

4. naloga – Kubitne verige

Jože Zobec

1 Uvod

Oznake in definicije bodo iste, kot so v skripti.

V tej nalogi imamo propagacijo po času, kot tudi po temperaturi (imaginarnem času).

Ker bomo imeli opravka z veliko množenji, matrika pa je le dimenzije 4×4 in je simetrična tridiagonalna itd. se splača to transformacijo zakodirati naravnost v kodo. Če moramo transformirati vektor $\underline{x} = (x_0, x_1, x_2, x_3)^T$, se ta po propagaciji s kompleksnim koeficientom α za kompleksni korak z transformira tako:

$$\begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \xrightarrow{U^{(2)}(\alpha z)} e^{-\alpha z} \begin{bmatrix} \exp(2\alpha z)x_0 \\ \cosh(2\alpha z)x_1 + \sinh(2\alpha z)x_2 \\ \sinh(2\alpha z)x_1 + \cosh(2\alpha z)x_2 \\ \exp(2\alpha z)x_3 \end{bmatrix}. \quad (1)$$

Temperaturni propagator je $\sim \exp(-\beta H)$, in časovni je $\sim \exp(-itH)$, kjer sta β in t realni količini¹. Odtod vidimo, da je z čisto imaginarno število za čas in čisto realno število za temperaturni premik. Oz. drugače povedano

$$|\psi(t)\rangle = e^{-itH}, \quad |\psi(-i\beta)\rangle = e^{-\beta H}.$$

2 Optimizacija propagatorja

Naš propagator smo razbili na sode in na lihe. To pomeni da najprej po enačbi (1) delujemo na stanja².

$$\begin{aligned} &|s_N s_{N-1} \dots s_3 \overline{0} \overline{0}\rangle, \\ &|s_N s_{N-1} \dots s_3 \overline{0} 1\rangle, \\ &|s_N s_{N-1} \dots s_3 1 \overline{0}\rangle, \\ &|s_N s_{N-1} \dots s_3 1 1\rangle. \end{aligned}$$

Če je takih četvorčkov več, vedno delujemo na vsakega le enkrat. Potem se moramo premakniti na drug tak blok četvorčkov, s_4, s_3 :

¹ t – brezdimenzijski čas, β – recipročna brezdimenzijska temperatura

²delamo v bazi σ_j^z , 0 pomeni spin „dol“, 1 pa pomeni spin „gor“

$$\begin{aligned}
&|s_N s_{N-1} \dots \overline{0\ 0} s_2 s_1\rangle, \\
&|s_N s_{N-1} \dots 0\ 1\ s_2 s_1\rangle, \\
&|s_N s_{N-1} \dots 1\ 0\ s_2 s_1\rangle, \\
&|s_N s_{N-1} \dots 1\ 1\ s_2 s_1\rangle.
\end{aligned}$$

Iskanje ustreznih četvorčkov za biti časovno zahteven posel in nam lahko oteži že tako časovno zahteven izračun, saj ne vemo, kako so tile četvorčki urejeni v decimalnem zapisu. Izkaže se, da lahko iz prvega bloka, četvorkčkov, ki so VEDNO urejeni že v decimalnem (tj. lahko zapišemo njihove indekse kar kot $\{[1, 2, 3, 4], [5, 6, 7, 8], \dots\}$), pridemo v poljuben blok z ustreznim zamikom bitov (seveda z ustreznimi periodičnimi pogoji, za katere poskrbi modulo po največjem možnem indeksu za dano dolžino verige). Če želimo iz prvega sodega bloka priti na n -ti sodi blok, potem bite zamaknemo za $2n$ v levo. Če želimo iz sodega priti v n -ti lihi blok, potem bite zamaknemo za $2n - 1$ v levo. Modulo že sam po sebi poskrbi za naše periodične robne pogoje.

Zamikanje bitov je zelo hitra operacija, tako to delovanja našega programa močno zmanjša, prav tako pa nam ni treba računati lihih in sodih posebej.

