

# Metropolisov algoritem

## 8. domača naloga pri predmetu Modelska analiza I

Domen Vaupotič

V nalogi obravnavamo Metropolis-Hastingsov algoritem, s katerim lahko naključno žrebamo vzorce, porazdeljene po neki porazdelitve  $\pi$ , iz katere je sicer zahtevno žrebrati. Osnovna ideja Metropolisovega algoritma je zasnova Markovske verige, ki ima stacionarno porazdelitev enako  $\pi$ . S simulacijo Markovske verige torej se torej po nekem času približamo stacionarni porazdelitvi.

Metropolisov algoritem je v fizikalnih problemih posebno uporaben za vzorčenje ravnovesnih stanj sistema (pri čemer običajno obravnavamo kanonično porazdelitev pri dani temperaturi  $T$ ), ki je pogosto visokodimenzionalen. Stanje  $X$  s pripadajočo energijo  $E(X)$  ima tedaj verjetnost sorazmerno z  $e^{-E(X)/(kT)}$ .

Algoritem poteka po sledečih korakih:

1. Žrebamo naključno stanje  $X_0$
2. Tvorimo dovoljeno potezo  $X_t \rightarrow X_{t+1}$
3. Novo potezo sprejmemo, če:
  - $\Delta E = E(X_{t+1}) - E(X_t) < 0$  ali
  - $\xi \leq e^{-\Delta E/(kT)}$ , pri čemer žrebamo naključno  $\xi \approx U(0, 1)$ .
4. Ponavljamo koraka 2 in 3.

Brez izgube splošnosti bomo v vseh nalogah uporabljali reskaliranje temperature s  $k = 1$ .

## 1. Molekularna verižnica

Obravnavajmo sprva diskretno molekularno verižnico, sestavljeno iz 17 členkov, ki je obešena na svojih koncih. Vsak izmed členkov lahko zavzame enega izmed 19 višinskih nivojev. Celotno energijo verižnice sestavimo iz potencialnega prispevka, ki ga utežimo s prefaktorjem  $\alpha$ , in prožnostnega dela, ki predstavlja napetost vezi med dvema sosedoma.

$$E(X) = \alpha \sum_{i=1}^N h_i + \frac{1}{2} \sum_{i=1}^{N-1} (h_{i+1} - h_i)^2$$

Poteza v Metropolisovem algoritmu bo premik naključnega členka navzgor ali navzdol za en nivo, pri čemer obstaja najnižji možni nivo (-19).

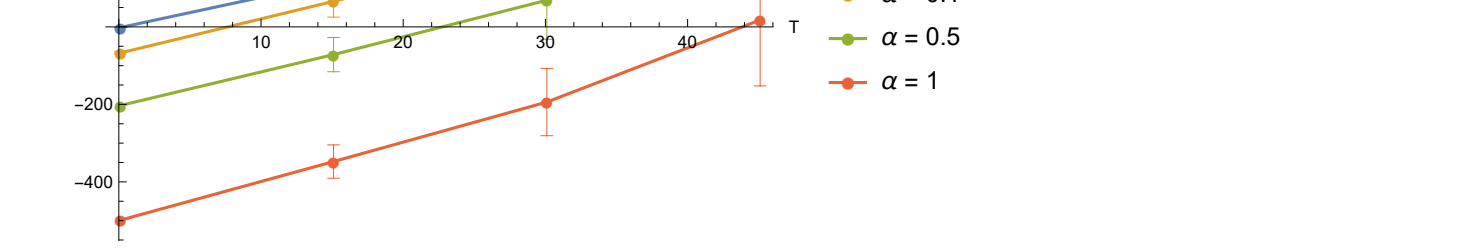
```
poteza[X_, α_, T_, hmin_ := -19] := Module[{i, δi, ΔE, Xn},
  i = RandomInteger[{2, Length[X] - 1}];
  δi = RandomChoice[{1, -1}];
  ΔE = δi^2 - δi (X[[i + 1]] - 2 X[[i]] + X[[i - 1]] - α);
  Xn = X;
  Xn[[i]] += δi;
  If[Xn[[i]] ≥ hmin ∧ (ΔE < 0 ∨ RandomReal[] ≤ Exp[-ΔE/T]), Sow[1]; Xn, Sow[0]; X]
]
```

energija[X\_, α\_] := α Total[X] +  $\frac{1}{2}$  Total[Differences[X]^2]

### Časovni potek

Oglejmo si najprej primere ravnovesnih stanj verižnice pri različnih temperaturah in izbranem parametru  $\alpha$ . Opazimo, da višja temperatura povzroči večje energijske fluktuacije. Z večjim koeficientom  $\alpha$  se več osrednjih členkov verižnice posede v najnižji možni energijski nivo. To je skladno z intuitivnim razumevanjem parametra, ki predstavlja razmerje med potencialnim prispevkom in prožnostnim prispevkom k energiji.

