UNIVERZA V LJUBLJANI
FAKULTETA ZA MATEMATIKO IN FIZIKO
ODDELEK ZA FIZIKO

VIŠJE RAČUNSKE METODE

10. naloga: Tenzorska renormalizacijska grupa

Žiga Šinigoj, 28222025

Ljubljana, avgust 2023

1 Uvod

TRG metoda nam omogoča prevedbo vseh klasičnih modelov na mreži z lokalno interakcijo na tenzorsko mrežo. S kontrakcijo tenzorskega vezja je mogoče izračunati particijsko vsoto

$$Z(\beta) = \sum_{\underline{\sigma}} e^{\beta E[\underline{\sigma}]} \tag{1}$$

, kjer teče vsota po vseh možnih konfiguracijah spinov, ki imajo q prostostnih stopenj. Za izračun je potrebno prevesti particijsko vsoto v obliko popolnoma kontrahirane tenzoske mreže

$$Z = \sum_{ijk...} A_{ijk...} A_{ilm...} A_{jpq...} ... \tag{2}$$

, kjer so $A_{1,...,n}$ tenzorji n-tega ranga. Rang tenzorja običajno ustreza številu najbližjih sosedov. Tenzor A kodira interakcijo med najbližjimi sosedi. Ker je računsko zelo zahtevno izvesti kontrakcijo tenzorskega vezja je ideja metode ta, da s pomočjo SVD razcepa preoblikujemo tenzor A na tak način, da je mogoče tenzorsko mrežo zmanjšati za polovico. To je osnoven korak metode, ki ga ponavljamo dokler ne dobimo trivialne tenzorske mreže z enim samim tenzorjem oz. dokler ne dosežemo fiksne točke

$$RG(A_*) = A_* \ , \tag{3}$$

kjer je $RG()$ korak metode.

1.1 Pottsov model na kvadratni mreži

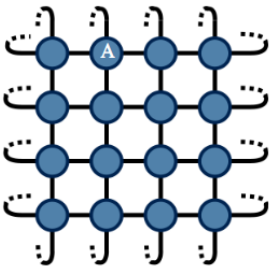
Energija Pottsovega modela je

$$E(\underline{\sigma}) = -J \sum_{i,j=1}^{N} (\delta_{\sigma_{i,j},\sigma_{i,j+1}} + \delta_{\sigma_{i,j},\sigma_{i+1,j}}) \tag{4}$$

in pripadajoči tenzor tenzorskega vezja

$$A_{\sigma_l,\sigma_r,\sigma_t,\sigma_d} = e^{\beta J(\delta_{\sigma_t,\sigma_r}+\delta_{\sigma_t,\sigma_l}+\delta_{\sigma_d,\sigma_r}+\delta_{\sigma_d,\sigma_l})} \ . \tag{5}$$

Ob predpostavki, da imamo periodične robne pogoje, lahko vse lokalne interakcije in posledično particijsko vsoto, zapišemo kot popolno kontrakcijo tenzorske mreže V tenzorski mreži spini vedno ležijo na povezavah



(žicah). Da prevedemo tenzorsko mrežo velikosti $N \times N$ na $N/2 \times N/2$ naredimo transformacijo $RG()$. Tenzor A^j, kjer j označuje korak metode, na dva načina zapišemo kot produkt matrik

$$A^j_{\sigma_l,\sigma_r,\sigma_t,\sigma_d} = (F_1F_3)(\sigma_l,\sigma_t),(\sigma_d,\sigma_r) = (F_2F_4)(\sigma_t,\sigma_r),(\sigma_l,\sigma_d) \ , \tag{6}$$