Zamik bitov z n v levo števila x je ekvivalentno $x \leftarrow x \cdot 2^n \pmod{x_{\max}}$, vendar je čisti zamik hitrejši, tako da priporočam tega, če ga programsko okolje dopušča, saj je ta že vgrajen v samo elektroniko naših procesorjev.

3 Optimizacija hamiltoniana

Stanja bom od sedaj številčil tako kot v programu, tj. namesto od $1 \dots N$, raje od $0 \dots N - 1$.

Propagator smo dobili tako, da smo Hamiltonian razcepili na operatorja A in B , eden delce „prime“ na sodih, drugi pa na lihih mestih. Če to matriko zapišemo zgolj za dvodelčno stanje, lahko hamiltonian zapišemo s pomočjo matrike $A^{(2)}$, ki sedaj deluje le na enem kubitnem paru:

$$A^{(2)} = \begin{bmatrix} 1 & & & \\ & -1 & 2 & \\ & 2 & -1 & \\ & & & 1 \end{bmatrix} = h_{01},$$

kjer moramo s to matriko izmenično delovati na pare fermionov – enkrat na vse sode, drugič na vse lihe. Hamiltonian ima še dele, s katerim delujemo na lihe. Radi bi ga zapisali v isti bazi, kot h_{01} , da ga bomo lahko sešteli in tako imeli le en Hamiltonski blok za dva kubita. Po izračunu dobimo

$$h_{10} = \begin{bmatrix} 1 & & & \\ & -1 & 2 & \\ & 2 & -1 & \\ & & & 1 \end{bmatrix} = h_{01}$$

kar lahko z nekaj algebre potrdimo. Po definiciji mora veljati

$$H^{(2)} = h_{01} + h_{10} = 2h_{01}, \quad (2)$$

ki ima lastne vrednosti $(-6, 2, 2, 2)$. Če je naša numerična simulacija pravilna, moramo dobiti torej dobiti $\langle H^{(2)} \rangle_\beta \xrightarrow{\beta \rightarrow \infty} -6$. Daljše kubitne verige bodo osnovno skalirale proti 0, ker imamo anti-feromagnet. Recimo, za verigo dolžine 3 se lahko v potu svojega obraza prebijemo do hamiltoniana

$$H^{(3)} = \begin{bmatrix} 3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 2 & 0 & 2 & 0 & 0 & 0 \\ 0 & 2 & -1 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & -1 & 2 & 0 \\ 0 & 0 & 0 & 2 & 0 & 2 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3 \end{bmatrix}, \quad (3)$$

ki se v diagonalni obliki zapiše kot $\text{diag}(-3, -3, -3, -3, 3, 3, 3, 3)$.

Vendar pa je narediti tako matriko težko in se potрати veliko procesorskega časa s Kroneckerjevim produktom, nato pa še z matričnim množenjem. Vendar pa mi lahko isto naredimo na boljši način. Izkoristimo namreč lahko to, da je naš vektor \underline{x} v resnici nekakšen spinor

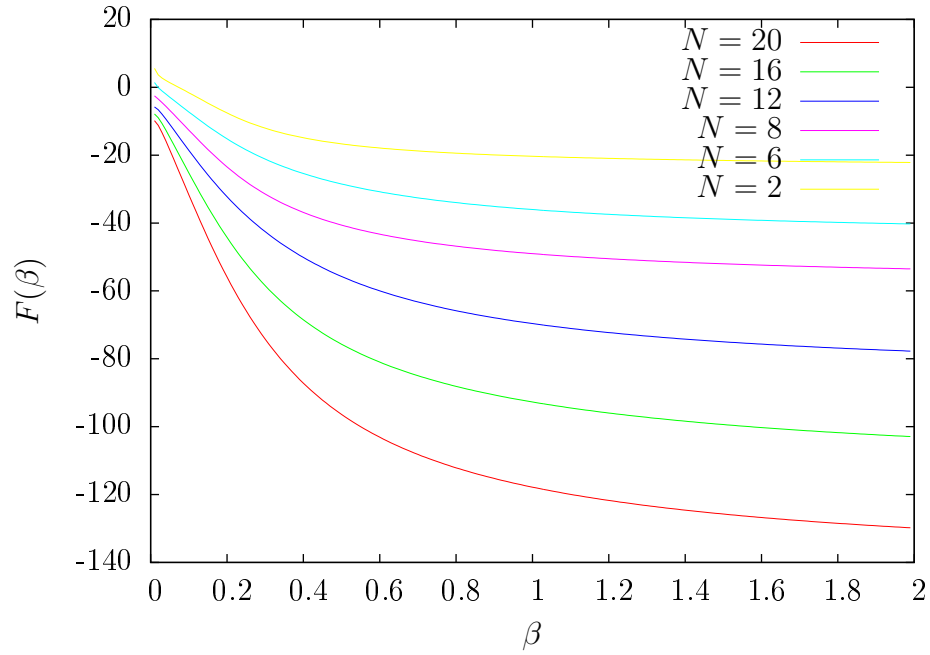
$$\underline{x} = x_0|000\rangle + x_1|001\rangle + x_2|010\rangle + x_3|011\rangle + \dots$$