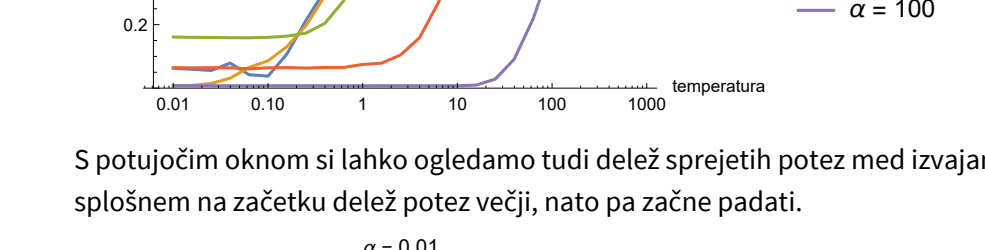
Časovni potek energije sestavljata značilni fazi: faza začetnih vzorcev (*burn-in*), ki traja prvih nekaj 1000 korakov, in faza stacionarnih vzorcev.



### Ovisnost ravnovesne energije od temperature in parametra α

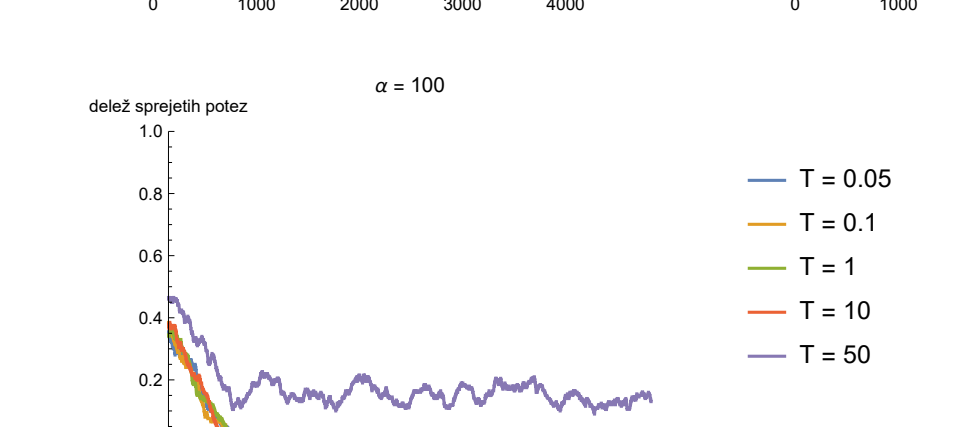
Za izračun ravnovesne energije moramo preiti fazo začetnih vzorcev in izračunati statistiko na prvi stacionarnih vzorcev.

Spodnji graf prikazuje ravnovesno energijo in standardni odklon žrebanih stanj, ki smo ju izračunali s povprečenjem žrebanih stanj med 5 000. in 100 000. korakom. Povprečna energija s temperaturo narašča, z njo pa tudi fluktuacije energije v ravnovesju – oboje skladno s splošno statistično fiziko.