, kjer je M^j dimenzija σ_i v j-tem koraku, nato naredimo SVD razcep na matrikah

$$(F_1F_3)(\sigma_l,\sigma_t),(\sigma_d,\sigma_r) = \sum_{m=1}^{M} {}^{j+1}U_{(\sigma_l,\sigma_t),m}\sqrt{s_m}\sqrt{s_m}V_{(\sigma_d,\sigma_r),m} = \sum_{m=1}^{M} {}^{j+1}F_{1(\sigma_l,\sigma_t),m}F_{3m,(\sigma_d,\sigma_r)} \tag{7}$$

$$(F_2F_4)(\sigma_t,\sigma_r),(\sigma_l,\sigma_d) = \sum_{m=1}^{M} {}^{j+1}U_{(\sigma_t,\sigma_r),m}\sqrt{s_m}\sqrt{s_m}V_{(\sigma_l,\sigma_d),m} = \sum_{m=1}^{M} {}^{j+1}F_{2(\sigma_t,\sigma_r),m}F_{4m,(\sigma_l,\sigma_d)} \ . \tag{8}$$

M^{j+1} je maksimalna dimenzija, določena z številom neničelnih singularnih vrednosti, ki jih ohranimo. Sedaj kontrahiramo matrike F_1, F_2, F_3, F_4 na način, da razpolovimo tenzorsko vezje

$$A^{j+1}_{\sigma'_l,\sigma'_r,\sigma'_t,\sigma'_d} = \sum_{x,y,z,w=1}^{M^j} F_{3\sigma'_l,w,x} F_{1z,y,\sigma'_r} F_{4\sigma'_t,x,y} F_{2\sigma'_d,w,z} \ . \tag{9}$$

Novo tenzorsko vezje je sedaj sestavljeno iz A^{j+1} tenzorjev in je pol manjše. Postopek ponavljamo do trivialne mreže z enim tenzorjem iz katerega dobimo particijsko vsoto kot

$$Z = \sum_{\sigma,\sigma'=1}^{M_N} A^N_{\sigma,\sigma,\sigma',\sigma'} \ . \tag{10}$$

Povprečno energijo sistema dobimo z numeričnim odvodom particijske vsote.

1.2 Pottsov model na heksagonalni mreži

Particijsko funkcijo sedaj zapišemo kot

$$Z = \sum_{\underline{\sigma}} e^{-\frac{\beta}{2}E[\underline{\sigma}]} = \sum_{\underline{\sigma}} \prod_{\langle i,j\rangle} e^{\frac{\beta}{2}\delta_{\sigma_i,\sigma_j}} = \sum_{\underline{\sigma}} \prod_{\langle i,j\rangle} \Theta_{\sigma_i,\sigma_j} \ , \tag{11}$$

kjer je Θ tenzor, ki opisuje interakcijo med dvema sosednjima spinoma, i teče po podmreži A in j po podmreži B. Na heksagonalni mreži ima vsak spin 3 najbližje sosede, torej potrebujemo tenzorsko vezje sestavljeno iz tenzorjev tretjega ranga. Heksagonalna mreža je tudi bipartitna in bo v splošnem potrebno ločevati med tenzorjem na podmreži A in B. Da dobimo tenzor 3 ranga naredimo najprej SVD razcep na $\Theta_{\sigma_i,\sigma_j}$

$$\Theta_{\sigma_i,\sigma_j} = \sum_{k=1}^{q} U_{\sigma_i,k}\sqrt{s_k}\ \sqrt{s_k}V_{\sigma_j,k} = Q^a_{\sigma_i,k}Q^b_{\sigma_j,k} \ , \tag{12}$$

kjer q teče po številu prostostnih stopenj spina. Sedaj tvorimo tenzorja T^A in T^B kot

$$T^A_{xyz} = \sum_{\sigma_i} Q^a_{\sigma_i,x}Q^a_{\sigma_i,y}, Q^a_{\sigma_i,z} \tag{13}$$

$$T^B_{xyz} = \sum_{\sigma_j} Q^a_{\sigma_j,x}Q^a_{\sigma_j,y}, Q^a_{\sigma_j,z} \ . \tag{14}$$

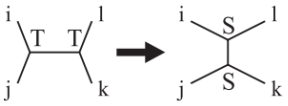
Garfično tenzor T predstavlja vozlišče in tri polovice žic v tenzorskem vezju. Osnovni korak $RG(T^j) \to T^{j+1}$, ki zmanjša vezje za polovico, naredimo po naslednjem postopku. Najprej združimo tenzorja T^A in T^B

$$A^j_{ijkl} = \sum_m T^A_{ijm}T^B_{klm} \ , \tag{15}$$

in naredimo SVD razcep, kjer najprej tenzor 4 ranga A_{ijkl} predstavimo kot matriko $M_{(l,i),(j,k)}$