komponente pa ima prikladno urejene. Naš hamiltonian je simetričen preko diagonale $H_{00} - H_{NN}$, vendar pa je simetričen tudi preko diagonale $H_{0N} - H_{N0}$, kar pomeni, da lahko število ničel in enic v stanju zamenjamo in bo matrični element isti.

Kakorkoli že, če naše indekse prepišemo v binarno, opazimo, da je hamiltonian odvisen le od indeksa našega stanja, saj denimo $|000\dots 0\rangle$ vedno stoji na prvem mestu, ne glede na dimenzijo. Prav tako stanje $|000\dots 01\rangle$ ipd. Hamiltonska matrika je razpršena in za $N = 20$ je ogromna ter dejansko lahko zasede gigabajte našega pomnilnika, s selektivnim postopkom pretvorbe v binarno, pa lahko v nekaj sekundah izluščimo neničelne elemente in potem množimo le z njimi in tako prihranimo na času.

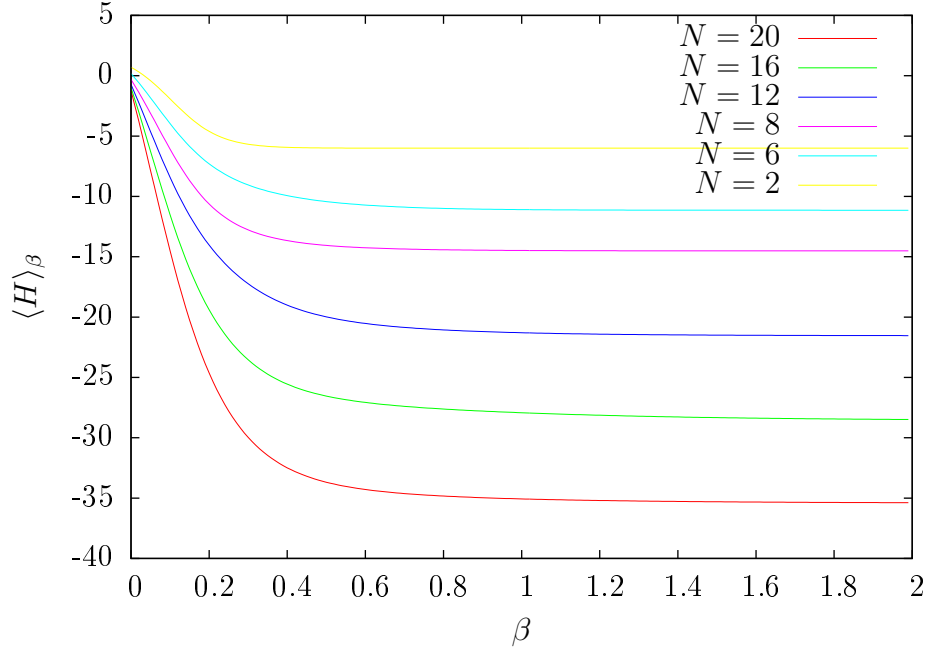
4 Rezultati

Dober pokazatelj, če naš integrator deluje pravilno, so grafi z energijo. Rezultati so konsistentni s pričakovanji.



