oblike:

$$\begin{aligned} H &= T + V \\ T &= -\frac{1}{2} \sum_{j=1}^N \frac{\partial}{\partial q_j} \\ V &= V(q) \end{aligned}$$

Pogosto želimo izračunati particijsko funkcijo in termične pričakovane vrednosti:

$$\begin{aligned} Z(\beta) &= \text{Tr} \exp(-\beta H) \\ \langle A \rangle &= Z(\beta)^{-1} \text{Tr} A \exp(-\beta H) \end{aligned}$$

S pomočjo Trotterjeve formule lahko izračunamo $\exp(-\beta H)$:

$$\begin{aligned} \exp(-\beta(T + V)) &\approx (\exp(-\beta V/M) \exp(-\beta T/M))^M \\ \langle q | \exp(-\beta V) | q' \rangle &= \delta(q - q') \exp(-\beta V(q)) \\ \langle q | \exp(-\beta T) | q' \rangle &= G_0(q, q'; \beta) = \left(\frac{1}{2\pi\beta} \right)^{N/2} \exp\left(-\frac{1}{2\beta}(q - q')^2\right) \end{aligned}$$

Nato lahko particijsko funkcijo zapišemo s pomočjo identitete $\int dq |q\rangle \langle q| = I$:

$$\begin{aligned} Z(\beta) &= \int \prod_{j=1}^M dq_j e^{-\frac{\beta}{M} V(q_1)} G_0\left(q_1, q_2; \frac{\beta}{M}\right) e^{-\frac{\beta}{M} V(q_2)} G_0\left(q_2, q_3; \frac{\beta}{M}\right) \dots e^{-\frac{\beta}{M} V(q_M)} G_0\left(q_M, q_1; \frac{\beta}{M}\right) = \\ &= \left(\frac{1}{2\pi\beta} \right)^{MN/2} \int \prod_{j=1}^M dq_j \exp(-E(q_1, q_2, \dots, q_M)) \\ E(q_1, q_2, \dots, q_M) &= \sum_{j=1}^M \left(\frac{M}{\beta} (q_{j+1} - q_j)^2 + \frac{\beta}{M} V(q_j) \right) \end{aligned}$$

S pomočjo zgornjega lahko pričakovano vrednost opazljivk napišemo kot:

$$\begin{aligned} \langle A \rangle &= \int \prod_{j=1}^M dq_j A_{q_1} P_{q_1, q_2} \dots P_{q_M, q_1} \\ P_{(q_1, q_2)} &\propto \exp\left(-\frac{M}{\beta} (q_2 - q_1)^2 - \frac{\beta}{M} V(q_1)\right) \end{aligned}$$

S pomočjo Metropolisla lahko vzorčimo po porazdelitvi:

$$P_{(q_1 \dots q_M)} = Z^{-1} P_{q_1, q_2} \dots P_{q_M, q_1}$$

Za potezo Metropolisovega algoritma lahko izberemo naključno izbrano časovno rezino j med 1 in M in majhna sprememba vektorja $q_j \rightarrow q'_j$. Potezo potem sprejmemo z verjetnostjo:

$$P = \min\left(1, \frac{P_{q_{j-1}, q'_j} P_{q'_j, q_{j+1}}}{P_{q_{j-1}, q_j} P_{q_j, q_{j+1}}}\right)$$

Opazljivke, diagonalne v pozicijski bazi lahko potem enostavno izračunamo, nekatere nediagonalne pa lahko izračunamo iz particijske funkcije, na primer:

$$\frac{\langle H \rangle}{N} = -(NZ)^{-1} \partial_\beta Z(\beta) = \left\langle \frac{M}{2\beta} - \frac{M}{2\beta^2 N} \sum_{j=1}^M (q_{j+1} - q_j)^2 + \frac{1}{MN} \sum_{j=1}^M V(q_j) \right\rangle$$