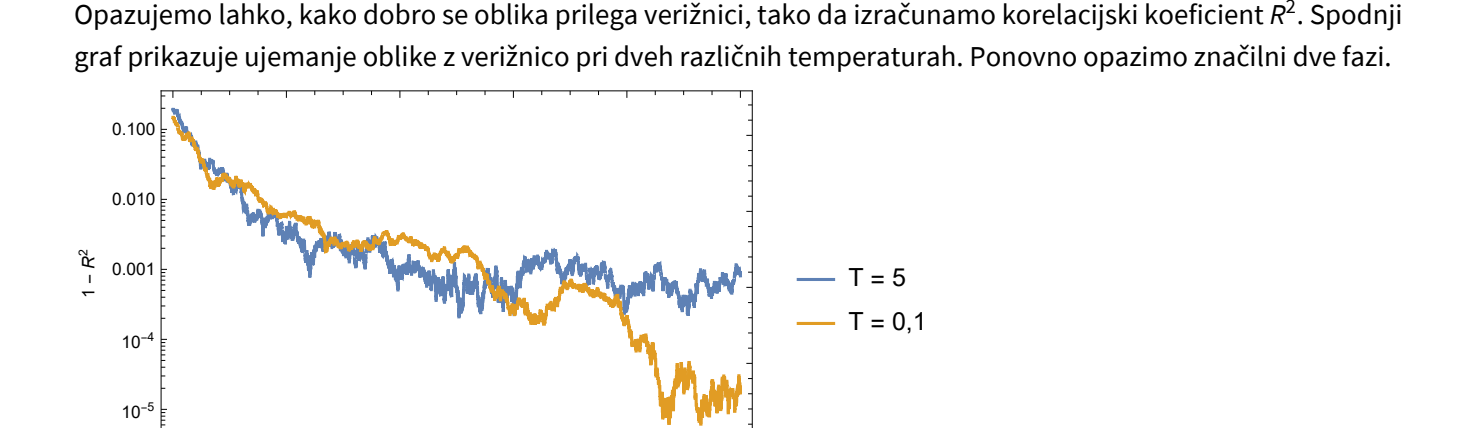
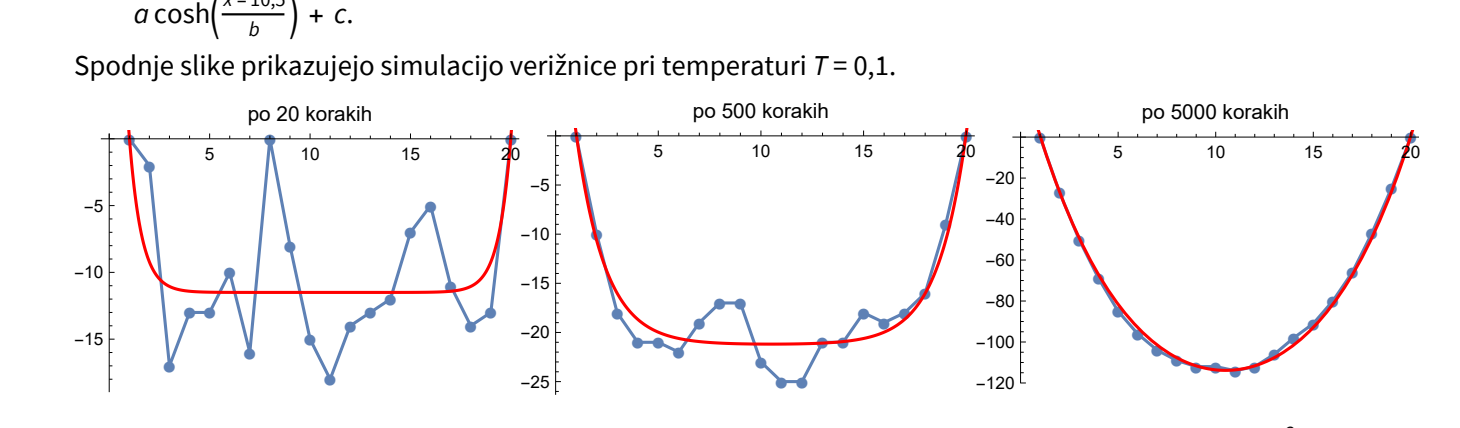


### Delež sprejetih potez

Med izvajanjem Metropolisovega algoritma lahko spremljamo tudi delež sprejetih potez. Pričakujemo, da bo pri višjih temperaturah delež sprejetih potez večji. Zanimivo opazimo, da pri nižjih temperaturah pomembno vpliva tudi parameter  $\alpha$ : pri zelo majhnih ali zelo velikih vrednostih je delež sprejetih potez nižji.



S potujočim oknom si lahko ogledamo tudi delež sprejetih potez med izvajanjem algoritma. Opazimo, da je v splošnem na začetku delež potez večji, nato pa začne padati.

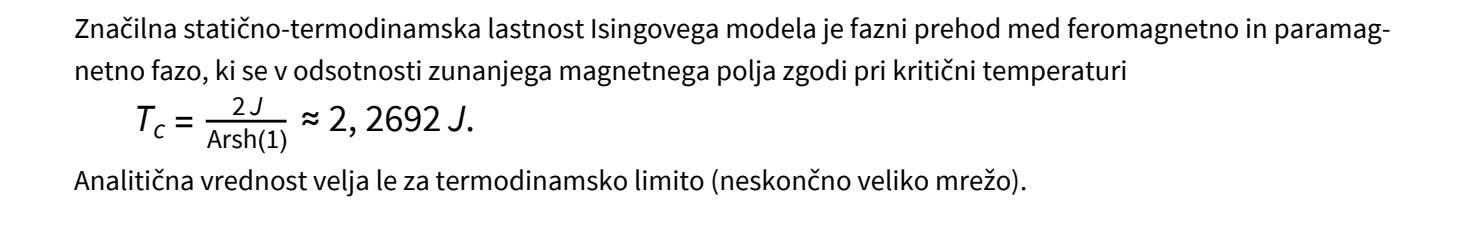


### Limita verižnice

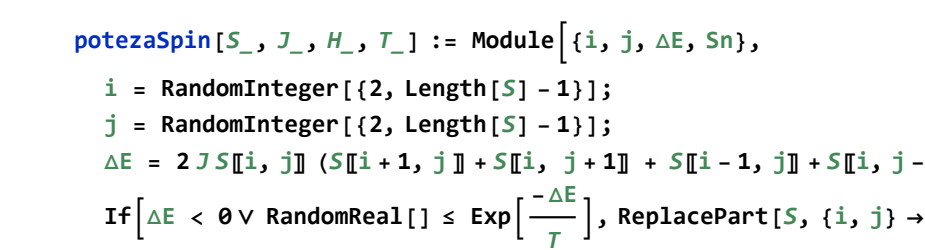
Če dovolimo, da členki zavzamejo poljubno višino, ugotovimo, da ravnovesna oblika sovпада s krivuljo verižnico s splošnim predpisom:

$$a \cosh\left(\frac{x-10.5}{b}\right) + c.$$

Spodnje slike prikazujejo simulacijo verižnice pri temperaturi  $T = 0.1$ .