Slika 1: Prosta energija se pogloblja z večanjem N . Padec ni čisto potenčen, vendar sploh ni eksponenten. Odvodi na začetku so napačni, vendar številka ni – v ničli namreč še vedno dobimo pol, ki ga program vrne pravilno. Če bi vzel krajši časovni korak, bi se to verjetno popravilo in rezultat ne bi bil dvomljiv zaradi odvoda.

Iz slike 1 sklepamo, da bo v termodinamski limiti prosta energija $F(\beta) \xrightarrow{\beta \rightarrow \infty} -\infty$. Iz teh slik morda to ni tako očitno, vendar prosta energija tu limitira proti končni vrednosti za velike β .

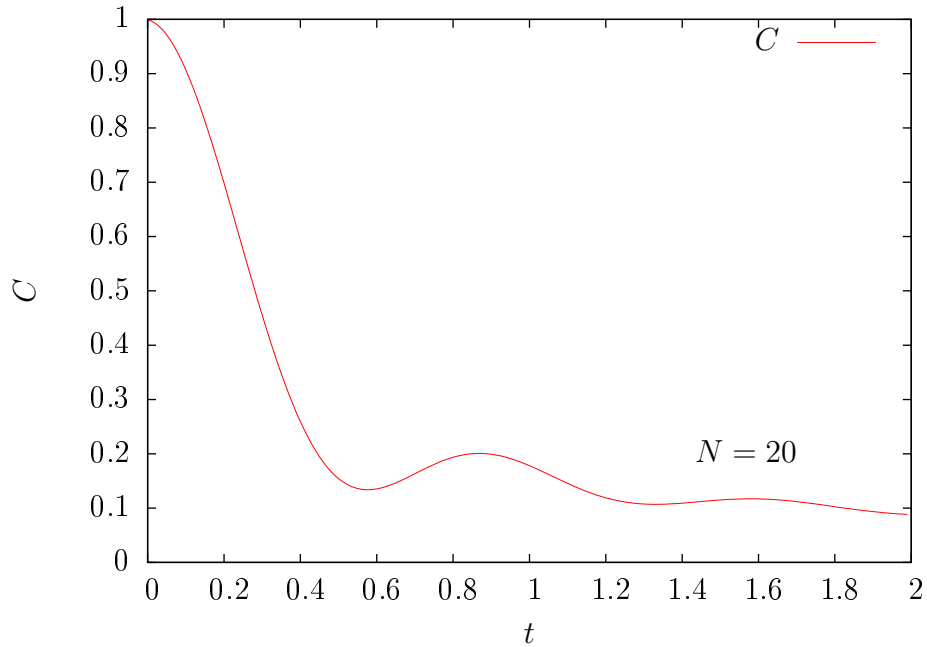


Slika 2: Energije osnovnih stanj za različne dolžine verig. Za $N = 2$ in $N = 4$ sem rešitev analitično primerjal in pride prava vrednost, tj. -6 in -8 . Za šalo sem poizkusil tudi s prej omenjeno verigo dolžine 3, katero moramo obravnavati kot aperiodično, vendar tudi v tem primeru dobimo pravo energijo, to je -3 .

