

Univerza v Ljubljani
Fakulteta *za matematiko in fiziko*



VRM
DMRG Metode

Andrej Kolar - Požun

30. 5. 2019

Uvod

V tej nalogi bomo gradili na našem znanju iz MPA iz prejšnje naloge. Za začetek si pogledjmo, kako izračunamo skalarni produkt

$$\langle \phi | \psi \rangle$$

med dvema stanjema, če sta ti podani kot kanonični matrično produktni nastavek. Recimo, da je ψ podan z matrikami $A_s^{(j)}$, ϕ pa z matrikami $\tilde{A}_s^{(j)}$. Skalarni produkt najenostavneje izračunamo s pomočjo sledečih tako imenovanih prehodnih matrik:

$$\begin{aligned} T^{(j)} &= \sum_{s=0}^{d-1} \tilde{A}_s^{(j)} \otimes A_s^{(j)}, j = 2, \dots, n-1 \\ L_{(k,l)} &= \tilde{A}_k^{(1)} A_l^{(1)} \\ R_{(k,l)} &= \tilde{A}_k^{(n)} A_l^{(n)} \\ \langle \phi | \psi \rangle &= L T^{(2)} T^{(3)} \dots T^{(n-1)} R \end{aligned}$$

V našem primeru bomo obravnavali dvonivojski sistem - spin $1/2$, torej bo $d = 2$. Posplošitev na računanje skalarnih produktov tipa $\langle \phi | O_j | \psi \rangle$, kjer je O_j lokalni operator, ki deluje na mestu j je enostavna: prehodno matriko na j -tem mestu nadomestimo z

$$V_{(k,l),(k',l')}^{(j)} = \sum_{s,s'=0}^{d-1} O_{s,s'}(\tilde{A}_s^{(j)})_{k,k'}(A_{s'}^{(j)})_{l,l'}$$

in lahko izračunamo

$$\langle \phi | O_j | \psi \rangle = L T^{(2)} \dots T^{(j-1)} V^{(j)} T^{(j+1)} \dots T^{(n-1)} R$$

Analogno velja za dvodelčni operator, le da moramo potem matriko V v zgornji produkt vstaviti dvakrat:

$$\langle \phi | O_j O_k | \psi \rangle = L T^{(2)} \dots T^{(j-1)} V^{(j)} T^{(j+1)} \dots T^{(k-1)} V^{(k)} T^{(k+1)} \dots T^{(n-1)} R$$

Poleg enostavnega računanja skalarnih produktov, lahko z MPA tudi učinkovito simuliramo propagator. To je tako imenovani TEBD algoritem, ki poteka takole:

Recimo, da imamo lokalni dvodelčni Hamiltonjan:

$$H = \sum_j^{n-1} h_{j,j+1}$$

Hamiltonjan lahko razdelimo na dva dela, v katerih vsi členi med seboj komutirajo, kot smo to naredili že v 4. nalogi. Propagator lahko potem napišemo s pomočjo Trotterjevega razcepa

$$U(z) = \left(\prod_{k=1}^{n/2} U_{2k-1,2k}(z/m) \prod_{j=1}^{(n-1)/2} U_{2j,2j+1}(z/m) \right)^m,$$

oziroma podobne sheme. Tekom naloge bom uporabljal $S_4(z)$ shemo. Parameter z je $-i\tau$, če nas zanima časovni razvoj, oziroma $-\beta$ če nas zanima časovni razvoj v imaginarnem času, kot smo že videli v prejšnjih nalogah.

Zanima nas, kako naj s propagatorjem delujemo na naše stanje, zapisano v MPA obliki. Priročno je stanje zapisati z:

$$\psi_{s_1, \dots, s_n} = B_{s_1}^{(1)} \lambda^{(1)} \dots \lambda^{(n-1)} B_{s_n}^{(n)},$$

kjer so λ Schmidtovi koeficienti, matrike B pa preprosto dobimo iz A :

$$\begin{aligned} B_{s_1}^{(1)} &= A_{s_1}^{(1)} \\ B_{s_j}^{(j)} &= [\lambda^{(j-1)}]^{-1} A_{s_j}^{(j)}, j = 2, \dots, n \end{aligned}$$

Zanima nas, kako stanje

$$|\psi'\rangle = U_{j,j+1}|\psi\rangle$$

zapisati v kanoničnem MPA. Lahko zapišemo:

$$\psi'_{s_1, \dots, s_n} = B_{s_1}^{(1)} \lambda^{(1)} \dots \lambda^{(j-1)} B_{s_j, s_{j+1}}^{(j, j+1)} \lambda^{(j+1)} \dots \lambda^{(n-1)} B_{s_n}^{(n)},$$

kjer je

$$B_{s_j, s_{j+1}}^{(j, j+1)} = \sum_{s'_j, s'_{j+1}} U_{(s_j, s_{j+1}), (s'_j, s'_{j+1})}^{(2)} B_{s'_j}^{(j)} \lambda^{(j)} B_{s'_{j+1}}^{(j+1)}$$

Definirajmo matriko

$$Q_{(k_{j-1}, s_j), (k_{j+1}, s_{j+1})} = \lambda_{k_{j-1}}^{(j-1)} \left(B_{(s_j, s_{j+1})}^{(j, j+1)} \right)_{k_{j-1}, k_{j+1}} \lambda_{k_{j+1}}^{(j+1)}$$

In naredimo njen SVD razcep

$$Q = UDV^\dagger$$

s pomočjo katerega definiramo MPA po delovanju operatorja:

$$\begin{aligned} (B_{s_j}^{(j)})'_{k_{j-1}, k_j} &= (\lambda_{k_{j-1}}^{(j-1)})^{-1} U_{(k_{j-1}, s_j), k_j} \\ \lambda_{k_j}^{(j)'} &= D_{k_j, k_j} \\ (B_{s_{j+1}}^{(j+1)})'_{k_j, k_{j+1}} &= V_{k_j, (k_{j+1}, s_{j+1})}^\dagger (\lambda_{k_{j+1}}^{(j+1)})^{-1} \end{aligned}$$

Ostale $\lambda^{(i)}$ in $B_{s_i}^{(i)}$ ostanejo isti.

Naloga

Rezanje

Preden začnemo opazimo, da bomo tekom algoritma prisiljeni rezati majhne singularne vrednosti. To pomeni, da bomo po SVD razcepu neke matrike

$$X = UDV^\dagger$$

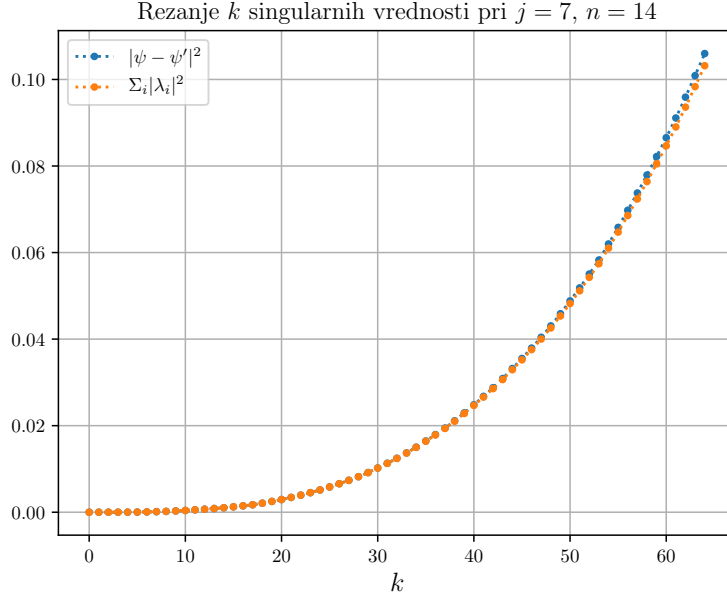
majhne singularne vrednosti (diagonalce D -ja) enostavno zavrgli - zmanjšali velikost matrike D . Seveda je potem treba primerno zmanjšati tudi širino matrike U in višino matrike V^\dagger . Najprej bo rezanje potrebno pri SVD razcepih, ki jih delamo ko iz nekega stanja pridobimo matrike $A_{s_j}^{(j)}$. Namreč, v TEBD algoritmu nastopajo Schmidtovi koeficienti v imenovalcu, kar bo problematično, če bodo ti zelo majhni. Tekom konstrukcij matrik $A_{s_j}^{(j)}$ bom zato odstranil singularne vrednosti, ki bodo manjše od 10^{-3} .

