
2.2. Kvantni mnogodelčni problemi - Kubitne spinske verige

MIHA SRDINŠEK

I. UVOD

Pri tej nalogi bomo spoznali metode za numerično preučevanje kvantnih kubitnih spinskih verig. Pri teh verigah gre za to, da se ukvarjamo s sistemom spinov, ki ineragirajo med sabo. Vsak spin lahko opišemo v bazi $|\uparrow\rangle = |0\rangle$ in $|\downarrow\rangle = |1\rangle$, med tem ko sistem večih spinov opišemo v produktni bazi

$$\mathbb{C}^2 \otimes \mathbb{C}^2 \otimes \dots \otimes \mathbb{C}^2 = (\mathbb{C}^2)^{\otimes N} = \mathbb{C}^{2^N}. \quad (1)$$

Zdaj nam hitro postane jasno, da bo opisovanje takih sistemov izredno računsko in časovno zahtevno, saj imamo že pri precej malem številu delcev zelo veliko bazo, ki mora popisati vsako možno pozicijo spinov, sam se spini lahko nahajajo le v superpoziciji različnih sgtanj. Če bi sedaj želeli opisati spinski operator, ki ne deluje na celotni verigi, ampak le na izbranem spinu, se moramo zopet ukvarjati s tenzorskim produktom. Dobimo

$$\sigma_j^\alpha = \mathbb{1}_{2^{j-1}} \otimes \sigma^\alpha \otimes \mathbb{1}_{2^{N-j}}; \quad \alpha \in \{x, y, z, +, -, 0\}, \quad (2)$$

če nam σ^α predstavlja Paulijevo matriko, indeks j pa mesto v verigi na katerega želimo delovati. Da bi se s temi metodami in notacijo spoznali smo se lotili Heisenbergovega modela, pri katerem hamiltonjan zapišemo kot

$$H = \sum_{j=1}^N \vec{\sigma}_j \cdot \vec{\sigma}_{j+1}, \quad (3)$$

s periodičnim robnim pogojem $\vec{\sigma}_{N+1} = \vec{\sigma}_1$. Če bi pri takem modelu želeli izračunati $\exp(zH)$ bi se to precej zapletlo, saj bi morali sešteti čez vse elemente in potem eksponirati matriko. Pri tem na reši split-step metoda, saj v primeru da je število spinov sodo lahko najdemo dva dela hamiltonjana, ki med seboj komutirata.

Ta dela sta sodi in lihi del, kot lahko vidimo na 1. Operatorji na sodih in lihih mestih sami s sabo komutirajo in jih jih lahko zato pri eksponaciji zapišemo kot produkt eksponentov, brez dodatnih popravkov. Torej smo se znašli v identični situaciji kot pri prejšnji nalogi. Imamo hamiltonjan sestavljen iz dveh operatorjev, pri čemer delovanje teh dveh operatorjev znamo zapisati, delovanje njune vsote pa ne. Torej se le poslužimo split-step metode in je.

Split step metoda nam torej delovanje hamiltonjana prelevi v izmenično delovanje sodih in lihih delov, kot si lahko ogledamo na sliki 1. Kako pa poračunamo delovanje sodih in lihih delov? V bazi $\{|00\rangle, |01\rangle, |10\rangle, |11\rangle\}$ lahko delovanje enega operatorja zapišemo kot

$$\vec{\sigma}_1 \cdot \vec{\sigma}_2 = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 \\ 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad (4)$$

In delovanje elementa pri eksponaciji kot

$$e^{z\vec{\sigma}_1 \cdot \vec{\sigma}_2} = e^{-z} \begin{pmatrix} e^{2z} & 0 & 0 & 0 \\ 0 & ch(2z) & sh(2z) & 0 \\ 0 & sh(2z) & ch(2z) & 0 \\ 0 & 0 & 0 & e^{2z} \end{pmatrix}. \quad (5)$$

Od tu dalje postane vse skupaj že precej jasno. Zdaj bi radi izračun H razdelili po split-step shemi na množenje s sodim in lihim delom, z vsakim delom pa delujemo enostavno, par za parom. Najprej dobimo začetno stanje in ga transformiramo z $e^{z\vec{\sigma}_1 \cdot \vec{\sigma}_2}$, potem to novo stanje stransformiramo z $e^{z\vec{\sigma}_3 \cdot \vec{\sigma}_4}$ in tako nadaljujemo po vseh parih. To je zdaj en korak split-step sheme, ki ga izvedemo pri pravi izbiri koeficienta z , nato pa uzvedemo naslednji korak, ki bo verjetno sodi itd. To lahko povzamemo z enačbo

$$|\psi'\rangle_{sodi} = e^{z\sigma_{N-1}\sigma_N} \dots e^{z\sigma_3\sigma_4} e^{z\sigma_1\sigma_2} |\psi\rangle. \quad (6)$$