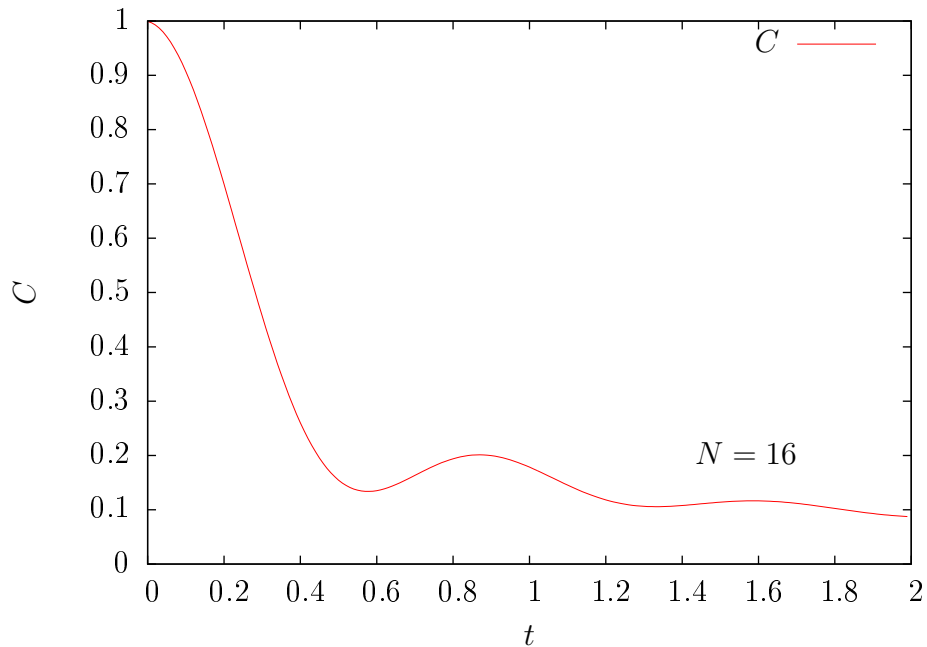
Iz slike 2 lahko sklepamo, da bo $\langle H \rangle_\beta \xrightarrow{\beta \rightarrow \infty} -\infty$. To je anti-feromagnet, kar pomeni, da se bolj splača imeti čim bolj neurejeno kubitno verigo. Ker imamo v termodinamski limiti neskončno stanj, pomeni da moramo pričakovati tako energijo ter da je v bistvu tako obnašanje pričakovano. Lihe verige, kot prej omenjeni hamiltonian za verigo dolžine tri se obnašajo nekako čudno, vendar nisem šel preverjati energije takih verig.

4.1 Časovna korelacija

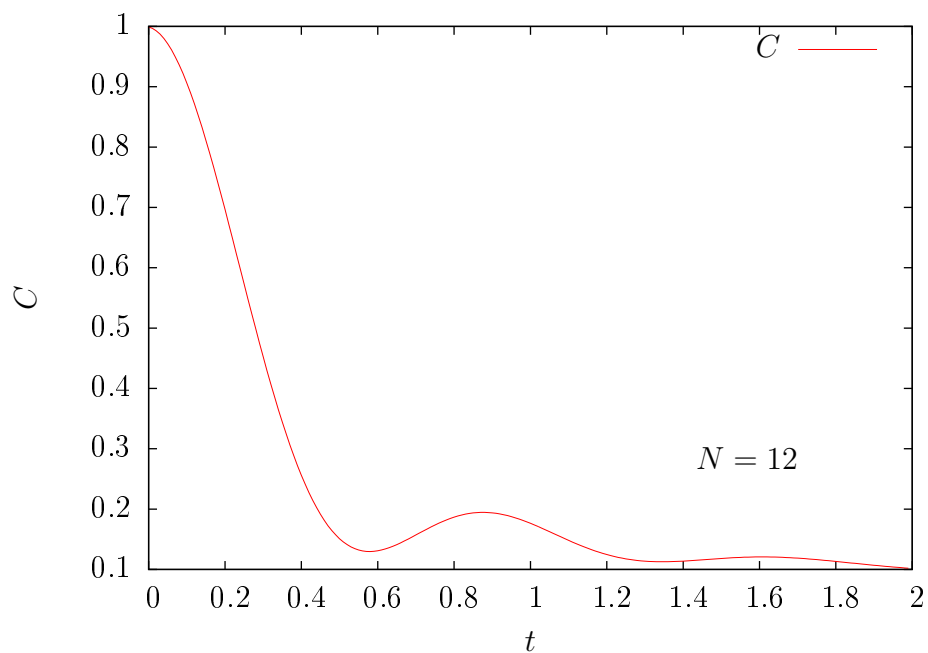
Avtokorelacijo prvega spina smo definirali kot $C(t) = \langle \sigma_1^z(t) \sigma_1^z(0) \rangle$, delamo v limiti $\beta \rightarrow 0$, torej pri neskončni temperaturi, kar pomeni, da je imaginarna komponenta časa enaka 0.