Rezati moramo tudi tekom TEBD algoritma, saj se velikost matrike Q pri vsakem koraku podvoji. Ker nečemo imeti opravka z ogromnimi matrikami, si bomo izbrali neko maksimalno velikost matrike $M \times M$, in SVD razcep matrike Q odrezali, če je singularnih vrednosti več kot M .

Označimo z $|\psi'\rangle$ stanje, ki ga dobimo, če tekom konstrukcije MPA stanja $|\psi\rangle$ odrežemo nekaj singularnih vrednosti λ_i . Velja:

$$||\psi - \psi'||^2 = \sum_i |\lambda_i|^2,$$

kjer gre desna vsota po vseh singularnih vrednosti, ki smo jih odrezali. Vidimo, da je napaka, ki jo z rezanjem naredimo odvisna od velikosti odrezanih singularnih vrednosti. Na spodnjem grafu sem formulo za napako še eksplicitno preveril s tem, da sem naključno generiral neko stanje ga na določen način odrezal in pogledal napako:



Na grafu sta prikazani napaka zaradi rezanja ter vsota kvadratov odrezanih lambd. Stanje, ki ga režem je naključno stanje $n = 14$ spinov. Na x osi je prikazano število odrezanih singularnih vrednosti (zmeraj sem odrezal najmanjše), pri čemer vedno režem $\lambda^{(7)}$. Opazimo, da se napaka praktično sklada z velikostjo odrezanih lambd. Poleg tega vidimo, da si lahko privoščimo (vsaj v tem primeru) kar veliko odrezati, saj skrajno desna točka predstavlja odstranitev polovice vseh singularnih vrednosti.

Osnovno stanje AFM

Za začetek poiščimo osnovno stanje antiferomagnetne Heisenbergove verige. Začnimo z naključno generiranim stanjem $|\psi_0\rangle$. Predstavimo ga v MPA obliki in nanj delujemo z propagatorjem v imaginarnem času:

$$\exp(-\beta H)|\psi_0\rangle,$$

kar lahko spet predstavimo kot MPA s pomočjo Trotterjeve formule in TEBD algoritma, kot je napisano v uvodu. Osnovno energijo potem dobimo preko naslednje zveze:

$$\langle\psi_0|e^{-\beta H}|\psi_0\rangle = e^{-\beta E_0} + \mathcal{O}(e^{-\beta(E_1-E_0)}).$$

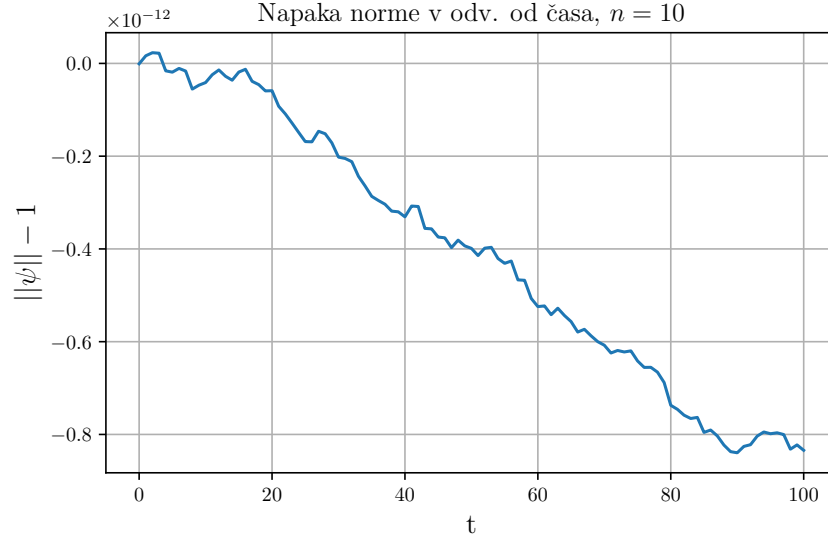
Levo stran znamo izračunati, za velike β bo prvi člen na desni prevladal in bo veljalo

$$\log(\langle\psi_0|e^{-\beta H}|\psi_0\rangle) \approx -\beta E_0$$

Na zgornjo enačbo pa lahko preprosto fittamo premico. Hkrati lahko dobimo osnovno stanje enostavno z

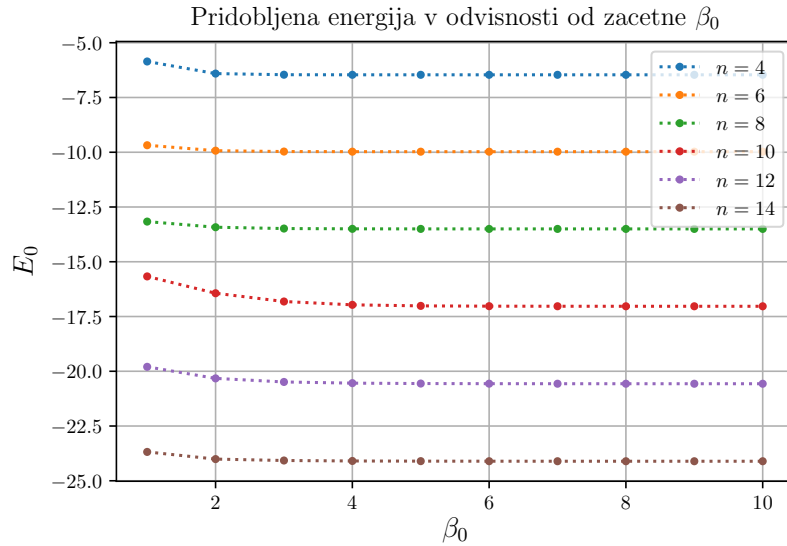
$$|E_0\rangle = \lim_{\beta \rightarrow \infty} e^{\beta E_0} e^{-\beta H} |\psi_0\rangle$$

Preden začnemo z iskanjem stanja, si pogledjmo le časovni razvoj naključnega stanja in preverimo, da se norma ohranja, saj je algoritem dovolj kompliciran, da se lahko pojavijo kakšne napake.