Opazujemo lahko, kako dobro se oblika prilega verižnici, tako da izračunamo korelacijski koeficient  $R^2$ . Spodnji graf prikazuje ujemanje oblike z verižnico pri dveh različnih temperaturah. Ponovno opazimo značilni dve fazi.



## 2. Dvodimenzionalni Isingov model

V drugem delu naloge obravnavamo klasični dvodimenzionalni Isingov model s Hamiltonianom

$$\mathcal{H}(X) = -J \sum_{i,j} S_i S_j - H \sum_i S_i,$$

pri čemer parameter  $J$  predstavlja način sklopitve med spini (+1 za feromagnetno in -1 za antiferomagnetno),  $H$  pa zunanje magnetno polje. Konfiguracijska matrika  $X$  je velikosti  $L \times L$  in predstavlja stanje spinov (+1 ali -1). Poteza v algoritmu bo sprememba orientacije naključno izbranega spina.

Značilna statično-termodinamska lastnost Isingovega modela je fazni prehod med feromagnetno in paramagnetno fazo, ki se v odsotnosti zunanjega magnetnega polja zgodi pri kritični temperaturi

$$T_c = \frac{2J}{\text{Arsh}(1)} \approx 2,2692 J.$$

Analitična vrednost velja le za termodinamsko limito (neskončno veliko mrežo).

Opazujemo lahko značilne termodinamske spremenljivke:

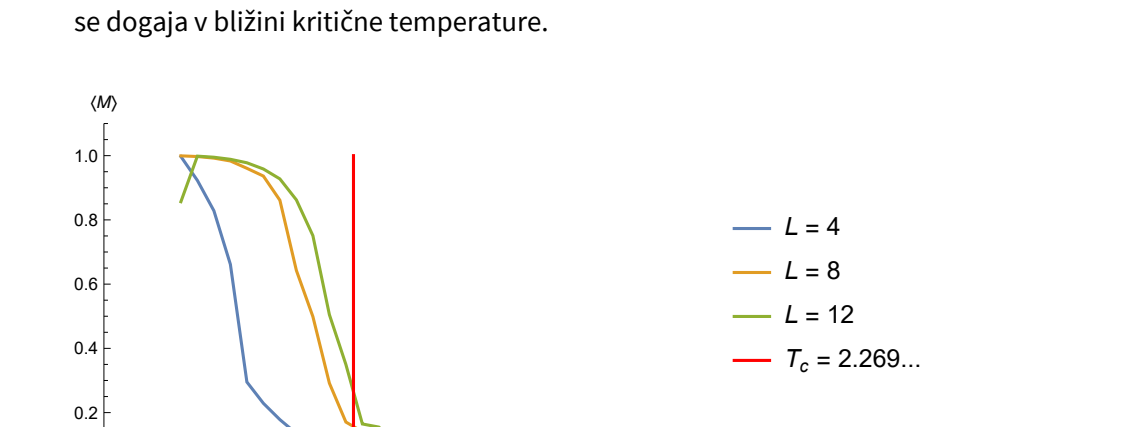
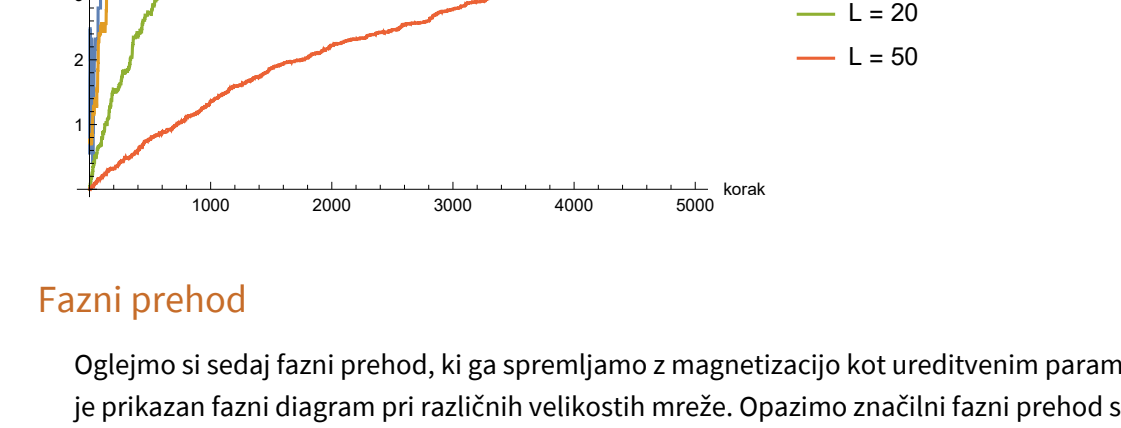
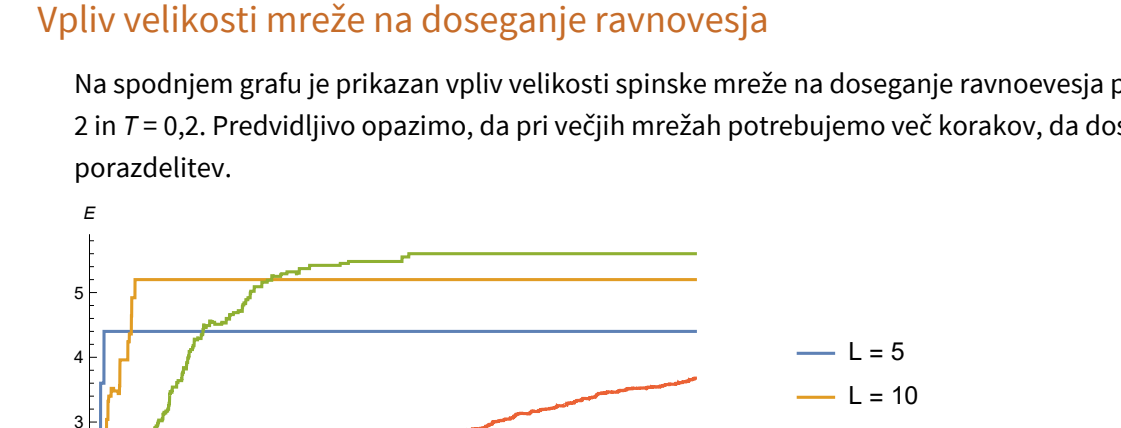
$$M = \frac{1}{L^2} \sum_i S_i \quad \text{magnetizacija}$$
$$\chi = \frac{\langle M^2 \rangle - \langle M \rangle^2}{L^2 T} \quad \text{susceptibilnost}$$
$$c = \frac{\langle E^2 \rangle - \langle E \rangle^2}{L^2 T^2} \quad \text{specifična toplota}$$