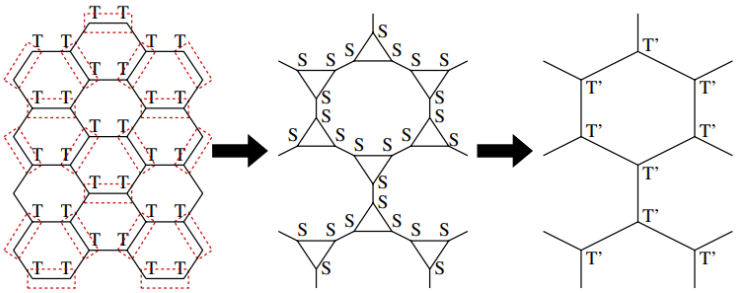
$$M_{(l,i),(j,k)} = \sum_n^{M^j} U_{(l,i),n}\sqrt{s_n}\ \sqrt{s_n}V_{(j,k),n} = S^A_{(l,i),n}S^B_{(j,k),n} \ . \tag{16}$$

Na ta način smo naredili prevezavo vezja , ki je ključna za prepolovitev mreže. Sedaj lahko kontrahiramo



tri tenzorje S (slika 1)

$$T^{A,j+1}_{ijk} = \sum_{p,q,r} S^{A,j}_{irp}S^{A,j}_{jqr}S^{A,j}_{kpq} \ , \tag{17}$$



Slika 1: TRG korak na heksagonalni mreži.

in podobno za $T^{B,j+1}_{ijk}$. Na ta način dobimo nova tenzorja ki sestavljata manjšo tenzorsko mrežo. Na vsakem koraku se površina osnovne celice poveča za faktor 3. Dani RG postopek ponavljamo do fiksne točke. Osnovna celica je tenzor šestega ranga (pravilni šestkotnik) sestavljen iz treh tenzorjev A_{ijkl}

$$C_{ijklmn} = \sum_{a,c,e} A_{akel} A_{ciaj} A_{emcn} \ , \tag{18}$$

ki tudi predstavlja trivialno tenzorsko mrežo po N korakih metode. Particijsko vsoto dobimo s kontrakcijo tenzorja C

$$Z = \sum_{ijk}^{M_N} C_{ijkijk} \tag{19}$$

Povprečno energijo sistema dobimo z numeričnim odvodom particijske vsote.

2 Rezultati

Računski del naloge sem opravil v C++ in Pythonu, grafe v Pythonu. Najprej sem napisal vse kontrakcije 'na roke' torej s for (oznake na grafih c++,for ali Python,for) zankami in potem uporabil metode za kontakcijo tenzorjev v paketu Eigen in v Pytonu Tensorflow. Pričakoval bi, da bo Tenzorflow hitrejši, saj je napisan v C++. Ima pa to prednost, da je mogoče kontrahirati več tenorjev naenkrat in omogoča uporabo Einsteinovega sumacijskega pravila. Pri paketu Eigen pa je mogoče kontrahirati samo dva tenzorja hkrati. Časovno odvisnost sem pogledal na Pottsovem modelu na kvadratni mreži, prav tako odvisnost od števila $RG()$ korakov in maksimalne dimenzije SVD razcepa v j-tem koraku $M^j_{max} = M_{max} = M$. Izkaže se, da se vrednosti tenzorjev hitro večajo s številom korakov in pride do divergence, zato sem tenzorje normiral. Po normiranju je mogoče izračunati particijsko vsoto večjih sistemov $\approx 2^{40}$ spinov in več. Edina težava, na katero sem naletel je nestabilnost metode pri določenih parametrih. Pri izračunu energijskega profila sistema v odvisnosti od temperature pride pri določenih temperaturah oz. β do divergence, pri določenih vrednostih N_{iter} in M. Ob spremembi N_{iter} in M se lahko pojavijo divergence pri drugih temperaturah. Načeloma se pojavijo pri večjem številu iteracij $N_{iter} > 16$, vendar je pomemben tudi parameter M. Tukaj mislim predvsem na nekaj točk (2,3) v profilu 160 točk. V takem primeru sem jih zavrgel in potem računal odvod med sosednjima, zato je mogoče v profilu na kakšnem mestu krivulja nezvezno odvedljiva. Divergenca nima nekega trenda, npr. da bi se pojavljala pri manjših ali večjih β, niti ni nujno v temperaturah, ki bi lahko bile problematične v binarnem zapisu. Najverjetneje je težava v SVD razcepu, ki uporablja algoritem deli in vladaj in ni tako natančen, je pa hitrejši od Jacobijevega SVD razcepa. V Pythonu nisem imel teh težav, čeprav je metoda SVD razcepa enaka kot v knjižnici Eigen, vendar so bile tudi vrednosti N_{iter} in M manjše.