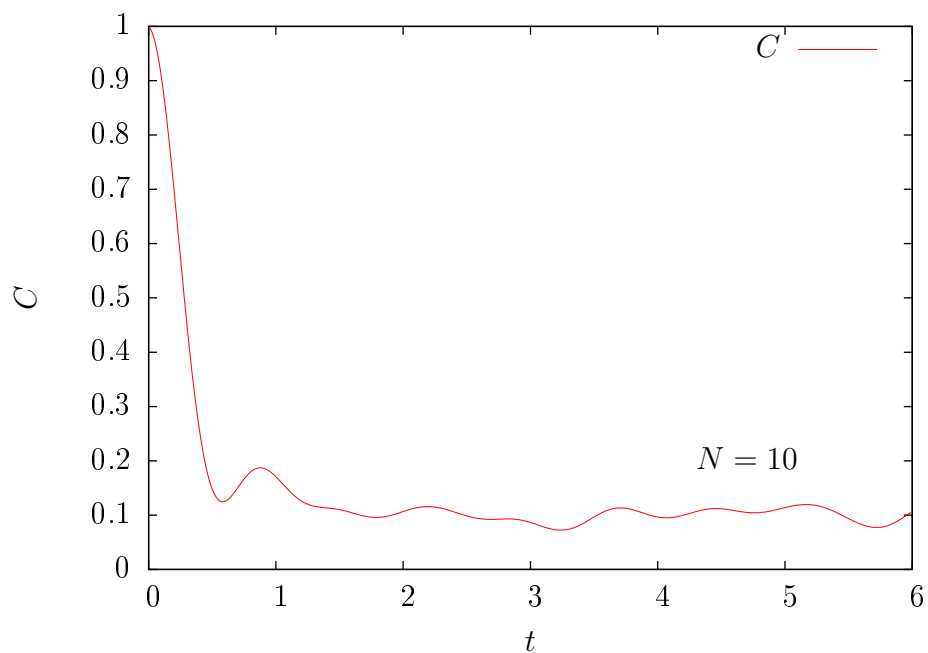
Slika 3: Korelacijska funkcija za 5 naključnih vektorjev. Naredil sem tudi z manjšo, vendar ni bilo razlike zaradi česar sklepam, da že zelo malo stanj dobro opiše naš problem.



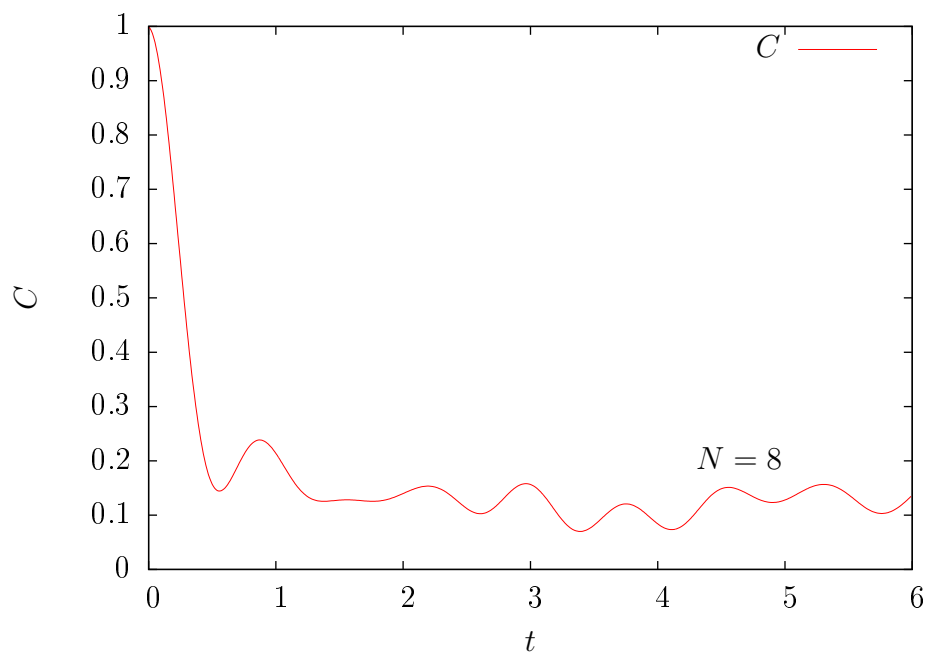
Slika 4: Korelacijska funkcija za kubitno verigo dolžine 16, z 10 naključnimi vektorji. Graf je identičen tistemu, ki bi ga dobili s petimi in celo tremi naključnimi vektorji.