Vsakega od teh korakov pa izvedemo po enostavnem predpisu

$$\psi'_{s'_1 s'_2 \dots s'_N} = \sum_{s_j, s_{j+1}=0}^1 (e^{z\vec{\sigma}_1 \cdot \vec{\sigma}_2})_{(s'_j, s'_{j+1})(s_j, s_{j+1})} \psi_{s'_1 \dots s'_{j-1} s_j s_{j+1} s'_{j+2} \dots s'_N}, \quad (7)$$

ki dejansko je enostaven, čeprav morda izgleda malo nenavaden. Pove nam, da transformacijo z dvodelčnim operatorjem izvedemo tako, da koeficiente pred novimi baznimi stanji izračunamo s tem predpisom, torej zgolj z ozirom na tista dva spina, ki jih naš operator tipa. Kako sem se tega lotil jaz in kje je prišlo do težav, bom opisal v poglavju o metodi. Pred tem pa si raje pogledjmo še kako bi izračunali pričakovano vrednost nekega operatorja, korelacijsko funkcijo in kako particijsko funkcijo.

Korelacijsko funkcijo izračunamo po predpisu

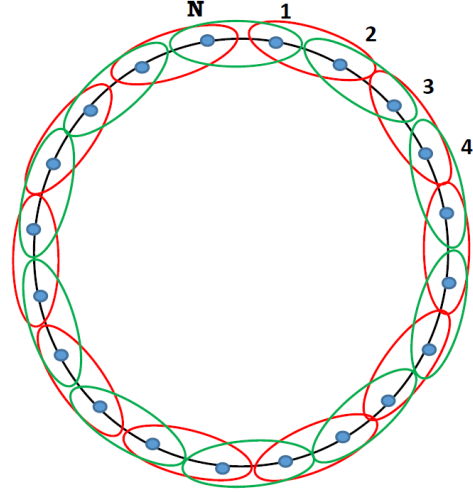
$$\langle x(t)x(0) \rangle = \frac{1}{2^N} \text{Tr}(e^{iHt} x e^{-iHt} x) = \frac{1}{N_\psi} \sum_{j=1}^{N_\psi} \langle \psi_j | e^{iHt} x e^{-iHt} x | \psi_j \rangle, \quad (8)$$

kjer sem pri zadnjem enačaju privzel Levy-jevo lemo, ki je nismo dokazali, da lahko dobimo izjemno natančen rezultat, če namesto prave sledi poračunamo zgolj vrednsot na nekaj 5, 10 naključno izžrebanih začetnih stanjih. Podobno izračunamo tudi $Z(\beta)$ kot

$$Z(\beta) = \frac{1}{N_\psi} \sum_{j=1}^{N_\psi} \langle \psi_j | e^{-\frac{\beta}{2} H} e^{-\frac{\beta}{2} H} | \psi_j \rangle \quad (9)$$

in $\langle X \rangle_\beta$

$$\langle X \rangle_\beta = \frac{1}{N_\psi} \sum_{j=1}^{N_\psi} \langle \psi_j | e^{-\frac{\beta}{2} H} X e^{-\frac{\beta}{2} H} | \psi_j \rangle \quad (10)$$



Slika 1: Na sliki vidimo, da so si sodi in lihi operatorji med seboj disjunktni, če je število spinov sodo.

Kasneje ko bomo preučevali rezultate, nam bo prav prišel analitični rezultat za primer $N = 2$, ki je zelo enostavno rešljiv. Zapišemo lahko

$$H = H_{0,1} + H_{1,0} = \vec{\sigma}_1 \cdot \vec{\sigma}_2 + \vec{\sigma}_2 \cdot \vec{\sigma}_1 = \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 4 & 0 \\ 0 & 4 & -2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}, \quad (11)$$

$$Z(\beta) = \text{Tr}(e^{-\beta H}) = \text{Tr}(e^{2\beta} \begin{pmatrix} e^{-4\beta} & 0 & 0 & 0 \\ 0 & \text{ch}(-4\beta) & \text{sh}(-4\beta) & 0 \\ 0 & \text{sh}(-4\beta) & \text{ch}(-4\beta) & 0 \\ 0 & 0 & 0 & e^{-4\beta} \end{pmatrix}) = 3e^{-2\beta} + e^{6\beta} \quad (12)$$