Poteza v algoritmu bo sprememba orientacije naključno izbranega spina.

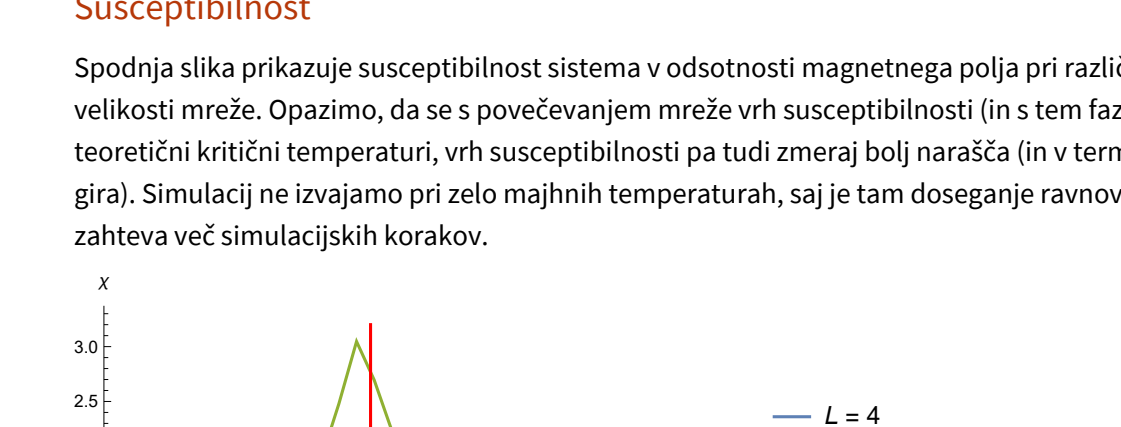
```
potezaSpin[S_, J_, H_, T_] := Module[{i, j, ΔE, Sn},
  i = RandomInteger[{2, Length[S] - 1}];
  j = RandomInteger[{2, Length[S] - 1}];
  ΔE = 2 J S[[i, j]] (S[[i + 1, j]] + S[[i, j + 1]] + S[[i - 1, j]] + S[[i, j - 1]]) + 2 H S[[i, j]];
  If[ΔE < 0 ∨ RandomReal[] ≤ Exp[-ΔE/T], ReplacePart[S, {i, j} → -S[[i, j]], S]
]
```

Oglejmo si najprej dinamiko algoritma in značilne ravnovesne vzorce pri različnih temperaturah in zunanjem magnetnem polju. Spodnje slike prikazujejo potek Metropolisovega algoritma (začetno stanje, po 10 000 korakov in po 50 000 korakih) za mrežo spinov velikosti  $50 \times 50$ .