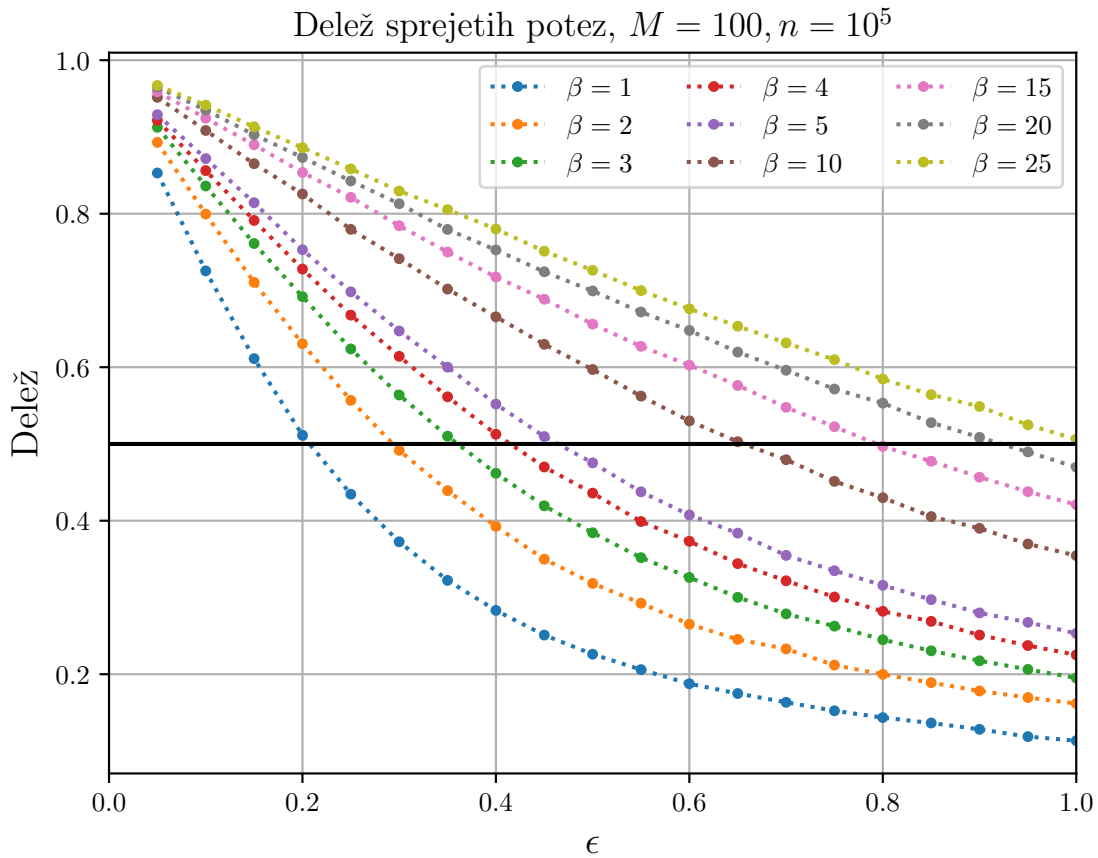
Enodimenzionalni oscilator

Za začetek obravnavajmo harmonski oscilator, torej:

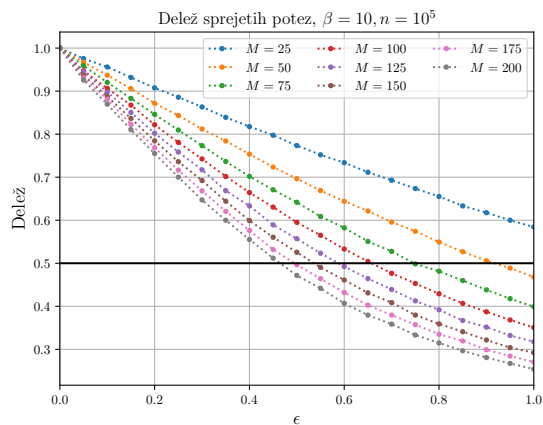
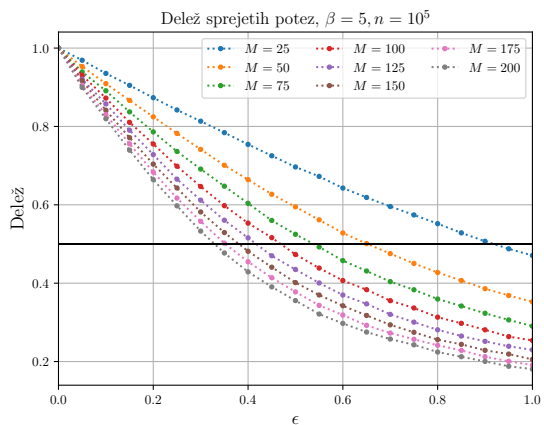
$$H = -\frac{1}{2}\partial_q + \frac{1}{2}q^2$$

Pri implementaciji algoritma bom za začetno konfiguracijo vzel M naključno generiranih števil med -1 in 1 , razen ko bom imel že na voljo stacionarno konfiguracijo višje temperature (nižje β). V tem primeru bom vzel slednjo.