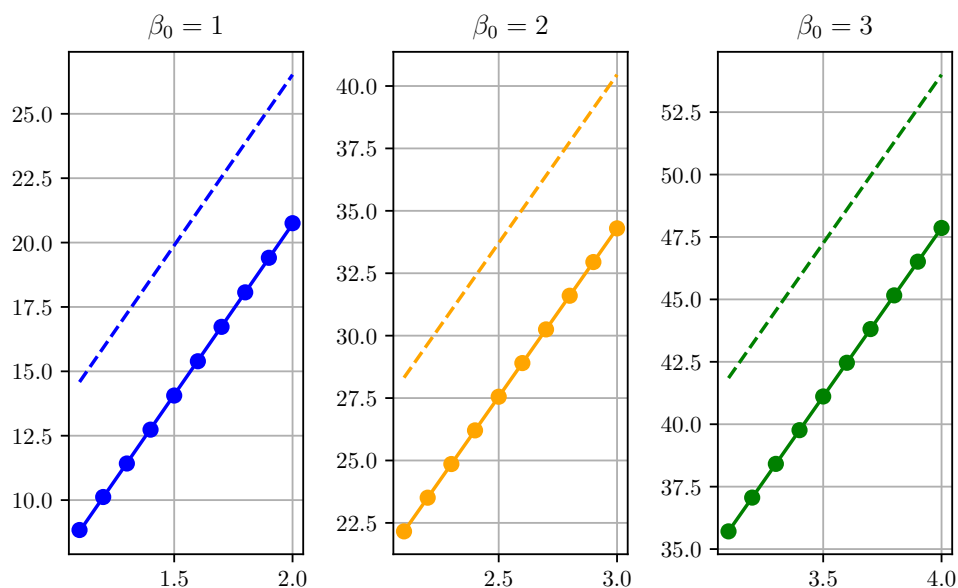
Na sliki je časovna odvisnost napake norme glede na pričakovano vrednost $||\psi|| = 1$, saj začnemo z normiranim stanjem za primer stanja $n = 10$ spinov. Vidimo, da se norma ohranja, kar kaže na pravilno delovanje TEBD algoritma.

Naslednje vprašanje je izbira primernega časovnega koraka $\Delta\beta$. Uporabil bom kar korak, ki se je izkazal za dobrega pri 4. nalogi in sicer $\Delta\beta = 0.1$. Poleg tega se moramo vprašati še, kdaj je β dovolj velik, da lahko rečemo, da je desna stran zgornje enačbe res le en člen, ki eksponentno pada. Pri 4. nalogi smo ugotovili, da smo v osnovnem stanju že pri β okoli 2.5, vendar nas tu zanima še oblika desne strani zgornje enačbe od β , saj moramo za razliko od 4. naloge tu energijo pridobiti s fitom in jo težje zgolj razberemo iz grafa.



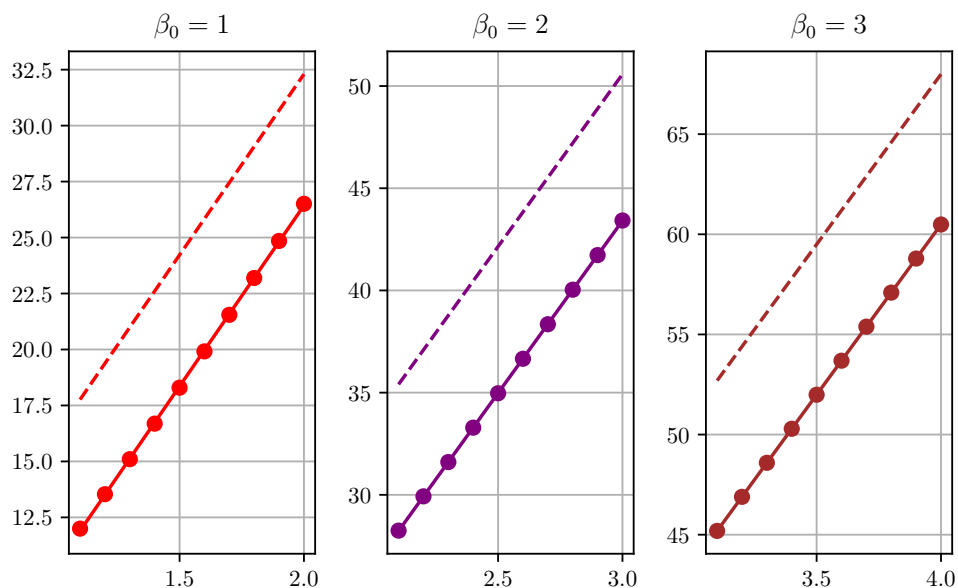
S fittom pridobljena energija v odvisnosti od izbranega β_0 , kjer začnemo fittati. Vedno sem fittal zgolj na intervalu $[\beta_0, \beta_{0+1}]$. Opazimo, da energija hitro skonvergira - za večino je dovolj že $\beta_0 = 2$, le $n = 10$ primer potrebuje okoli $\beta_0 = 3$.

Primerjava fitta s podatki, $n = 8$



Na sliki vidimo podatke in pripadajoče fitte za različne β_0 . S točkami so označeni podatki, s polno črto je označen fit premice, s črtkano črto pa je označen fit, kjer smo upoštevali le naklon premice, njeno presečišče z y osjo pa dali na 0, dejstvo da premice ne seka dejansko 0 je posledica tega, da desna stran enačbe ni res samo eksponent ampak imamo tudi druge člene. Vidimo, da se polna premica izjemno dobro prilega že pri $\beta_0 = 1$, kjer se energija sicer še ni ustalila, vendar je vseeno zelo blizu pravi vrednosti.