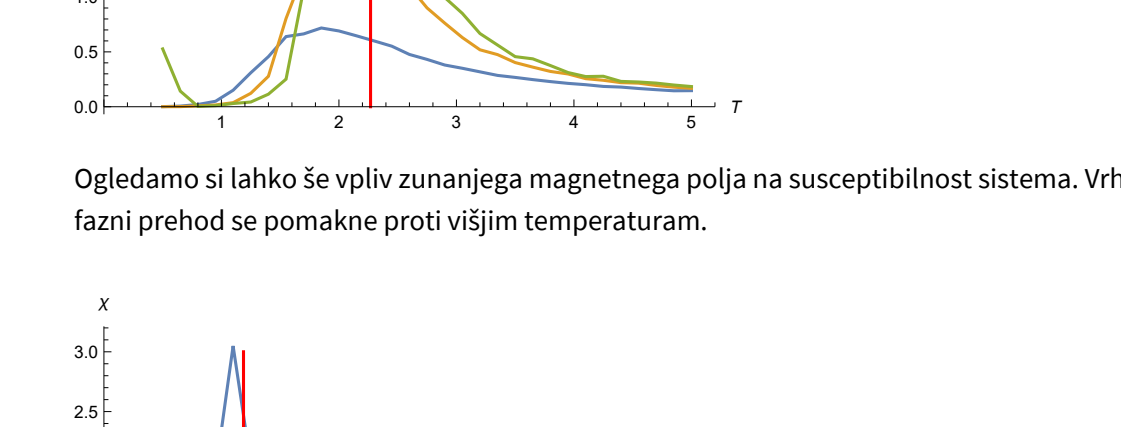
### Brez zunanjega magnetnega polja



### Vključeno magnetno polje

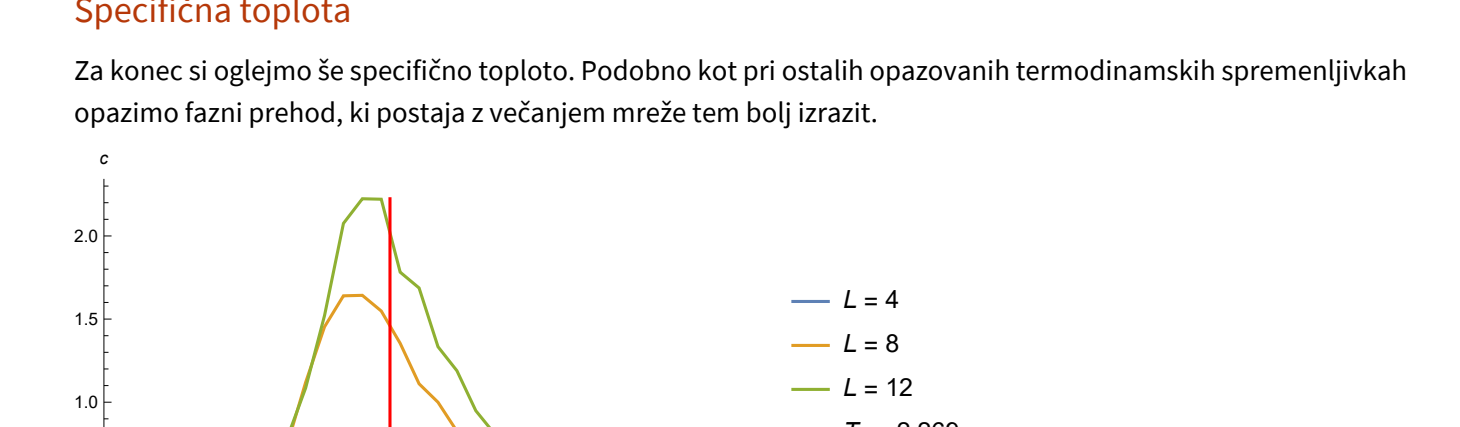


### Antiferomagnetizem



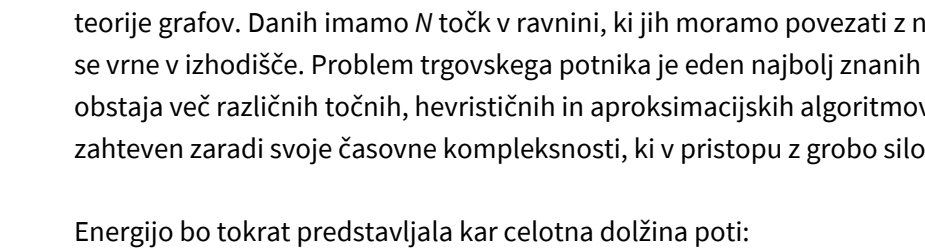
### Časovni potek algoritma

Podobno kot pri molekularni verižnici lahko opazimo značilni dve fazi v časovnem poteku algoritma. Pod kritično temperaturo opazimo značilnosti feromagnetne faze, nad njo pa značilnosti paramagnetne faze. Grafi prikazujejo sistem velikosti  $L = 15$  v odsotnosti zunanjega polja.