$$\langle H \rangle_\beta^0 = \text{Tr}(\rho H) = \text{Tr}(e^{2\beta} \begin{pmatrix} e^{-4\beta} & 0 & 0 & 0 \\ 0 & \text{ch}(-4\beta) & \text{sh}(-4\beta) & 0 \\ 0 & \text{sh}(-4\beta) & \text{ch}(-4\beta) & 0 \\ 0 & 0 & 0 & e^{-4\beta} \end{pmatrix} \begin{pmatrix} 2 & 0 & 0 & 0 \\ 0 & -2 & 4 & 0 \\ 0 & 4 & -2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}) \quad (13)$$

$$\langle H \rangle_\beta = \frac{\langle H \rangle_\beta^0}{Z(\beta)} = \frac{1}{Z(\beta)} 6(e^{-2\beta} - e^{6\beta}), \quad (14)$$

kjer nisem počel nič nenavadnega, le v prehodu iz prve v drugo vrstico, sem iz matrike izpostavil 2 in potem uporabil rezultat, ki sem ga uporabil že pri izpeljavi zgoraj. S tem bomo testirali našo metodo in izkazala se bo za precej dobro. Potrebovali bomo še

$$J = \sum_{i=1}^N J_i \quad (15)$$

$$J_i = \sigma_i^x \sigma_{i+1}^y - \sigma_i^y \sigma_{i+1}^x = \quad (16)$$

$$J_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & -2i & 0 \\ 0 & 2i & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad (17)$$

za izračun spinskega toka

$$C_{JJ} = \langle J(t)J(0) \rangle. \quad (18)$$

II. METODE

- Programa sem se lotil tako, da sem podobno kot pri prejšnji nalogi, definiriral dve funkciji z imenom liha in soda, ki sta za argument sprejeli stanje zapisano v bazi urejeni po vrsti tako, kot števila razporejena od najmanjšega do največjega v binarnem zapisu. Stanje ko so vsi spini obrnjeni navzgor sem torej opisal s številom 0, i v binarnem zapisu izgleda kot ...000., stanje stanje z vsemi spini gor in prvim spinovm dol, sem opisal s številom 1, ki v binarnem zapisu izgleda kot ...001. Na tak način sem se moral samo poigrati z binarnim zapisaom indeksa seznama, da sem opravljal operacije na stanju.

Algoritem je potekel tako, da je prejel začetno stanje, potem pa za vsak par pogljal for zanko po vseh elementih začetnega stanja. Pri vsakem elementu sem indeks elementa prevedel v binarno število in pogledal kako izgledata števki na izbranem paru. Če sta bili števki enaki, sem ju le pomnožil z eksponentom, če pa sta bili različni, sem na njima izvedel operacijo, pri čemer sem novi indeks poiskal tako, da sem skonstruiral število 0000110000 z enicami na mestu para in s tem številom izvedel XOR vrata na indeksu našega elementa. Potem sem rezultat shranil, in na novem stanju ponovil postopek za

naslednji par, dokler nisem šel čez vse pare. Tako je recimo izgledala funkcija lihi ali sodi. Potem pa sem enostavno uporabljal split step metodo.

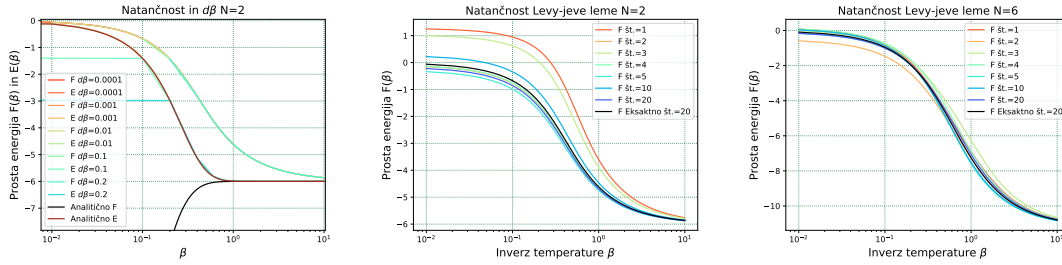
Da si lažje predstavljamo, operacija $1746326^{192} = 1746262$ poteka kot

$$\begin{aligned} &00110101010010110010110^{\wedge} \\ &00000000000000011000000 = \\ &00110101010010101010110. \end{aligned}$$