Najprej se je treba odločiti za število časovnih rezin M in "velikost" poteze ϵ . Želimo, da v povprečju sprejmemo približno polovico potez.

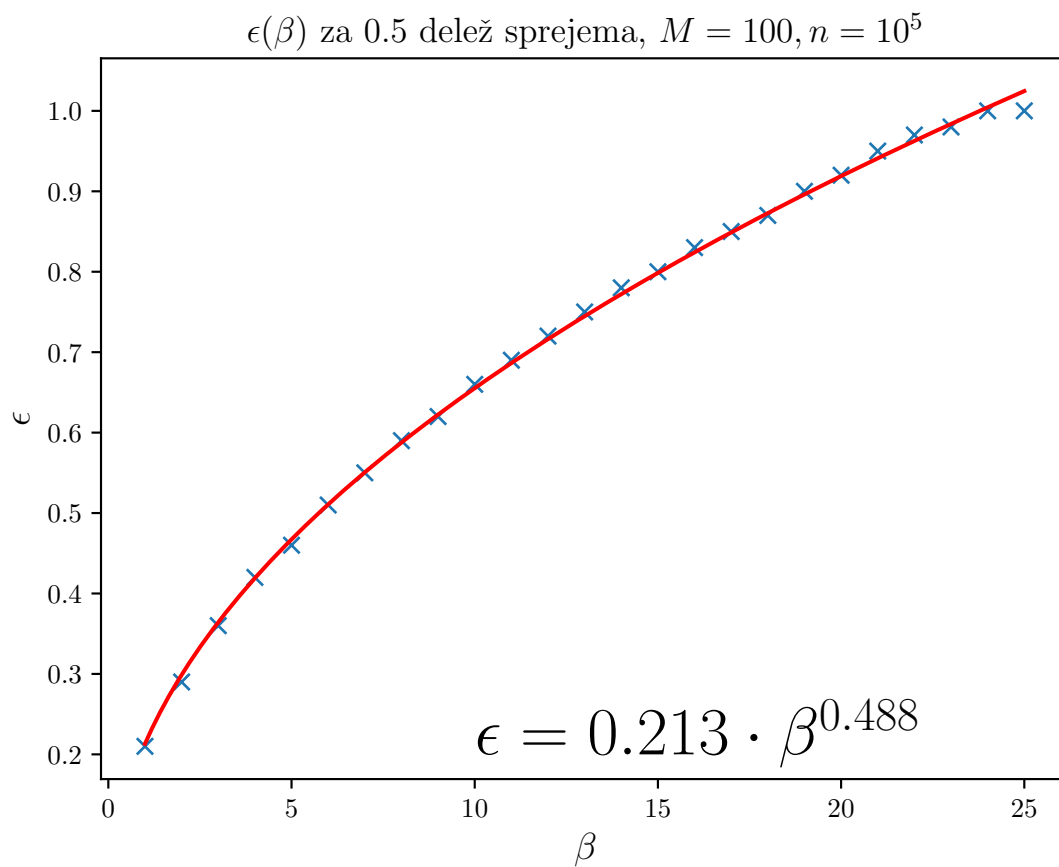


Na sliki je prikazan delež sprejetih potez v odvisnosti od parametra ϵ in β pri fiksnem številu časovnih rezin M . Algoritem sem pognal za n korakov brez kakršnekoli relaksacije. S črno črto je označen delež 0.5, ki ga hočemo. Opazimo, da se z β najprimernejši ϵ kar močno spreminja, in sicer narašča.