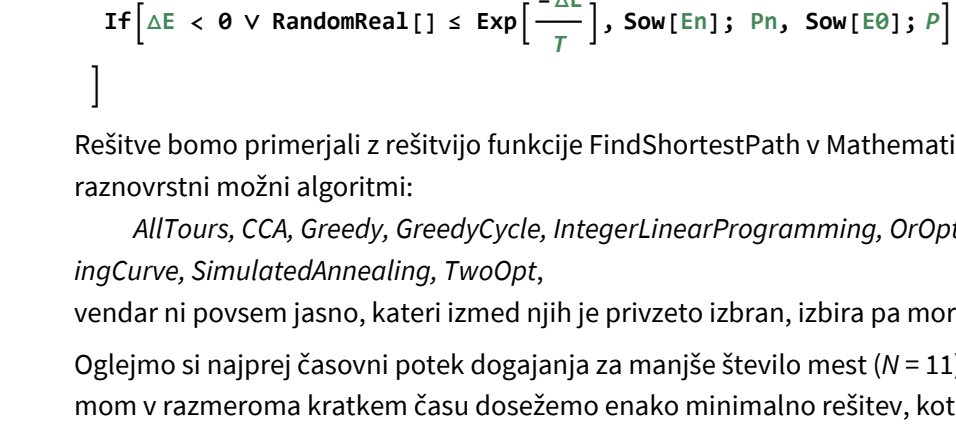
### Vpliv velikosti mreže na doseganje ravnovesja

Na spodnjem grafu je prikazan vpliv velikosti spinske mreže na doseganje ravnovesja pri izbranih parametrih  $H = 2$  in  $T = 0.2$ . Predvidljivo opazimo, da pri večjih mrežah potrebujemo več korakov, da dosežemo ravnovesno porazdelitev.



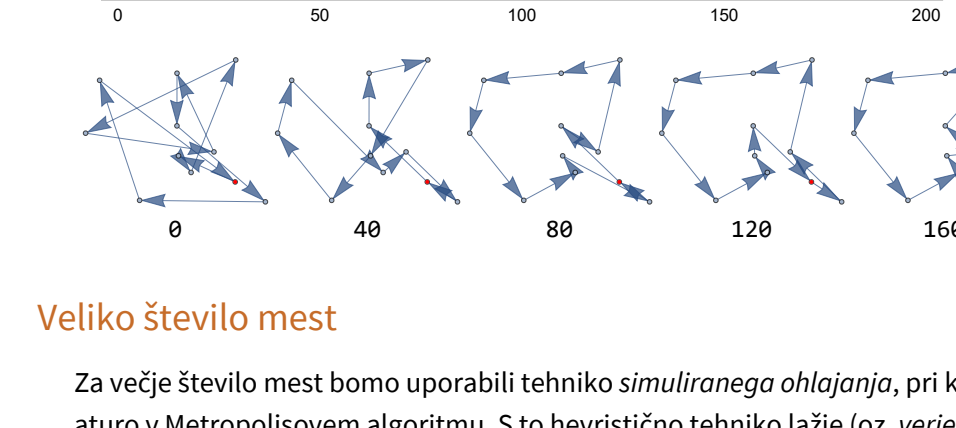
### Fazni prehod

Oglejmo si sedaj fazni prehod, ki ga spremljamo z magnetizacijo kot ureditvenim parametrom. Na spodnjem grafu je prikazan fazni diagram pri različnih velikostih mreže. Opazimo značilni fazni prehod s korenensko odvisnostjo, ki se dogaja v bližini kritične temperature.

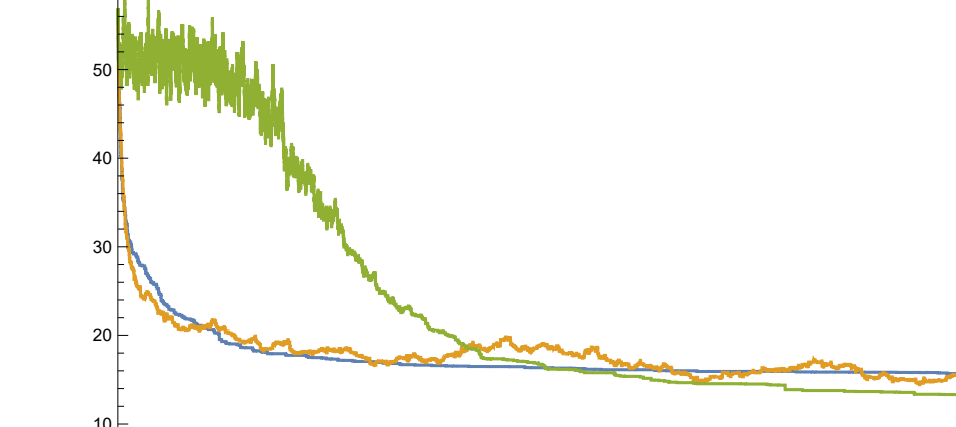


### Susceptibilnost

Spodnja slika prikazuje susceptibilnost sistema v odsotnosti magnetnega polja pri različnih temperaturah in velikosti mreže. Opazimo, da se s povečevanjem mreže vrh susceptibilnosti (in s tem fazni prehod) premika proti teoretični kritični temperaturi, vrh susceptibilnosti pa tudi zmeraj bolj narašča (in v termodinamski limiti divergira). Simulaciji ne izvajamo pri zelo majhnih temperaturah, saj je tam doseganje ravnovesja bistveno počasnejše in zahteva več simulacijskih korakov.