2.1 Pottsov model na kvadratni mreži

Pri računanju profila sem uporabil za sklopitveno konstanto vrednost J=0.5, saj sem tako lahko primerjal rezultate iz 6. naloge. Če želimo simulirati večje sisteme je potrebno normirati tenzorje A^j.

$$A^j \to \frac{A^j}{||A^j||} \xrightarrow{RGstep} \frac{A^{j+1}}{||A^j||^2} \ , \tag{20}$$

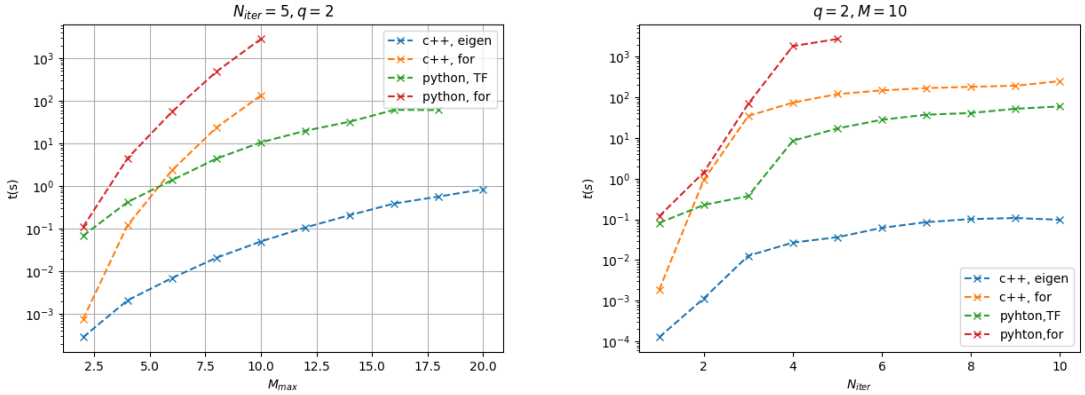
kvadrat norme se pojavi zaradi tega, ker potrebujemo 2 tenzorja A^j, da tvorimo nov tenzor. V N-tem koraku dobimo

$$\frac{A^N}{\prod_{j=0}^{N-1} ||A^j||^{2^{(N-j)}}} \tag{21}$$

Ker potrebujemo logaritem particijske funkcije se izraz prepiše v

$$ln(Z) = ln(A^N_{\sigma,\sigma,\sigma',\sigma'}) - \sum_{j=0}^{N-1} 2^{N-j} ln||A^j|| \ . \tag{22}$$