- Zanimivo je še bilo kaj sem najprej naredil narobe, ko sem zapisal algoritem za izračun Hamiltonjana. Hamiltonjan je potrebno izračunati tako, da delujemo na enem paru in stečemo čez vse elemente začetnega stanja, nato pa ne shranimo rezultata kot novo stanje, ampak smo prištejemo rezultat praznemu seznamu. Za naslednji par naredimo isto in prištejemo rezultat prejšnjemu rezultatu in tako dalje, čez vse pare.
- Naključno začetno stanje sem izračunal tako, da sem po gaussovski porazdelitvi z povprečjem 0 in deviacijo 1 izžrebal eno realno in eno imaginarno vrednost za vsak element začetnega stanja, nato pa sem stanje normaliral s skalarnim produktom (torej sem pomnožil stanje s konjugiranim stanjem). To sem naredil zato, ker nam je to svetoval profesor na predavanjih.

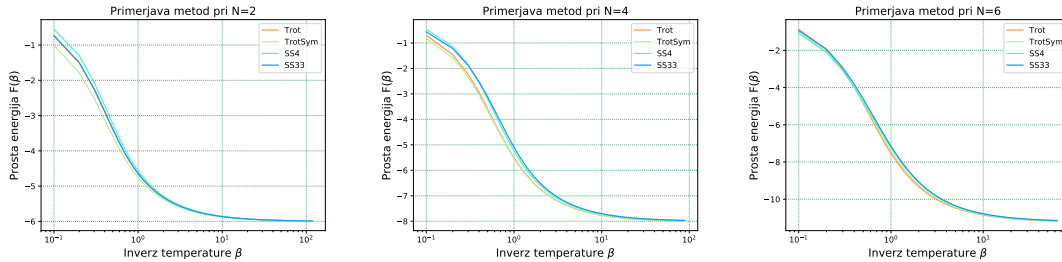
III. NATANČNOST

najprej sem samo preveril mojo metodo. Zanimalo me je katera izmed integracij bo zame bolj uporabna, kako natančna je metoda v odvisnosti od velikosti koraka in kako natančna v odvisnosti od tega koliko začetnih stanj izžrebamo. Na srečo sem zaradi tega takoj opazil, da metoda ne deluje, saj so mi različni integratorji dali različne rezultate. Potem sem izvedel nekaj popravkov in rezultati so še vedno bili napačni. Po zelo dolgih obdobjih iskanja napake sem ugotovil, da sem v nekaterih primerih deloval na stanjih tako, da sem zamenjal cosh in sinh, kar je rešilo vse probleme. Od takrat dalje sme pridelal veliko rezultatov in se ukvarjal z natančnostjo, pri čemer sem večinoma ugotovil, da je relativno nepomembno kaj si izbrem za dolžino koraka in kolikokrat žrebam začetno stanje. Tak rpiemr vidimo na slikah 2, kjer vidiom, da je korak precej irelevanten, da dobimo energijo zelo natančno, prosto energijo malo manj, in da število žrebanih stanj ne vpliva tako bistveno na rezultat. Včasih je celo večje število izžrebanih stanj dalo slabši rezultat kot manjše.



Slika 2: Slike prikazujejo kako se z našo metodo izračunana prosta energija v odvisnosti od inverzne temperature $F(\beta)$ spreminja z tem ko spreminjamo natančnost. Na levi sliki spreminjamo velikost koraka pri spit-step metodi in to primerjamo z analitično rešitvijo, ki smo jo izpeljali v uvodu. Pri tem tiste krivulje zgoraj predstavljajo F , tiste nižje pa E , kar vidimo, če se pogledimo v barve. Energijo dobimo povsem isto, kot jo dobimo analitično, med tem ko se prosta energija obnaša drugače od pričakovanega. Na desnih slikah spreminjamo število izžrebanih začetnih stanj in to primerjamo z eksaktno rešitvijo, kjer numerično izračunamo sled.