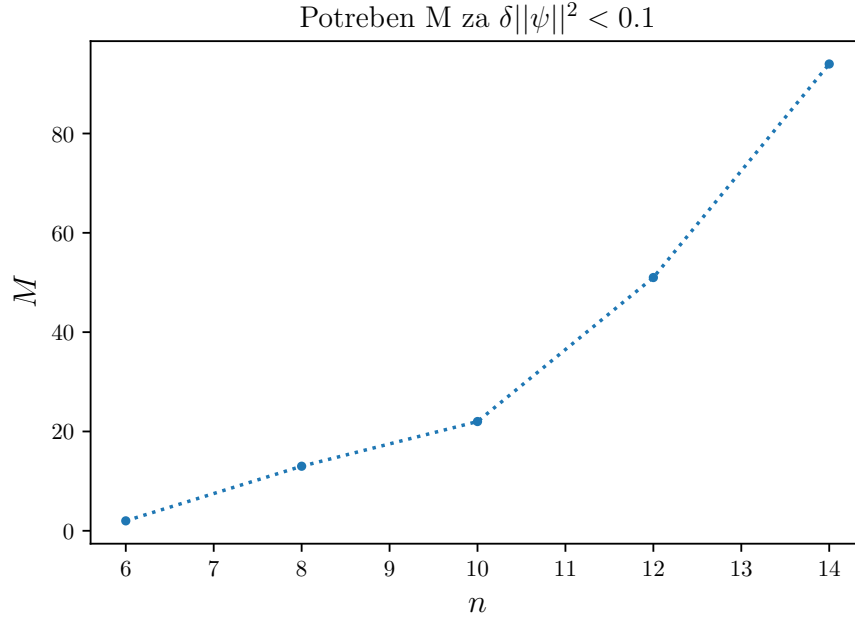
Primerjava fitta s podatki, $n = 10$



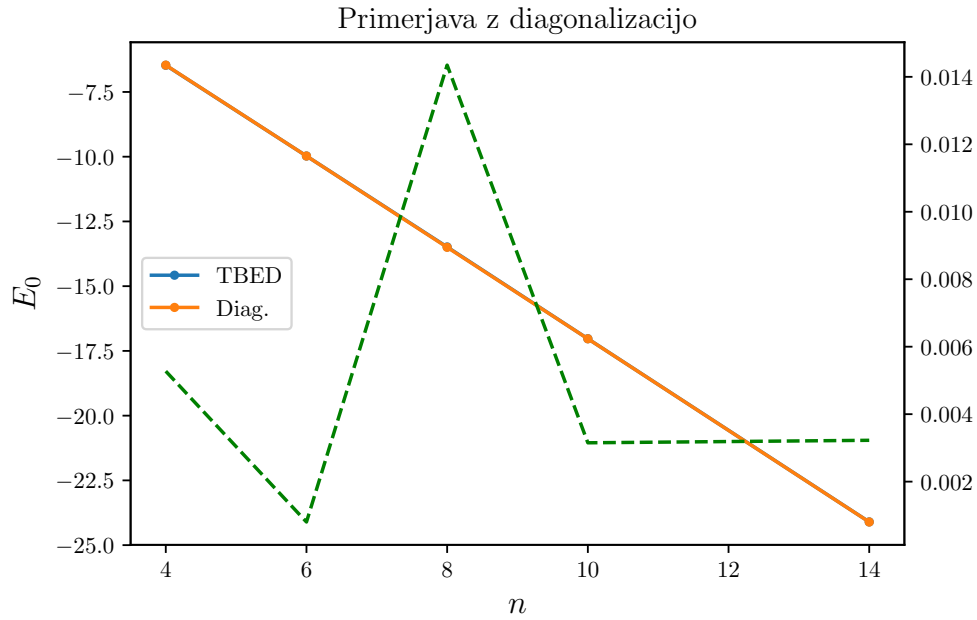
Tukaj gledamo primer $n = 10$, ki se je na predprejšnjem grafu najpočasneje približal pravi energiji. Kljub temu opazimo, da ga eksponentna funkcija kar dobro opiše, saj se premici odlično prilega.

Na podlagi zgornjih opazk sem se odločil, da bo zelo varno, če fittam od $\beta_0 = 5$ naprej.

Poglejmo si še, kakšne velikosti matrike M moramo imeti, da dobimo neko želeno natančnost.

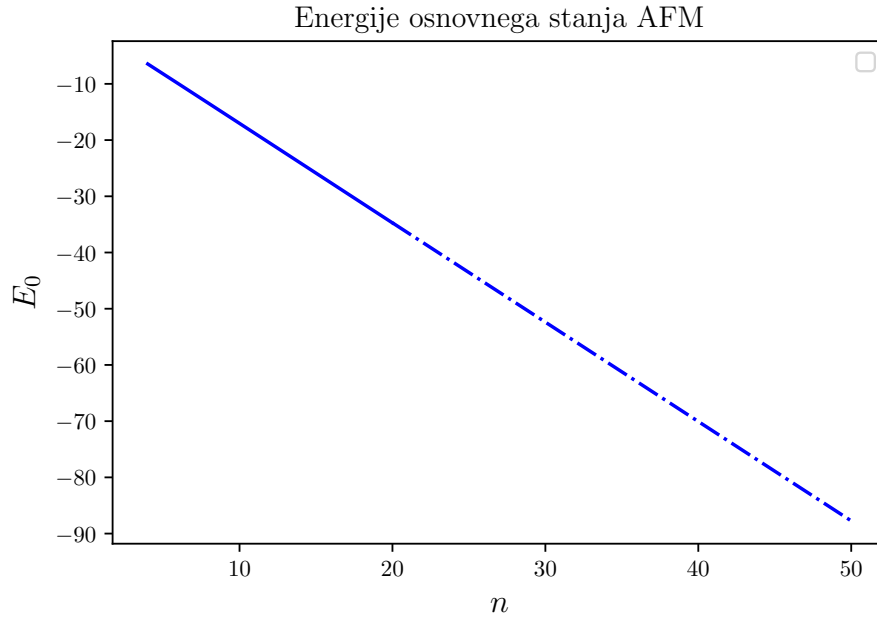


Na zgornjem grafu je prikazana odvisnost potrebnega M za napako norme valovne funkcije manj kot 0.1, kjer sem napako norme valovne funkcije definiriral kot razliko norme valovne funkcije po času $t = 100$ in $t = 0$. Odvisnost žal ne izgleda nič lepo (Upal sem, da bi iz tega lahko kaj ekstrapoliral). Vidi se pa, da seveda z n -jem narašča, kar smo itak že vedeli.



Na sliki je odvisnost energije osnovnega stanja od n , pridobljena z algoritmom *TBED* in z direktno diagonalizacijo. Opazimo, da je napaka (označeno z zeleno črto, pripadajoča y os pa je na desni) zanemarljiva.

Za konec povzamimo glavni rezultat prvega dela naloge v spodnjem grafu:

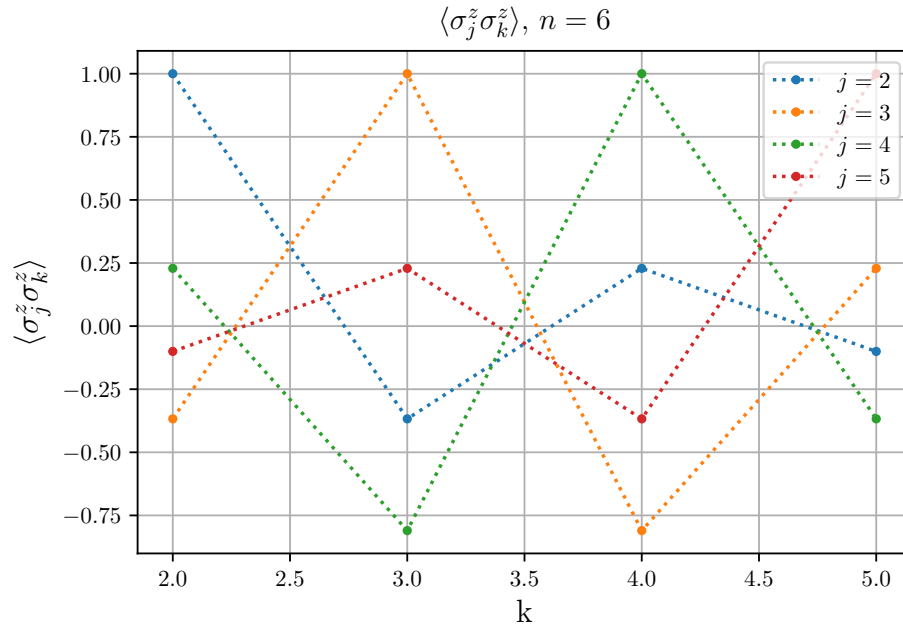


Na sliki je prikazana odvisnost energije osnovnega stanja antiferomagnetne verige v odvisnosti njene dolžine n . Energije do $n = 20$ sem uporabil za ekstrapolacijo do $n = 50$.