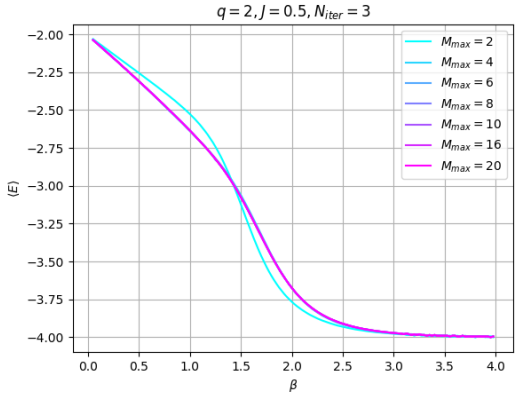
Najprej sem pogledal časovno zahtevnost izračuna logaritma particijske vsote v odvisnosti od maksimalne 'bond' dimenzije $M_{max} = M$ in števila RG() korakov N_{iter} (v to število ni vključena kontrakcija zadnjega tenzorja) (slika 2). Časovna zahtevnost je na začetku manjša, saj maksimalna bond dimenzija ni nujno dosežena. Eigen knjižnjica je vsaj dva velikostna reda hitrejša od Tensorflow knjižnjice. Pomembno je povdariti da sem primerjal obe implementaciji na CPU, saj ima Tensorflow možnost tudi računanja z grafično kartico, v tem primeru bi lahko bili rezultati drugačni. Naivna implementacija z for zankami pa je časovno zelo potratna.



(a) (b)

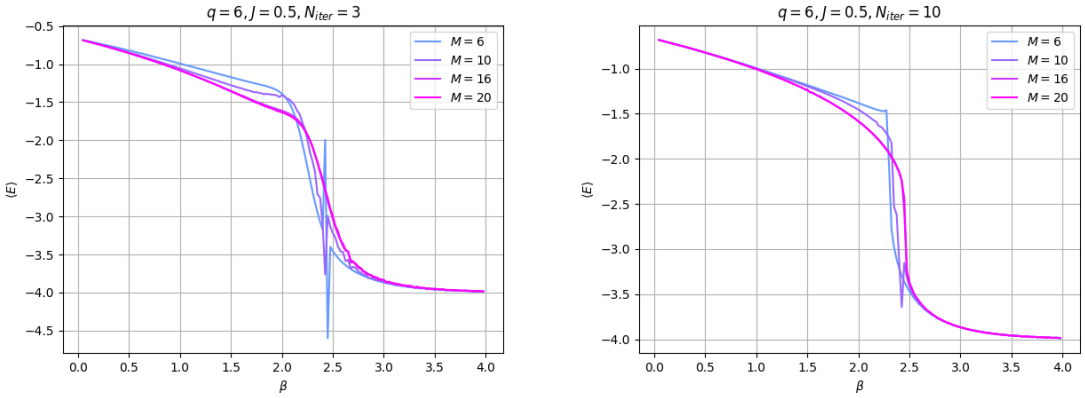
Slika 2: a) Časovna zahtevnost izračuna ln(Z) v odvisnosti od velikosti 'bond' dimenzije. b) Časovna zahtevnost izračuna ln(Z) v odvisnosti od števila korakov metode N_{iter}.

Odvisnost energijskega profila od velikosti 'bond' dimenzije prikazujejo slike 3 in 4. Pri $q = 2$ hitro TRG hitro konvergira, saj sta samo 2 prostostni stopnji. Že po treh iteracijah in $M = 4$ dobimo pravilno rešitev. Pri večjih q (q=6) pa je več prostostnih stopenj in potrebujemo večjo 'bond' dimenzijo, da dobimo pravilen



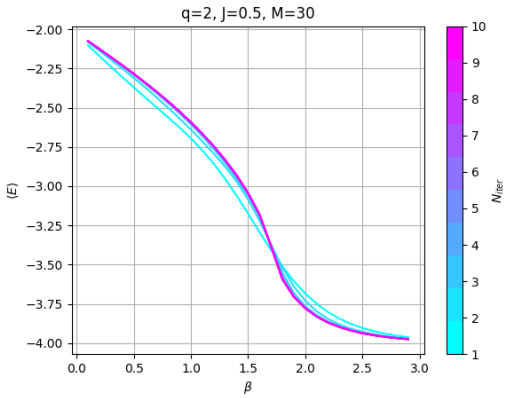
Slika 3: Povprečna energija sistema v odvisnosti od β pri različnih vrednostih M, $q = 2$.

rezultat. Če preveč porežemo lastne vrednosti se izgubi preveč informacije o sistemu in profil je napačen. Pomembno pa je tudi število iteracij oz. velikost sistema. Pri treh iteracijah ne pridemo do pravega energijskega profila, kljub velikem M ($M = 20$, slika 4).