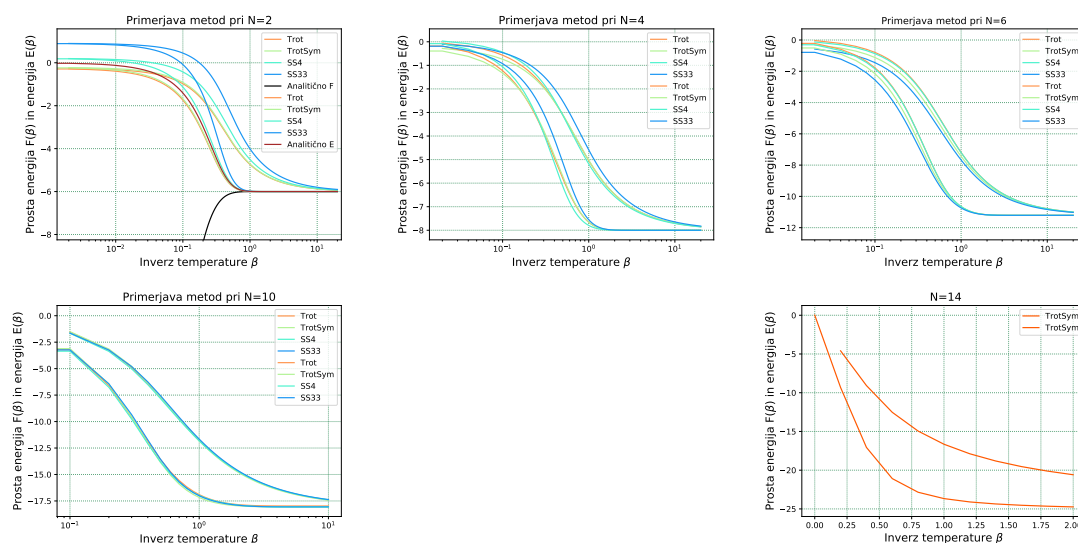
Poleg tega nas zanima kako zanesljive so različne split-step sheme. Zato sem prosto energijo in energijo izračunal pri istem N in vseh ostalih parametrih enakih, z različnimi metodami, ker vidimo na slikah 3.



Slika 3: Slike prikazujejo kakšen rezultat dajo pri istih začetnih pogojih različne split-step sheme.

Pod to rubriko si potem kar pogledamo rezultate ze energijo in prosto energijo v odvisnosti od β , saj sem ves čas uporabljal vse metode, da se vidi kako se razlikujejo med seboj metode. To vidimo na slikah 4. Vidimo, da v limiti ko gre β proti neskončno dobimo isti rezultat pri vseh metodah, poleg tega ugotovimo, da je pričakovana energija v tej limiti enaka prosti energiji. Ugotovimo tudi, da je v tem primeru eksaktni rezultat isti kot numerični rezultat v tej limiti.

S tem smo si že ogromno ogledali. S tem smo našli odgovor na prvi dve vprašanji. Amapak

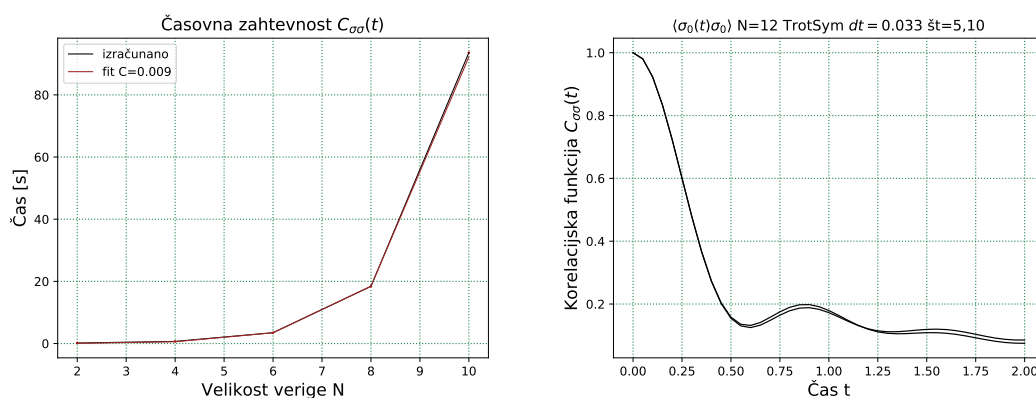


Slika 4: Slike prikazujejo kakšen rezultat dajo pri istih začetnih pogojih različne split-step sheme.

vseeno sem za te rezultate porabil čas, saj so same metode precej zahtevne in ni vse delalo že ob prvem zagonu, ker sem vsakič naredil kakšno butasto napako, pod drugo pa tudi za večje N metoda vzame kar nekaj časa.