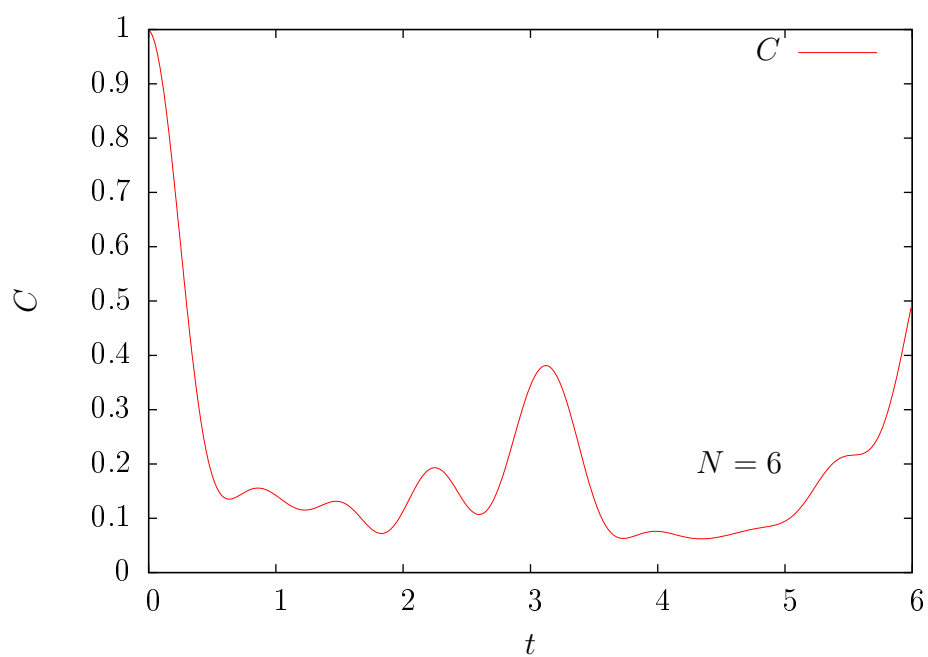
Slika 5: Slika je spet praktično enaka, zaključek je malo dvignjen pri $t = 2$, kar nam namiguje, da se bližamo meji izven režima termodinamske limite.