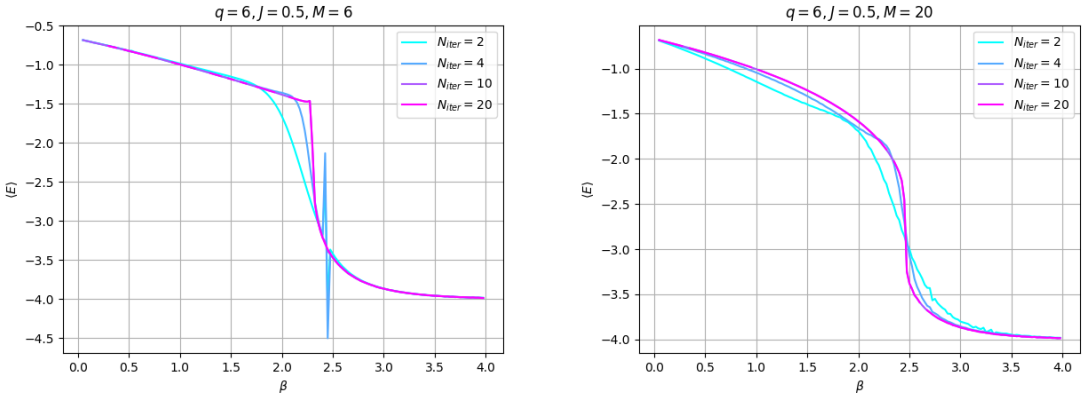
Slika 4: Povprečna energija sistema v odvisnosti od β pri različnih vrednostih M, $q = 6$.

Odvisnost od števila iteracij prikazujejo sliki 5 in 6. Podobno kot prej pri $q = 2$ sistem hitro konvergira. Pri večjih q ($q = 6$) pa je potrebno narediti več iteracij pri zadostni dimenziji M, da dobimo pravilen profil (slika 6). Da dobimo pravilen profil je potrebno imeti dovolj veliko bond dimenzijo in dovolj velik sistem. Zanimivo je, da lahko dobimo energijo osnovnega stanja pri nizkih temperaturah že z zelo majhnimi 'bond' dimenzijami in majhnim številom korakov.



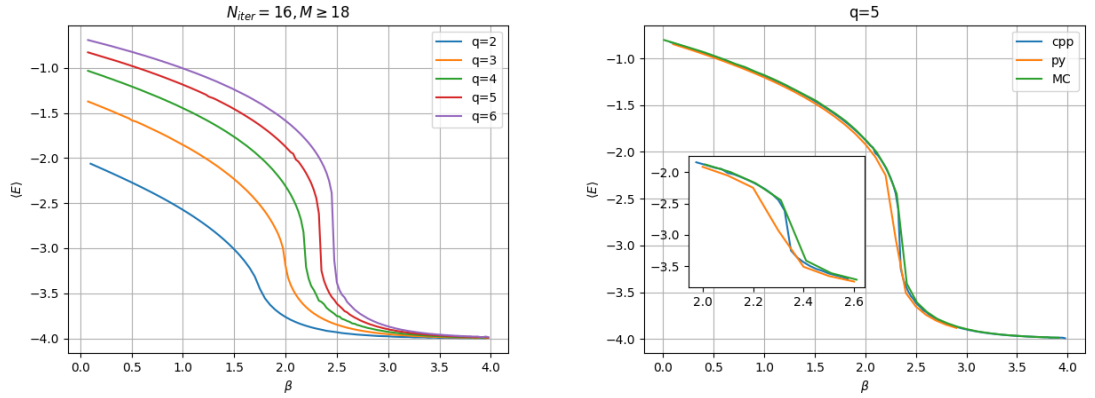
(a)

Slika 5: Povprečna energija sistema v odvisnosti od β pri različnem številu korakov N_{iter}, $q = 2$.