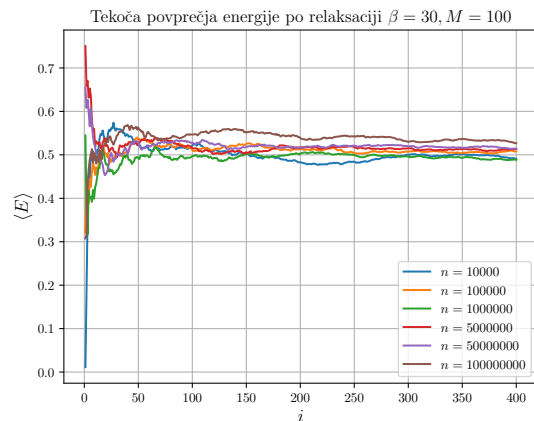
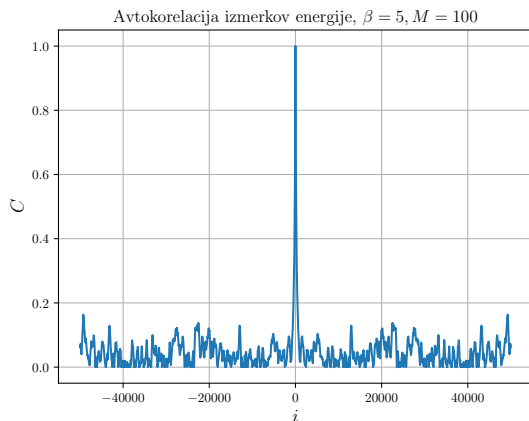


Na sliki je prikazan delež sprejetih potez v odvisnosti od parametra ϵ za več M . Opazimo, da je ϵ močno odvisen tudi od tega parametra. Funkcije, ki opisujejo to odvisnost so za majhne M zelo različne, nekje nad $M = 100$ pa gredo bolj skupaj. Obnašanje je podobno pri dvakrat večjem β .

Glede na zgornje rezultate, sem se odličil, da bom delal z $M = 100$ (Funkcija postane že kar podobna višjim M), ϵ pa spreminjal v odvisnosti od β .