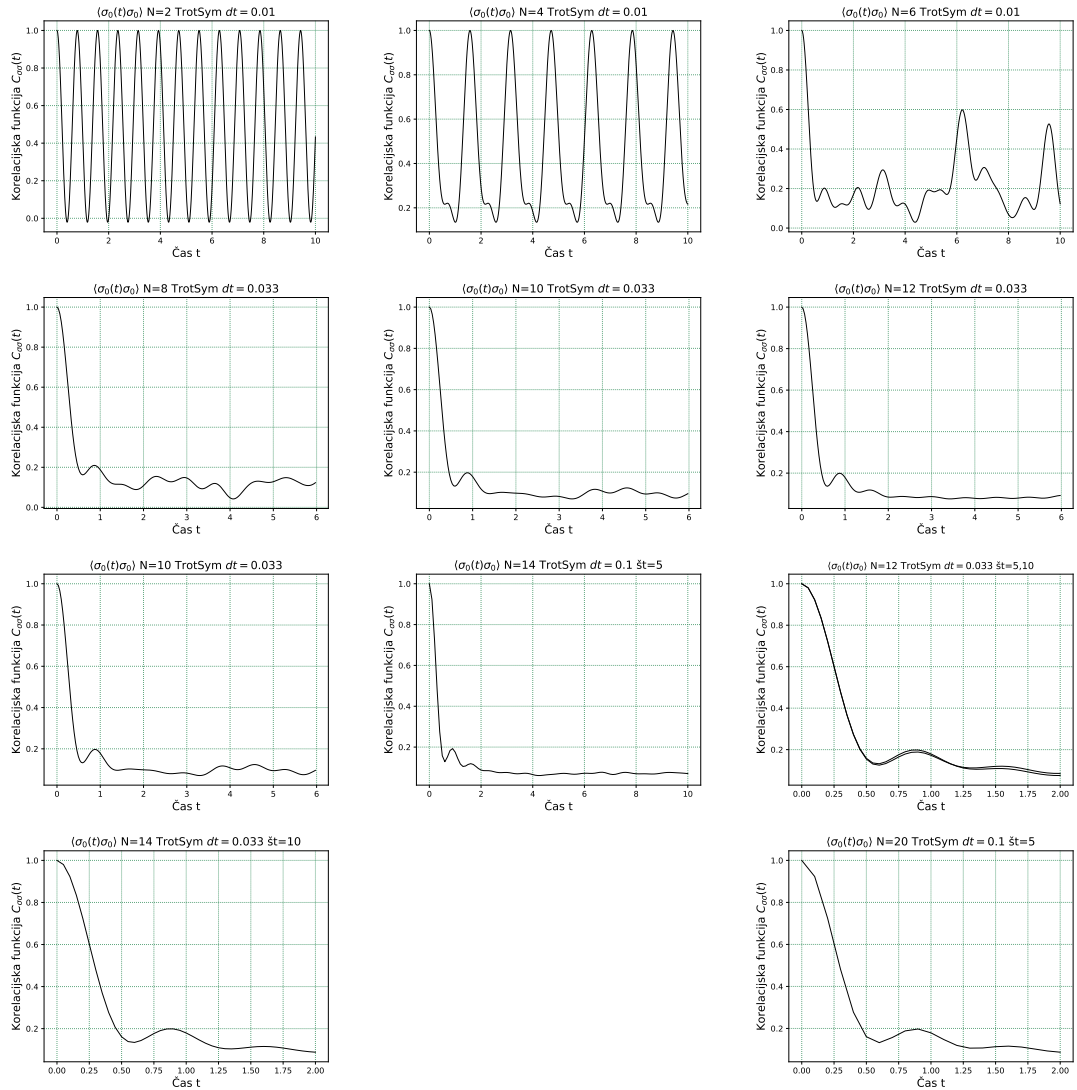
IV. AVTOKORELACIJSKA FUNKCIJA LOKALNE MAGNETIZACIJE

Zdaj si lahko pogledamo avtokorelacijsko funkcijo, ki smo jo že prej omenjali. Ta stvar se je računala še relativno hitro in zato sem enostavno poračunal rezultat pri različnih N do čim daljših časov pri čim višji natančnosti. Kako sem iskal čas računanje sem prikazal na sliki 5, saj sem oral biti pri tej metodi precej previden, da nisem zagnal programa, ki bi trajal par dni. Vse skupaj je potekalo počasi, ker se nisem toliko potrudil pri kodi in predvsem, sem zapisal program v jeziku *python*. Za višjo natančnost in bolj zanimive rezultate bi moral program prepisati v *C*, a sem predolgo odlašal in mi je zmanjkalo časa.



Slika 5: Leva slika prikaže kako se čas računanja rezultata spreminja z velikostjo verige, desna slika pa prikaže kako na rezultat vpliva če namesto 10 žrebcev, kot ponavadi, izvedemo le deset žrebcev..

Avtokorelacijske funkcije pa so potem prikazane na slikah 6



Slika 6: Slike prikazujejo avtokorelacijske funkcije $\langle \sigma_0^z(t) \sigma_0^z(0) \rangle$ za različne velikosti verig in različne natančnosti.