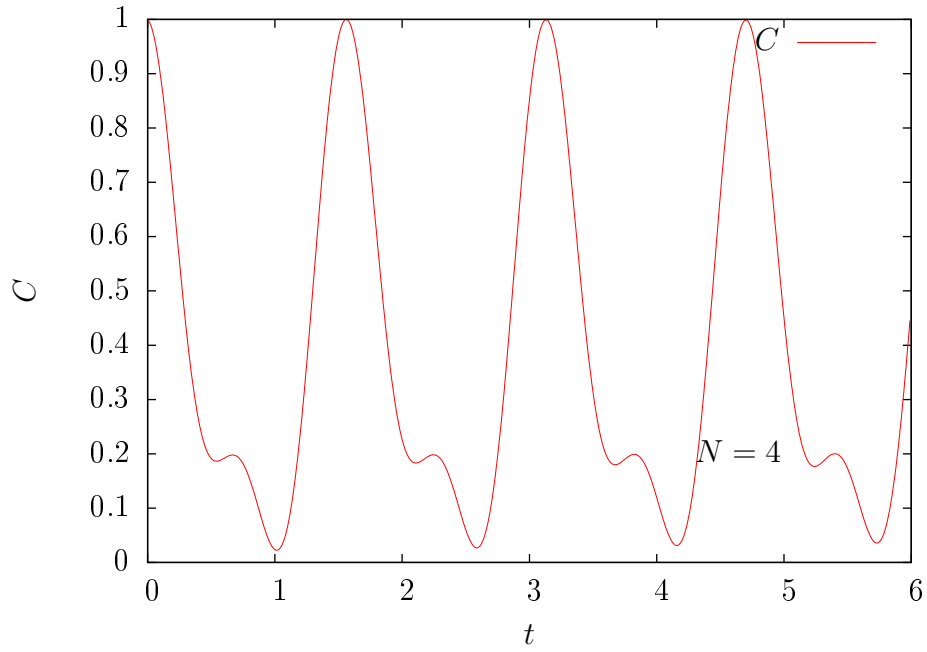
Slika 6: Pred časom $t = 2$, ki je bil prej meja grafov, se korelacijska funkcija prične dvigovati, kar pomeni da je čas za večjo sliko. Vidimo, da imamo še vedno normalno avtokorelacijsko funkcijo.



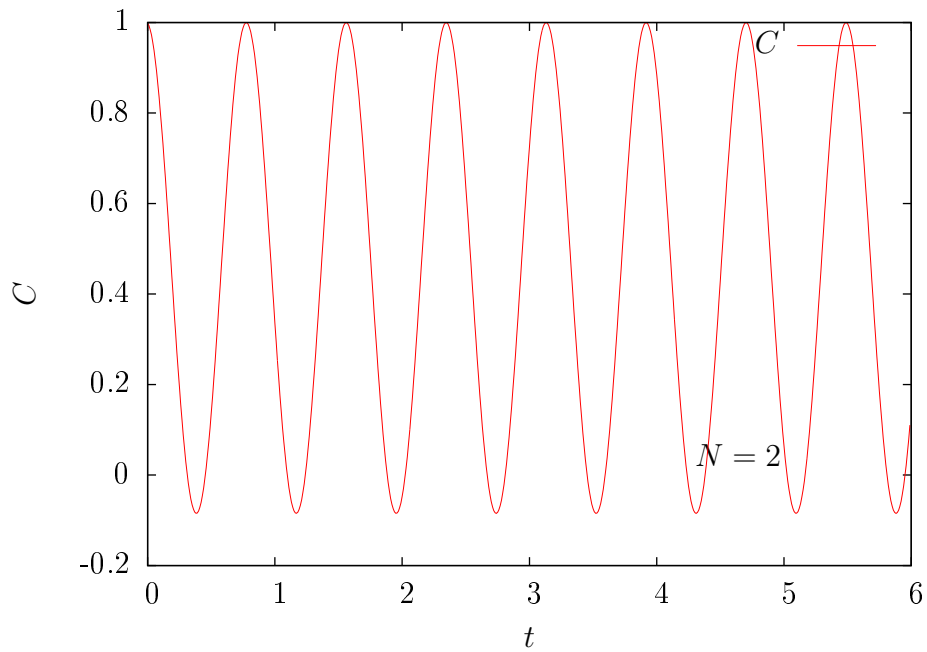
Slika 7: Graf je tu očitno drugačen od tistega, ko je $N = 10$.



Slika 8: V tem primeru nimamo več dobre avtokorelacijske funkcije, saj je naša končna veriga prekratka, da bi dobro simulirala neskončno. Avtokorelacijska funkcija postane periodična.



Slika 9: Periodičnost je tu še bolj očitna.



Slika 10: Veriga dolžine 2 vsekakor ne more simulirati termodinamske limite. Kar dobimo tu je za konstanto premaknjena čista kosinusna funkcija.

Iz teh slik smo se naučili, da za smiseln statistični opis potrebujemo kubitno verigo zadostne dolžine (tj. za naš primer vsaj $N = 8$). Kako hitro korelacijska funkcija pada proti nič lahko določimo s fitanjem.