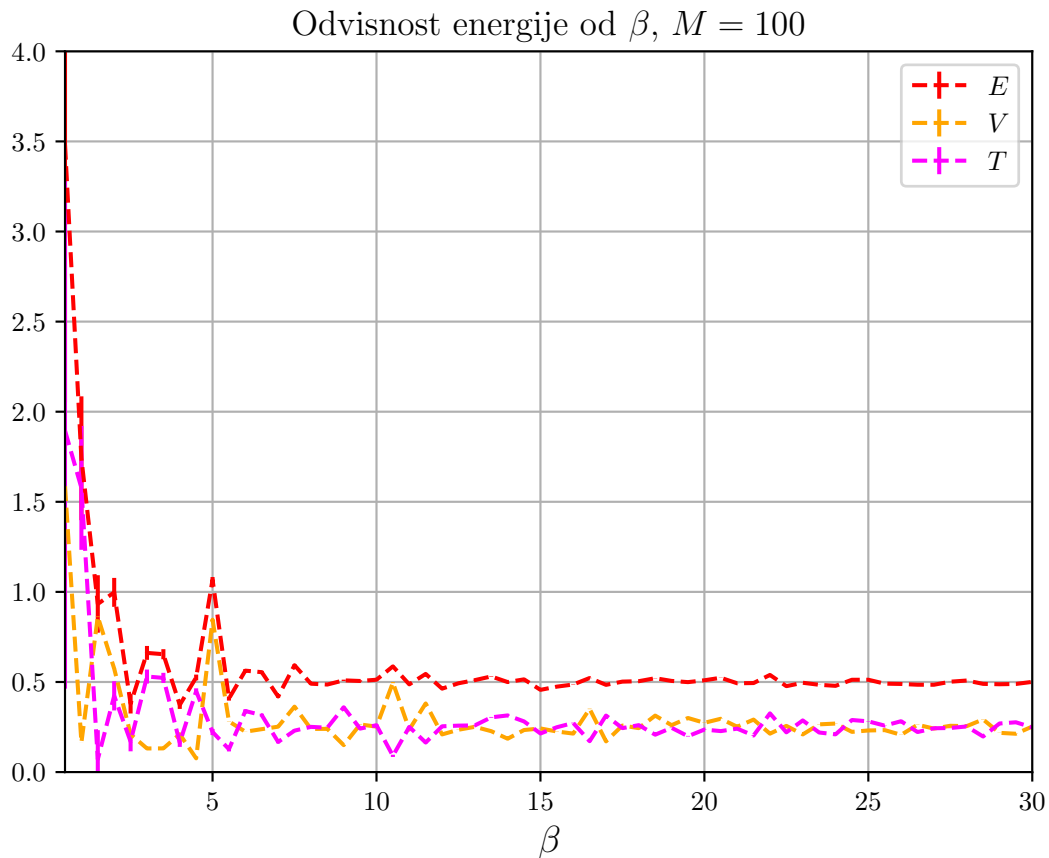


Na sliki je odvisnost optimalnega ϵ od β , pridobljena z izračunom deleža sprejema pri 100 različnih ϵ med 0 in 1. Točke na sliki ustrezajo odmiku od optimalnega deleža sprejetja 0.5 reda 10^{-3} , najboljši fit pa sem poiskal s scipyjevim curvefit.



Na levi je na sliki avtokorelacijska funkcija energije. Na x osi je zaporedno število izmerkov. Korelacija pade na minimum po okoli 2500 izmerkih. Med povprečevanjem bom torej pazil, da pri vzorčenju vzorčim preko dovolj korakov (nekajkrat 2500), da ne povprečujemo samo po močnejše koreliranih konfiguracijah. Na desni je na sliki tekoče povprečje energije. Simulacijo sem najprej pognal za nek relaksacijski čas n in še za naslednjih 10^6 korakov. Od slednjih sem izračunal povprečje vsake 2500-ega. Tekoče povprečje je na sliki. Opazimo, da relaksacijski čas ni zelo dolg, saj dobimo podobne rezultate ne glede na le-tega. Zaradi varnosti bom vseeno vzel $n = 10^6$.

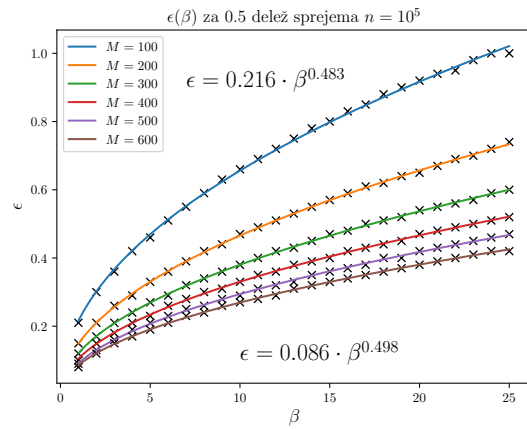
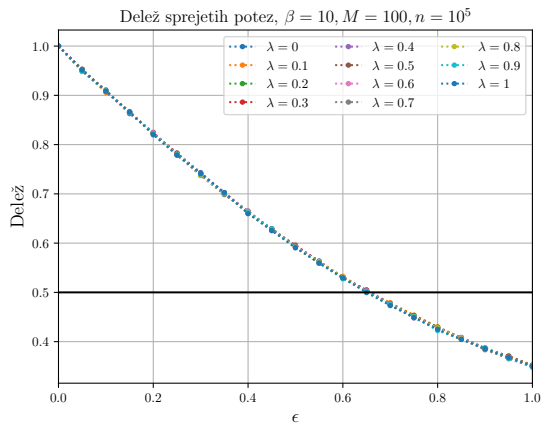
Sedaj lahko izračunamo energijo $\langle H \rangle$ iz partijske funkcije. Pričakovano vrednost potenciala $\langle V \rangle$ sem izračunal kar po povprečevanju $V(q_1)$ po izmerkih, saj je V diagonalen v bazi q . Kinetično energijo potem izračunam kot razliko prejšnjih dveh (in ima posledično največjo napako).