Slika 6: Povprečna energija sistema v odvisnosti od β pri različnem številu korakov N_{iter}, $q = 6$.

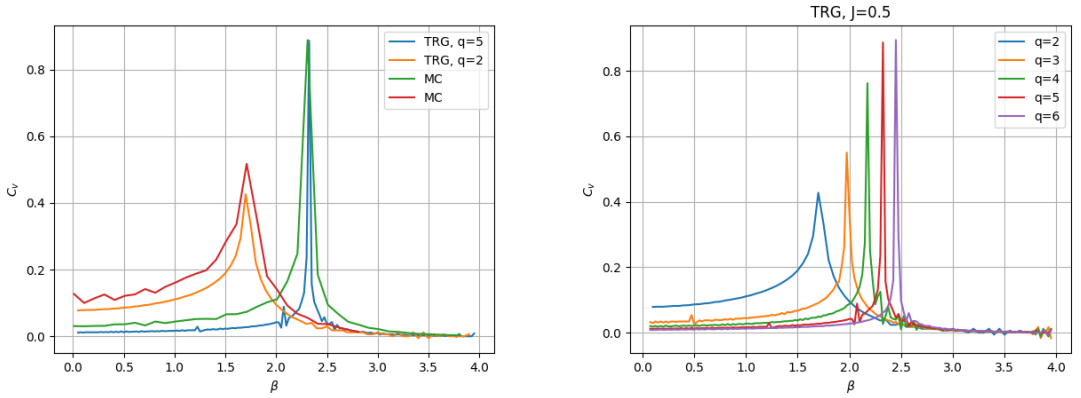
Pričakovano vrednost energije sistema pri različnem številu q prikazuje slika 7. Iz slike 7b lahko opazim, da je profil skoraj nezvezen (modra krivulja). Primerjal sem tudi s profilom izračunanim z Monte Carlo metodo (MC), ki pa da nekoliko slabše rezultate. Mogoče zaradi resolucije v x-osi, enako velja za izračun v Pythonu, saj so temperaturne točke bolj narazen zaradi večje časovne zahtevnosti. Če bi vzel še bolj gost interval temperatur, bi mogoče res opazil nezveznost pri $q = 5$.



(a) (b)

Slika 7: a) Pričakovana vrednost energije sistema v odvisnosti od β pri različnem q. b) Pričakovana vrednost energije sistema v odvisnosti od β pri $q = 5$. MC označuje metodo Monte Carlo, graf je kopiran iz 6. naloge.

Specifično toploto izračunamo z numeričnim odvodom energijskega profila. TRG metoda s dobro ujema z rezultati iz MC simulacije (slika 8a). Odvisnost od q prikazuje slika 8b.



(a) (b)

Slika 8: a) Specifična toplota v odvisnosti od β, rezultati iz MC simulacije so iz 6. naloge. b) Specifična toplota v odvisnosti od β pri različnih q.

2.2 Pottsov model na heksagonalni mreži

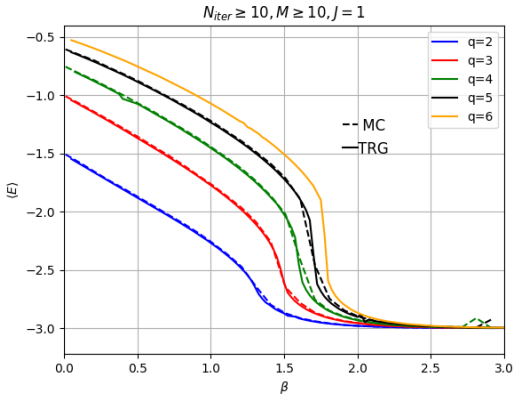
Postopek opisa $RG()$ koraka sem opisal v uvodu. Ker sem imel dodatno nalogo pri Modelski analizi ravno Pottsov model na heksagonalni mreži z MC metodo, sem lahko rezultate primerjal. Vse odvisnosti od parametrov (N_{iter}, M) sem že pogledal pri prejšnjem modelu, zato bom prikazal samo relevantne odvisnosti. Edina pomembna stvar, ki še ni pojasnjena je normalizacija tenzorjev. Ta je nujno potrebna, če želimo kaj izračunati, saj drugače ni mogoče narediti več kot $N_{iter} = 3$. Osnovna celica je tenzor šestega ranga (pravilni šestkotnik), kateremu se poveča površina za faktor 3 pri vsakem koraku. Pri $RG()$ koraku računamo tenzorje T^A in T^B. Normalizacija tenzorja C_{ijklmn} je računsko zahtevna in neuporabna, saj ga ni mogoče razbiti na 3 tenzorje A. Vseeno je potrebno normalizirati glede na tenzor C, saj je končna kontrakcija ravno po tenzorju C. Naredil sem približek