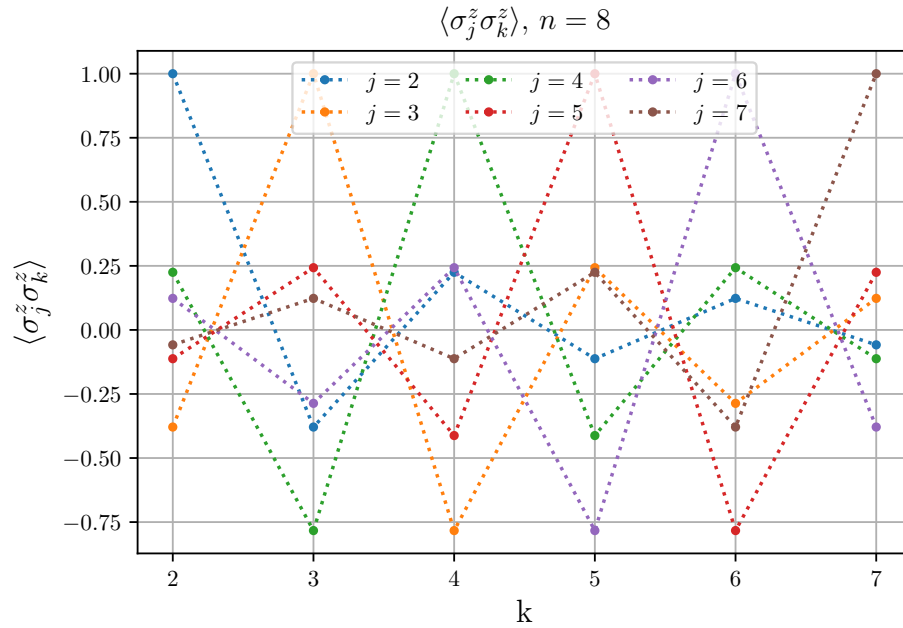
V nadaljevanju si pogledjmo obnašanje spin-spin korelacijske funkcije osnovnega stanja $\langle \sigma_j^z \sigma_k^z \rangle$.

Spinske korelacije

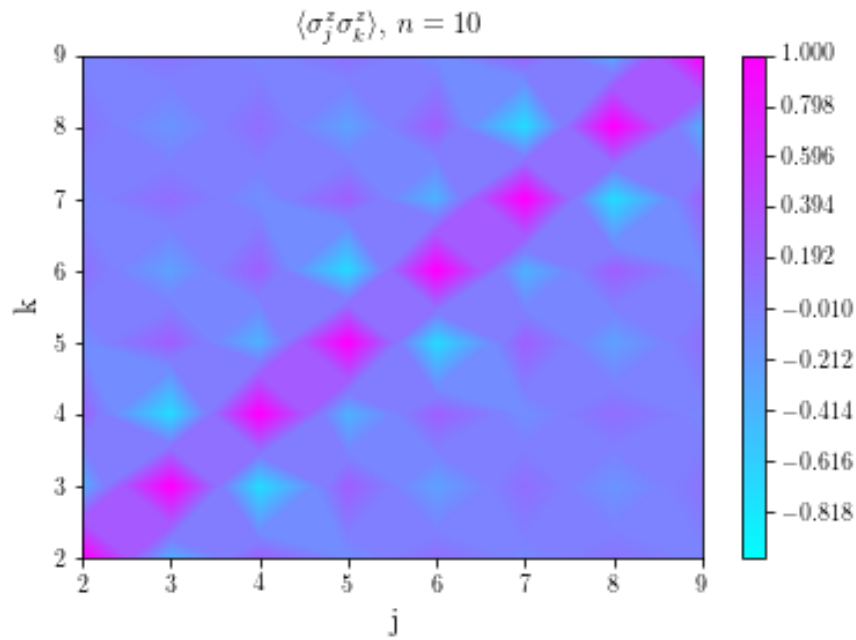
Pričakovane vrednosti dvodelčnih operatorjev znamo izračunati s pomočjo matrik L , R in $T^{(j)}$ osnovno stanje pa tudi že imamo zato si kar takoj pogledjmo korelacijsko funkcijo:



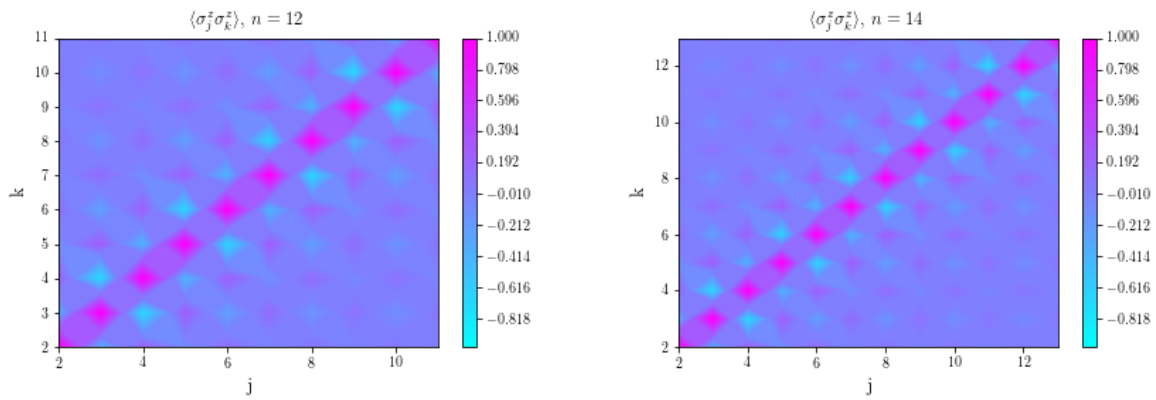
Na sliki je prikazana korelacijska funkcija z komponent spina. Opazimo, da korelacijska funkcija niha med pozitivnimi in negativnimi vrednostmi kar je smiselno, saj se nam zdi, da so v osnovnem stanju AFM sosednji spini nasprotno obrnjeni (kar je res vsaj v Isingovem modelu). Opazimo, da so grafi nastopajo v parih, ki so zrcalna slika drug drugih preko "središča verige", točke " $k = 3.5$ ". To je smiselno, saj ni važno v katero smer obrnemo verigo. Prav tako lahko opazimo, da seveda velja $\langle \sigma_j^z \sigma_k^z \rangle = \langle \sigma_k^z \sigma_j^z \rangle$, saj ti operatorji komutirajo.



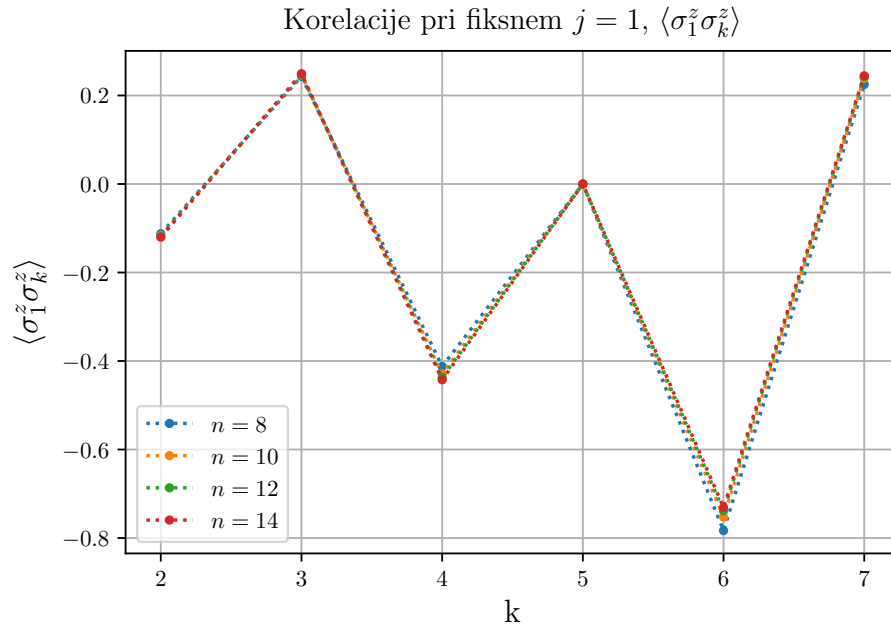
Še primer za malce večjo verigo - $n = 8$. Opazke so podobne kot v primeru $n = 6$