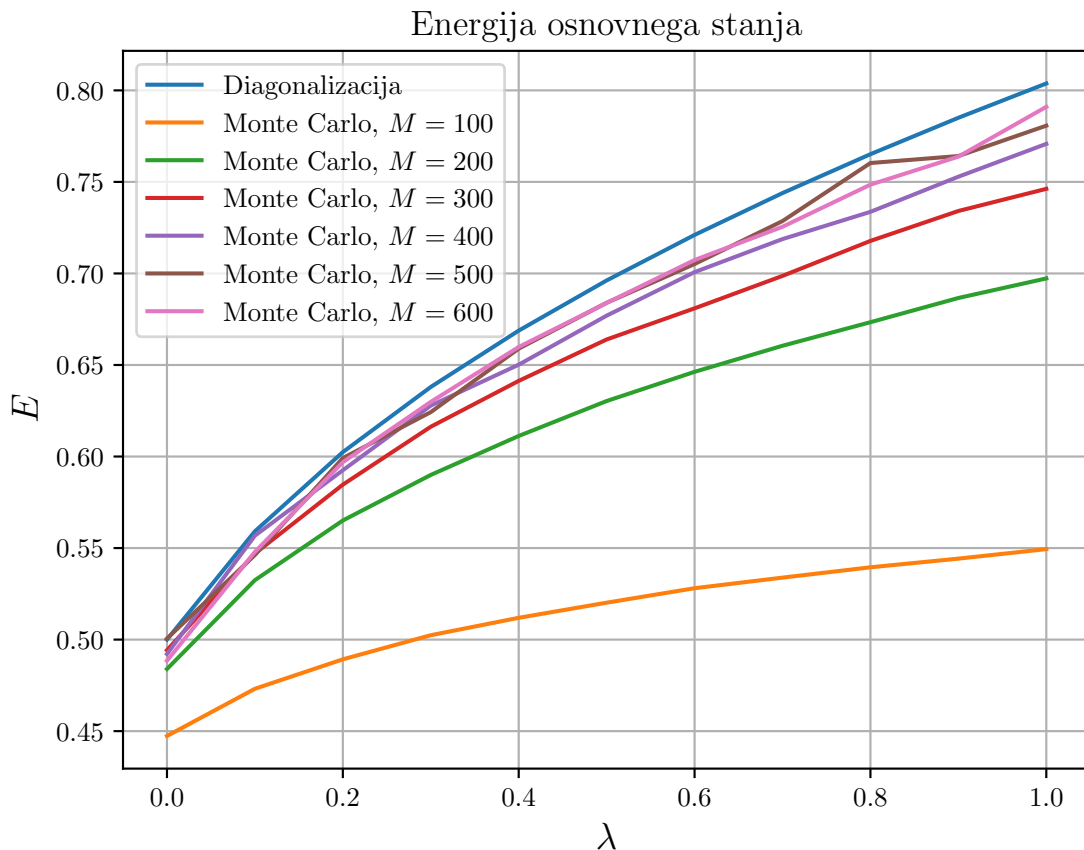
Na sliki je odvisnost energije (in posebej kinetičnega in potencialnega dela) od β . Kot vemo, mora za harmonski oscilator ta biti 0.5 ko gre $\beta \rightarrow \infty$. Za izračun sem najprej metropolisov algoritem poganjal za $n = 10^6$ (relaksacija) potem pa na naslednjih 10^6 korakih vzel vsakega 50-ega in povprečil. Opazimo, da pri večjih β kinetična in potencialna energija prispevata isto k celotni energiji, kar pri nižjih β ni res. Za majhne β je napaka kar velika

Poglejmo si še anharmonski oscilator:

$$H = -\frac{1}{2}\partial_q^2 + \frac{1}{2}q^2 + \lambda q^4$$



Ne levi sem preveril, če moramo pri izbiri ϵ paziti tudi na kakšno odvisnost od λ . Vidimo, da to ne bo potrebno. Na desni sem zaradi razlogov, ki bodo jasni na naslednjem grafu, pogledal kako je z odvisnostjo $\epsilon(\beta)$ za več M . Opazimo, da je funkcijska odvisnost zelo podobna razlika je v glavnem v predfaktorju. V nadaljevanju bom vzel predfaktor kar okoli 1.5 kar se izkaže za dovolj dobro za vse M na sliki.