$$\frac{C_{ijklmn}}{||C_{ijklmn}||} \approx \frac{1}{||A_{ijkl}||^3} \sum_{a,c,e} A_{akel} A_{ciaj} A_{emcn} = \frac{C_{ijklmn}}{||A_{ijkl}||^3} \ . \tag{23}$$

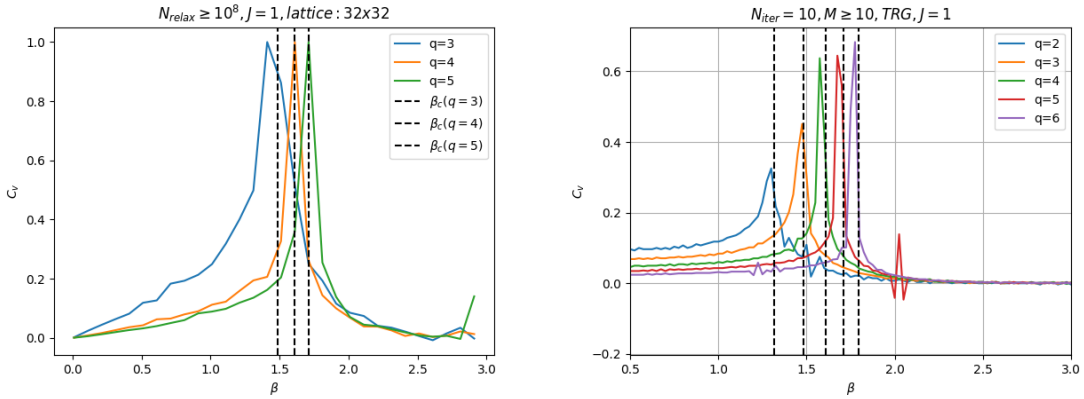
Logaritem particijske funkcije sedaj izračunamo kot

$$ln(Z) = ln(C^N_{ijkijk}) - \sum_{j=0}^{N-1} 3^{N-j} ln||A^j|| \ . \tag{24}$$

Povprečno energijo v odvisnosti od parametra q, izračunano z TRG in MC metodo prikazuje slika 9. Specifično toploto izračunano z MC metodo prikazuje slika 10a, izračunano s TRG metodo pa slika 10b. TRG metoda malo odstopa od analitičnih vrednosti pri $q = 4, 5, 6$. Razlog bi lahko bil približek pri normiranju tenzorjev.



Slika 9: Povprečna energija v odvisnosti od β.



(a) (b)

Slika 10: a) Specifična toplota izračunana z MC metodo, črne črtkane črte prikazujejo kritično temperaturo izračunano po [1]. b) Specifična toplota izračunana s TRG metodo, črne črtkane črte prikazujejo kritično temperaturo izračunano po [1].

3 Zaključek

V poročilu smo analizirali uporabo TRG metode na Pottsovem modelu na kvadratni in heksagonalni mreži. Z izračunom energije in specifične toplote kot funkcije temperature smo pridobili vpogled v toplotne lastnosti sistema. TRG metoda se je izkazala za učinkovito pri preučevanjeu faznih prehodov. Pomembna izboljšava metode je normalizacija tenzorjev, ki nam omogoča študije večjih sistemov.

Literatura

[1] F. Y. Wu, *The Potts model*, Rev. Mod. Phys. **54**, 235 (1982).