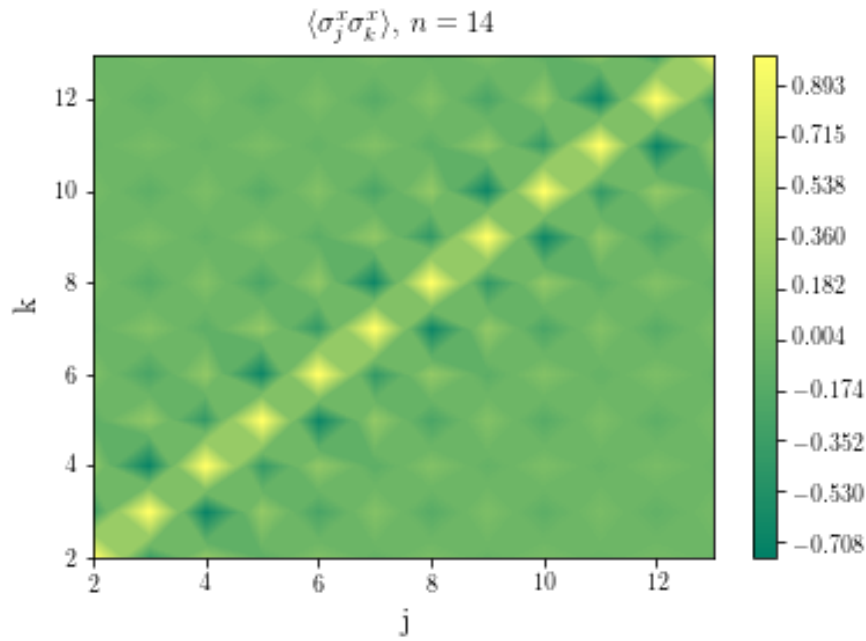
Še slika za $n = 10$, ki pa je tokrat zaradi večje preglednosti narisana kot contour plot.



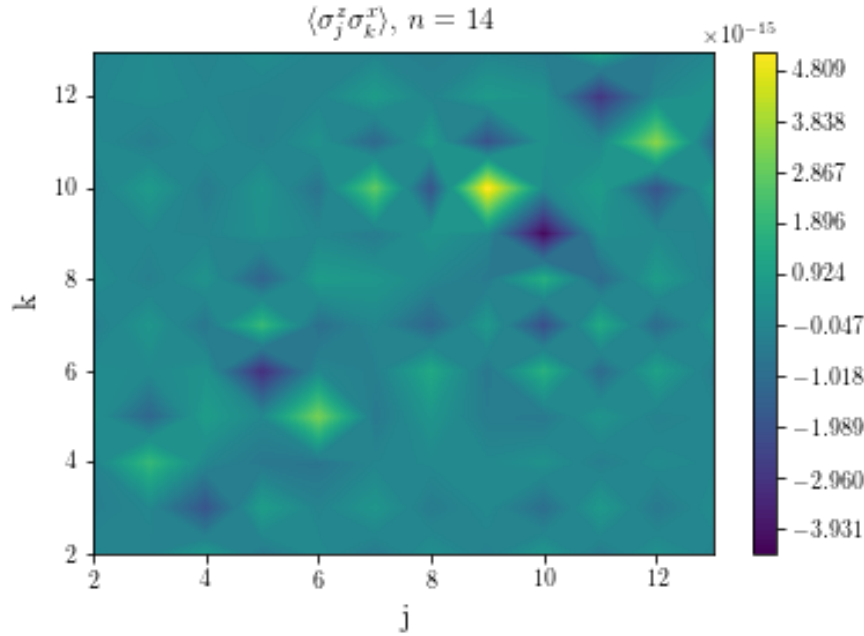
Na slikah sta prikazana še primera $n = 12$ in $n = 14$. Slike z večanjem n ostajajo po izgledu iste, le število točk se povečuje



Na sliki je prikazana korelacija pri fiksnem j za več velikosti sistema. Oblika je ista ne glede na velikost sistema, saj sklapljamo le najbližje sosedo.



Na sliki je prikazana korelacija x komponent spina za $n = 14$. Situacija je ista kot pri korelaciji z spinov, saj izotropni Heisenbergov model nima preferenčne smeri.



Na sliki je prikazana korelacija med z in x komponentami spina. Vidimo, da je ta praktično nič (na prvi pogled zglada, da imamo nekakšne "packe" a so te posledica numerike - glej barvno skalo).

Feromagnetna domenska stena

Naša naslednja naloga je slediti razvoju začetnega stanja, kjer leva polovica spinov kaže v eno smer (v mojem primeru bo to v smeri osi z), desna pa v drugo smer (torej smer osi $-z$). Poleg tega imamo zdaj namesto z antiferomagnetnim opravka z feromagnetnim hamiltonjanom, torej

$$H_{AFM} \rightarrow H_{FM}$$

$$\sum_j \vec{\sigma}_j \cdot \vec{\sigma}_{j+1} \rightarrow - \sum_j \vec{\sigma}_j \cdot \vec{\sigma}_{j+1}$$

Naš dvodelčni operator $U(z)$ se torej spremeni na naslednji način:

$$U_{j,j+1}(z) = \exp(z \vec{\sigma}_j \cdot \vec{\sigma}_{j+1}) \rightarrow \exp(-z \vec{\sigma}_j \cdot \vec{\sigma}_{j+1}) = U_{j,j+1}(-z),$$

torej lahko uporabimo isto kodo kot smo jo imeli v prejšnjih primerih, le da $U(z)$ zamenjamo z $U(-z)$. Tokrat nameravam gledati tudi daljše verige zato bom zaradi spominskih omejitev delal s produktnim stanjem:

$$|\psi\rangle = \bigotimes_{j=1}^N (\cos(\theta_j/2)|0\rangle + \sin(\theta_j/2)e^{i\varphi_j}|1\rangle)$$

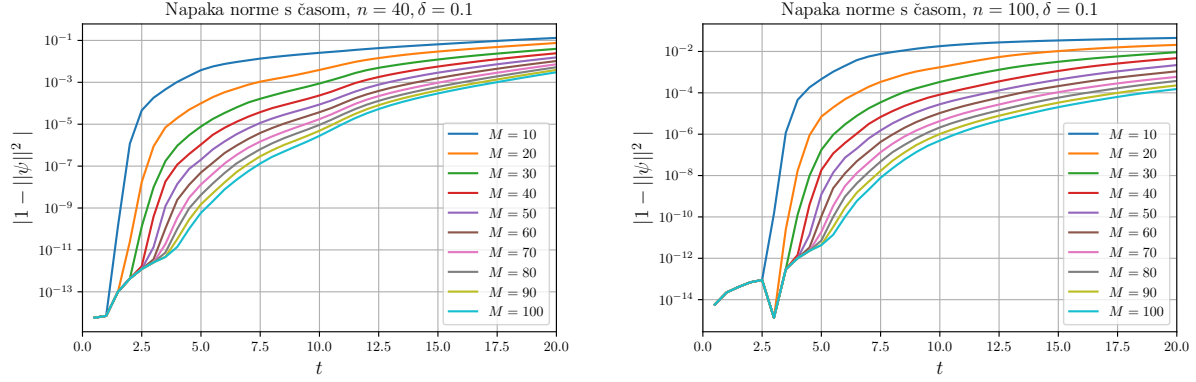
Za zgornje stanje sledi preprosto

$$A_0^{(j)} = \cos(\theta_j/2), A_1^{(j)} = \sin(\theta_j/2)e^{i\varphi_j}, \lambda^{(j)} = 1$$