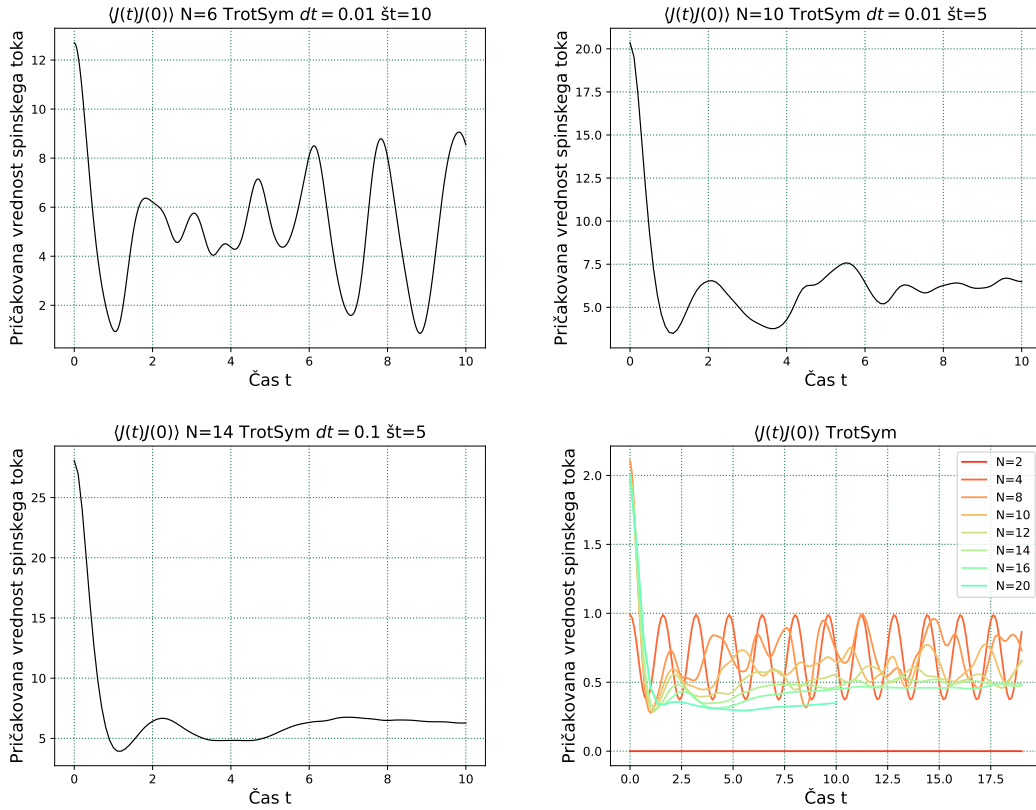
Rezultati so zelo zanimivi in delno tudi pričakovani. Zapisal sem tudi program za ostale spine, ampak so rezultati pravzaprav isti, kar bi seveda pričakovali, ker imamo periodične robne pogoje.

V. AVTOKORELACIJSKA FUNCKIJA SPINSKEGA TOKA

Od tu naprej pravzaprav v veliki meri ponovimo, kar smo naredili pri avtokorelaciji okalne magnetizacije, le da se moramo malo bolj potruditi za izračun J , ker ga na vajah nismo izpeljali, zato sem ga izpeljal v uvodu in od tam dalje uporabljal. V zapiskih, ki so objavljeni na internetu je predstavljena neka hitrejša možnost, ki je osebno nisem znal izpeljati, zato sem se držal definicije. Verjetno pa bi glede na trditve iz teksta moral dobiti enake rezultate, če bi uporabil

$$\langle J(t)J(0) \rangle = \sum_{r=0}^{N-1} \langle J_{r+1}(t)J_1(0) \rangle, \quad (19)$$

pri čemer je N velikost verige. Tudi v tem primeru sem si najprej pogledal kako izgledajo rešitve in si jih izrisal. Vidimo jih lahko na slikah7. Nasploh podobno kot prej vidimo da z večanjem sistema preidemo v stanje, ko nehamo dobivati neko periodično rešitev, temveč neko rešitev, ki precej hitro pada.



Slika 7: Slike prikazujejo avtokorelacijske funkcije $\langle J(t)J(0) \rangle$ za različne velikosti verig in različne natančnosti.