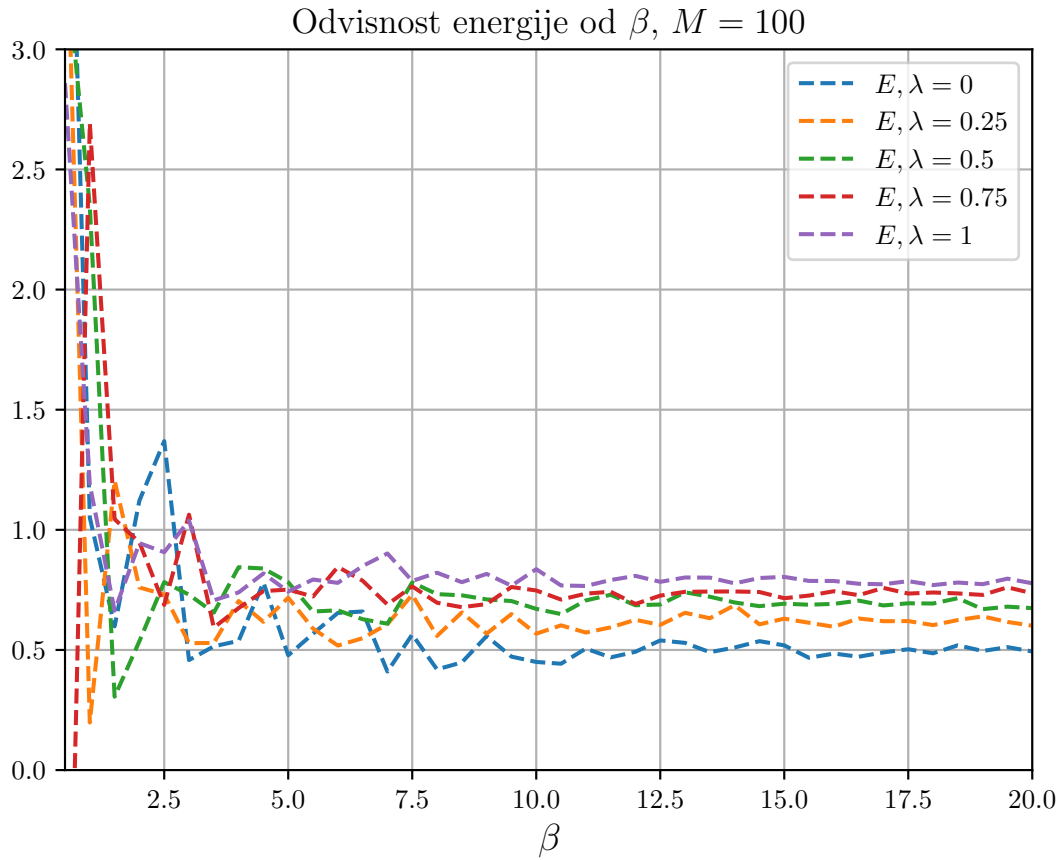


Na sliki je odvisnost energije od λ . Z modro je prikazan rezultat, pridobljen z diagonalizacijo iz prvega poglavja (vstavljen $\beta = 100$, velikost matrike je 500×500) z ostalimi barvami pa energije pri $\beta = 100$ pridobljene z Metropolisovim algoritmom (ohlajanje začeni s $\beta = 1$ in korako po 0.1). Za vsak β med ohlajanjem algoritem poženemo za 10^4 korakov. Pri zadnjem beta pa povprečimo vsakega 50-ega preko 10^7 korakov. Ta števila korakov so pridobljena z nekaj preizkušanja.

Za manjše β se veliki M izkažejo za manj stabilne. Za občutek - tu je energija pri $\beta = 5$ po 10^7 korakih (brez povprečevanja):

M	100	200	300	400	500	600
E	0.78	0.83	1.07	0.33	$1.8 \cdot 10^{10}$	$1.2 \cdot 10^{10}$

Za manjše β se torej spet vrnimo k $M = 100$, kar dovolj dobro deluje. Tako lahko spet narišemo podoben graf kot prej za $\lambda = 0$:



Na sliki je odvisnost energije od β za nekaj več λ . Obnašanje je podobno $\lambda = 0$ le, da so energije nekoliko večje. Za izračun sem najprej 10^7 korakov pustil za relaksacijo, potem pa po naslednjih 10^6 povprečil (vzamemo vsakega 50-ega). Opazimo še, da so energije pri $\beta = 20$ podobne tistim, pridobljenim pri $\beta = 100$ z veliko večjim M , kar lahko služi kot še ena potrditev delovanja metode.