V primeru domenske stene bom torej začel s stanjem, ki ustreza:

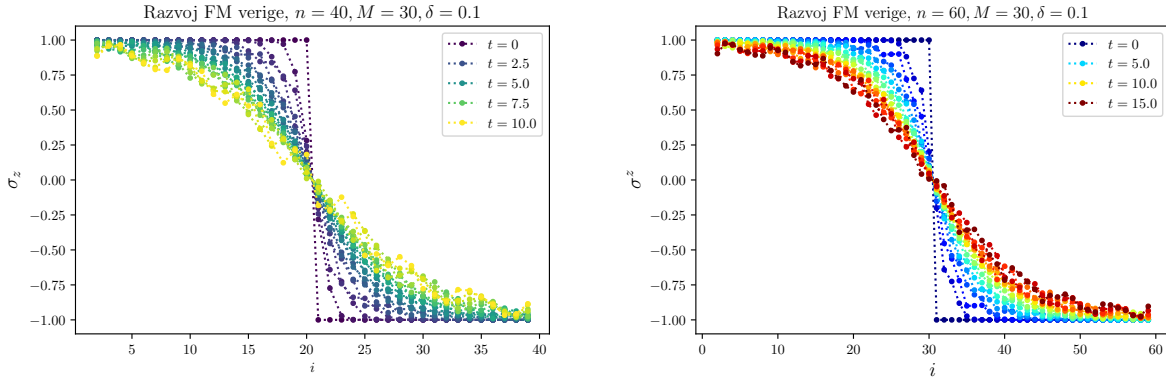
$$\begin{aligned} \theta_j &= 0, j = 1, \dots, N/2 \\ \theta_j &= \pi, j = N/2 + 1, \dots, N \\ \varphi_j &= 0, \forall j \end{aligned}$$

Za začetek si pogledjmo kakšen M moramo izbrati. Prejšnji graf potrebnega M za napako $\delta||\psi|| < 0.1$ tu ni preveč uporaben, ker bom imel opravka z veliko daljšimi verigami in krajšimi časi. Na naslednjem grafu je prikazana časovna odvisnost norme valovne funkcije za več vrednosti M .



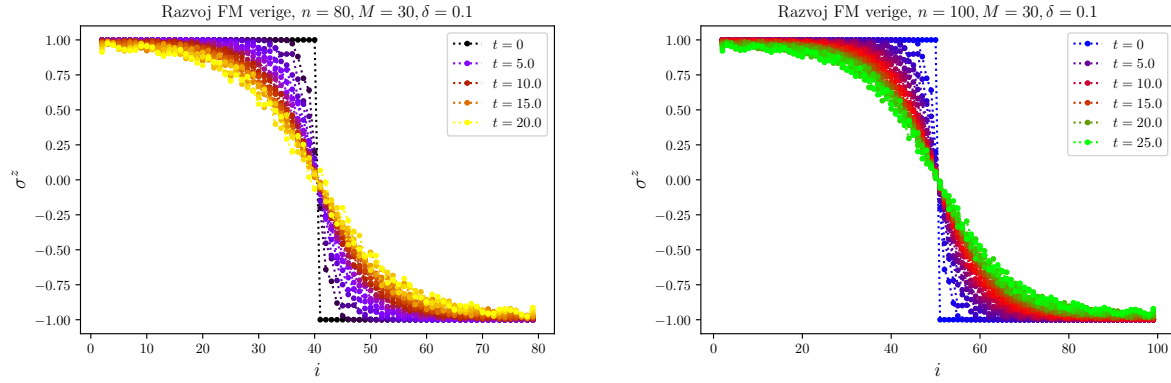
Na zgornjih grafih je prikazana časovna odvisnost napake kvadrata norme val. funkcije (glede na 1) za več maksimalnih velikosti matrik M . Opazimo, da je za obe velikosti verige n za napako pod 10^{-2} po 20 sekundah zadostni $M = 50$. Opazimo tudi, da so napake po 20 sekundah manjše pri večjem n , verjetno ker se pri večjem n efekt domenske stene počasneje širi proti robu. V zgornjem grafu delamo z časovnimi koraki velikosti $\delta = 0.1$

Sedaj si pogledjmo časovni razvoj za več velikosti verig:



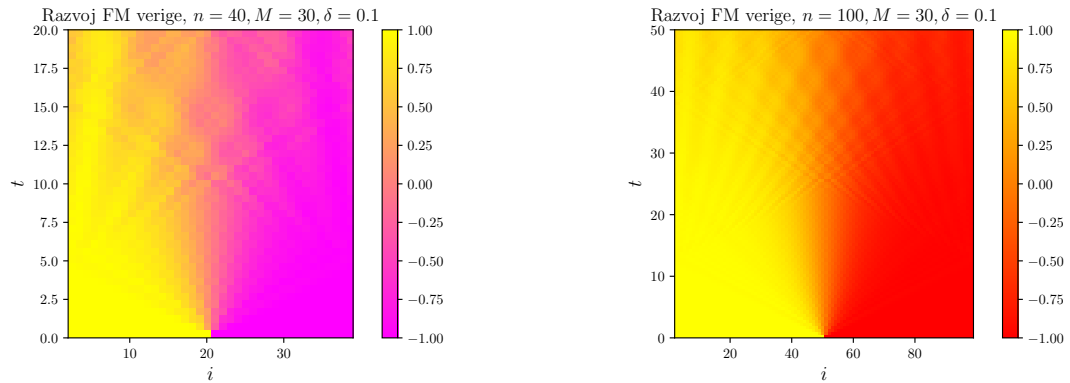
Na zgornjih grafih je veriga narisana pri več časih, kjer je časovni razvoj prikazan z barvo kot kaže legenda. Opazimo, da se na začetku ostra domenska stena s časom "gladi" in širi, kot bi pričakovali. Pri daljši verigi (na desni) je zgodba zelo podobna, le vse skupaj se dogaja na daljši časovni skali. Parametri so kot v naslovu $M = 30, \delta = 0.1$.

Po risanju zgornjih grafov sem se odločil, da bom kar ves čas uporabljal $M = 30$, saj pri večjih M zaidem v težave s spominom. Na zgornjem primeru je za $n = 40$ to gotovo dovolj dobro, saj smo pri napakah videli, da je pri $t = 10$ pridobljena napaka v normi reda 10^{-3} , pri daljših verigah pa morda tudi ni tako slabo, saj na grafu napak za $n = 100$ opazimo, da s časom napaka počasneje raste kot pri manjših verigah za zelo velike čase pa imamo itak že opravka z efekti zaradi roba verige.