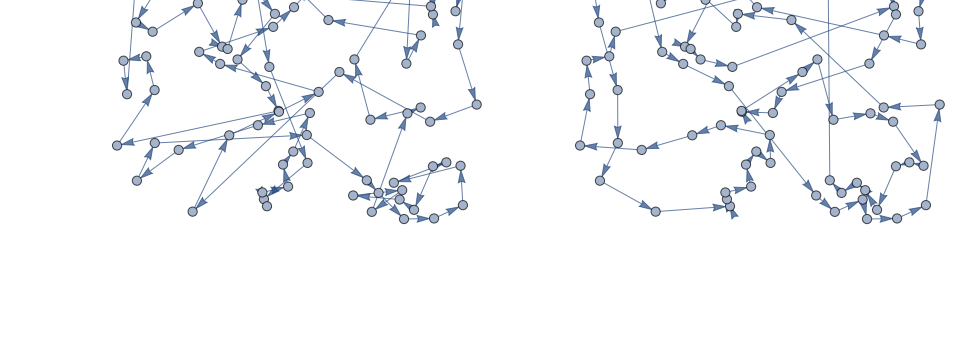


Ogledamo si lahko še vpliv zunanjega magnetnega polja na susceptibilnost sistema. Vrh susceptibilnosti in s tem fazni prehod se pomakne proti višjim temperaturah.



### Specifična toplota

Za konec si oglejmo še specifično toplotu. Podobno kot pri ostalih opazovanih termodinamskih spremenljivkah opazimo fazni prehod, ki postaja z večanjem mreže tem bolj izrazit.



## 3. Problem trgovskega potnika

V tretjem primeru uporabe Metropolisovega algoritma si bomo ogledali znani problem trgovskega potnika iz teorije grafov. Danih imamo  $N$  točk v ravnini, ki jih moramo povezati z najkrajšo možno potjo, ki obišče vse točke in se vrne v izhodišče. Problem trgovskega potnika je eden najbolj znanih optimizacijskih problemov, za katerega obstaja več različnih točnih, heurističnih in aproksimacijskih algoritmov. S kombinatoričnega vidika je problem zahteven zaradi svoje časovne kompleksnosti, ki v pristopu z grobo silo znaša  $O(N!)$ .

Energijo bo tokrat predstavljala kar celotna dolžina poti:

$$E = \sum (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2,$$

tipična poteza pa bo zamenjava dveh naključno izbranih točk v zaporedju (vendar ne prve in zadnje).

```
potezaTrgovec[P_, T_] := Module[{i, j, ΔE, E0, En, Pn},
  {i, j} = RandomSample[Range[2, Length[P] - 1], 2];
  E0 = Total@Sqrt@Total[Differences[P]^2, {2}];
  Pn = Permute[P, Cycles[{i, j}]]];
  Pn = Total@Sqrt@Total[Differences[Pn]^2, {2}];
  ΔE = En - E0;
  If[ΔE < 0 ∨ RandomReal[] ≤ Exp[-ΔE/T], Sow[En]; Pn, Sow[E0]; P]
]
```

Rešitve bomo primerjali z rešitvijo funkcije FindShortestPath v Mathematici. V dokumentaciji so omenjeni raznovrstni možni algoritmi:

AllTours, CCA, Greedy, GreedyCycle, IntegerLinearProgramming, OrOpt, OrZweig, RemoveCrossings, SpaceFill-ingCurve, SimulatedAnnealing, TwoOpt,

vendar ni povsem jasno, kateri izmed njih je privzeto izbran, izbira pa morda zavisi tudi od velikosti problema. Oglejmo si najprej časovni potek dogajanja za manjše število mest ( $N = 11$ ). Opazimo, da z Metropolisovim algoritmom v razmeroma kratkem času dosežemo enako minimalno rešitev, kot jo podaja Mathematica.



### Veliko število mest

Za večje število mest bomo uporabili tehniko *simuliranega ohlajanja*, pri katerem postopoma znižujemo temperaturo v Metropolisovem algoritmu. S to heuristično tehniko lažje (oz. *verjetneje*) najdemo globalni ekstrem iskane funkcije (v našem primer zaporedje mest z najkrajšo potjo).

Spodaj je prikazan primer za 100 mest z eksponentnim ohlajanjem, pri čemer opazimo, da s 30 000 koraki nismo uspelo doseči iste rešitve.