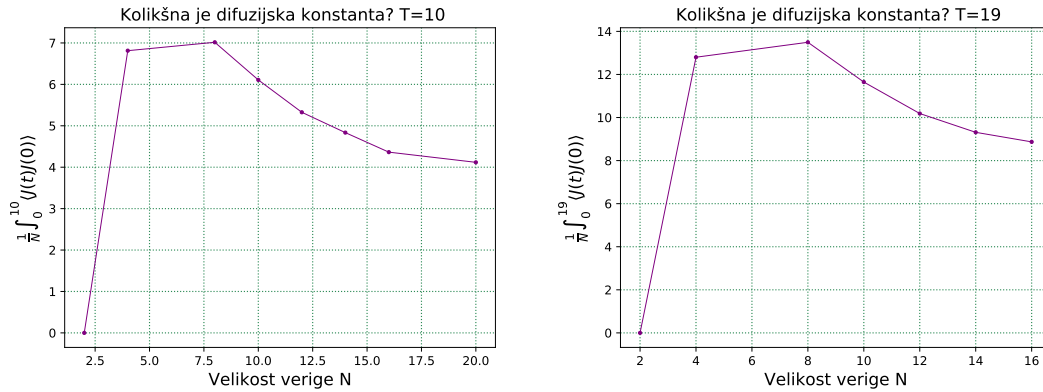
Zdaj se tako znajdemo pred zanimiv delom naloge, ko iščemo difuzijsko konstanto definirano kot

$$D = \lim_{t \rightarrow \infty} \lim_{N \rightarrow \infty} \frac{1}{N} \int_0^t \langle J(t')J(0) \rangle dt'. \quad (20)$$

Tu se vse kar hitro zaplete, saj je ta celotna limita zelo nenavadna. Pri vsaki velikosti verige je namreč ta integral načeloma povsod povsem drugačen. Torej je treba izbrati neko velikost verige, poračunati različne čase in potem to početi pri vseh N . Pri tem je velik problem, ker vsaj zaenkrat še ni na vidiku, da bo integral pri izbranem N konvergirал. Nasploh si pogledjmo

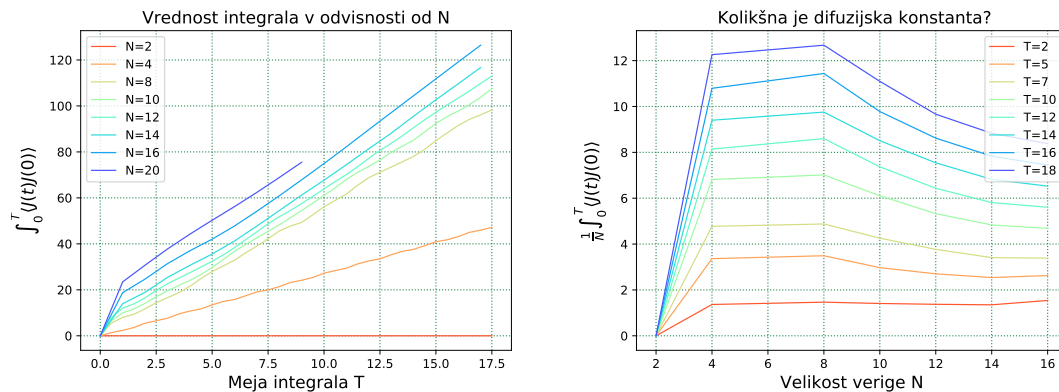
kako se spreminja vrednost integrala (ulomljeno N) do nekega izbranega časa v odvisnosti od izbire N . To vidimo na sliki 8.

Kar vidimo je, da vrednosti res očitno konvergirajo z limito N proti neskončno. Problem je le, da se ta vrednost, vsaj tako izgleda, približuje različnim vrednostim pri različnih izbirah T . Zaenkrat res težko ocenim kakšna naj bi bila difuzijska konstanta.



Slika 8: Slike prikazujejo rezultat $\frac{1}{N} \int_0^T \langle J(t)J(0) \rangle$, pri čemer se spreminja velikost verige za integracijo pri vsakem N ! Iz slike je res očitno, da pri obeh T vrednost konvergira k neki vrednosti in to precej hitro.

Ker s temi rezultati nisem zadovoljen sem izrisal rezultata za še večjo izbiro časov preveril tudi kako se obnaša integral, akr vidimo na slikah



Slika 9: Leva slika prikazuje vrednost integrala v odvisnosti od zgornje meje za različne velikosti verig N . Pri tem integrala ne delim z N ! Desna slika prikazuje kako se integral ulomljeno N spreminja s časom in velikostjo verige. Očitno je da se neka konvergenca dogaja. Stavil bi na vrednost tam okoli štirice.

Načeloma bi lahko samo aproksimirali integral s premico in potem pogledali kaj napovemo za difuzijsko konstanto.