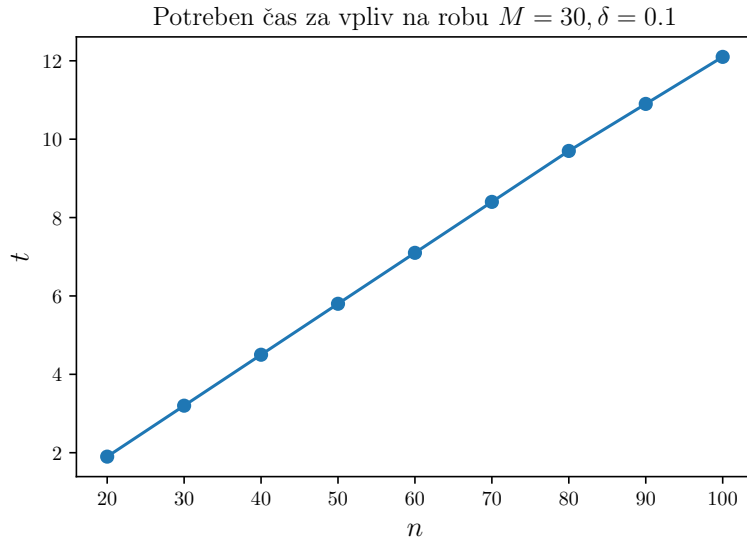


Na zgornjih grafih je časovni razvoj verige prikazan še za daljši verigi. Oblika je spet podobna kot prej le na še daljši časovni skali. Izbranim parametru $M = 30$ lahko tudi pri teh časih in velikosti verige zdaj morda bolj zaupamo, saj smo dobili kar smislen rezultat, poleg tega sem preveril normo funkcije, ki jo dobimo po koncu algoritma in v obeh primerih je ta okoli 0.985, torej napaka ni prevelika. Parametri so kot v naslovu $M = 30, \delta = 0.1$.

Časovni razvoj zgornjih štirih verig je viden tudi na priloženi animaciji razvoj.mp4. Ker domenska stena pri različnih dolžinah verige različno hitro doseže rob verige je na animaciji tudi obnašanje manjših verig dolgo po tem, ko domenska stena doseže rob. Tam opazimo nepravilno obnašanje in veliko napako. Za nekoliko daljše čase sem zaradi boljše nazornosti primere zgornjih verig narisal tudi na spodnjih contournih grafih.

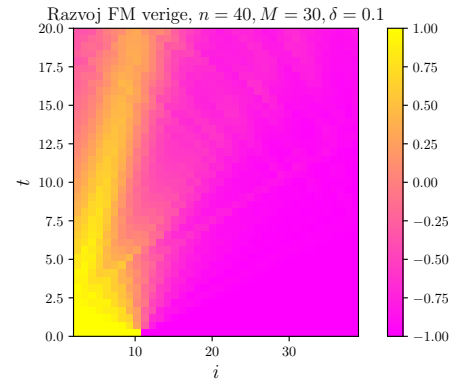
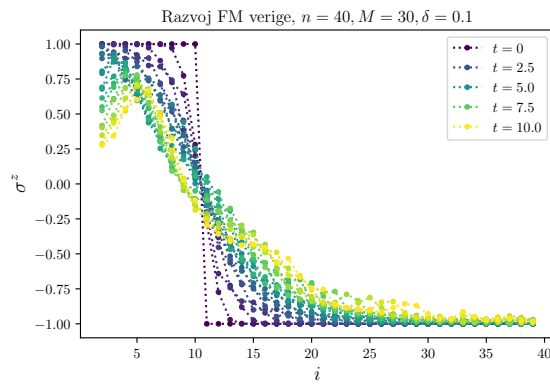


Na zgornjih grafih je prikazan časovni razvoj do dvakrat daljših časov kot prej za verigi $n = 40, n = 100$. Po daljšem času začne lokalna magnetizacija bolj oscilirat (vidimo, da horizontalno barve bolj alternirajo). Napaka norme na koncu je tu že večja - pri primeri $n = 40$ je norma na koncu nekje 0.94 pri $n = 100$ pa 0.86, saj delamo veliko več korakov.

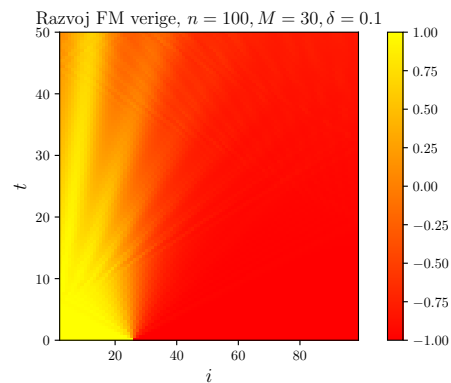
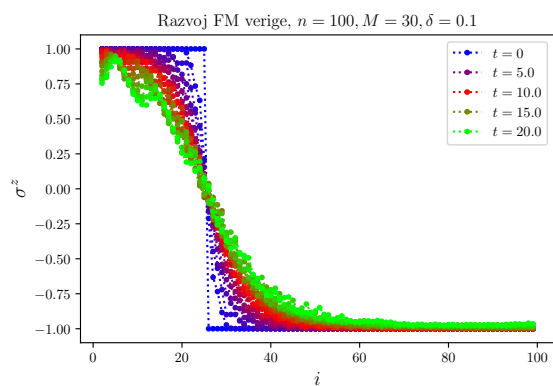


Tukaj sem gledal potreben čas, da se efekt stene čuti na robu v odvisnosti od n . To sem gledal tako, da sem opazoval, kdaj magnetizacija na levem robu pade pod 0.99. Vidimo, da je odvisnost od n linearna z naklonom približno $1/8$.

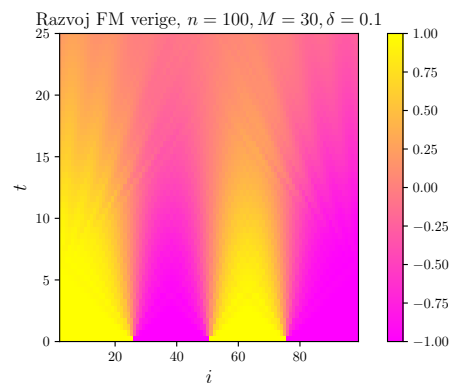
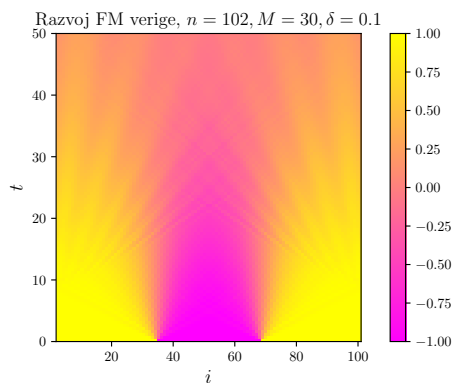
Kot zanimivost sledi še nekaj prikazov razvoja drugačnih začetnih stanj verige:



Na zgornjih grafih je prikazan časovni razvoj verige, kjer je na začetku le četrtnina spinov gor, tri četrtine pa dol. Rezultat je pričakovan: Motnja zaradi domenske stene veliko hitreje zadane levi rob in robni efekti so takoj vidni, magnetizacija na levem robu se močneje niža, medtem ko je desni rob še nedotaknjen.



Na zgornjih grafih je prikazan časovni razvoj verige, kjer je na začetku le četrtnina spinov gor, tri četrtine pa dol še za primer daljše verige. Obnašanje je podobno kot prej, čeprav se magnetizacija tu ne spreminja tako zelo veliko na levem robu kot v prejšnjem primeru.



Na zgornjih grafih je prikazan časovni razvoj verige, kjer imamo na začetku dve domenski steni. Rezultat je spet pričakovan - domenski steni se pač obe začneta širiti.