



# Modelska Analiza 1

## 10. naloga - Metropolisov algoritem

Avtor: Matic Lubej  
Asistent: dr. Simon Čopar  
Predavatelj: prof. dr. Alojz Kodre

Ljubljana, 18.12.2013

---

### Naloga:

Pri tej nalogi smo še bolj nadgradili svoje znanje o naključnih številih in Monte Carlo procesi. Spoznali smo Metropolisov algoritem, ki se uporablja za bolj razgibane funkcije, oz takšne porazdelitve, kjer preprosto preizkušanje pogoja kot pri MC metodi ne zadostuje več. V prvem delu naloge smo algoritem preizkusili na molekularni verigi, na katero vpliva potencialna energija in prožnostna energija vezi med členi. S tem algoritmom smo poiskali ravnovesno stanje in vpliv temperature na to stanje. Pri drugi nalogi smo spoznali Isingov model 2D mreže spinov, pri katerem smo raziskovali energijo, magnetizacijo, magnetno susceptibilnost in specifično toploto. V zadnjem delu naloge smo reševali problem trgovskega potnika, kjer smo morali najti najkrajšo pot med danimi mesti.

# Uvod

Metropolisov algoritem je algoritem, ki z uporabo naključnih števil pri metodi Monte Carlo omogoči, da dobimo zaporedje naključnih vzorcev iz neke verjetnostne porazdelitve, ki na zanima, direktno vzorčenje takšne porazdelitve pa je preveč zahtevno. Algoritem lahko uporabimo za aproksimacijo porazdelitve, ali izračun integrala, kar je še posebej uporabno za več dimenzionalne integrale. Uporaben je predvsem za minimizacijo razgibanih funkcij, kjer bi večina metod končala v lokalnih minimumih.

V praksi, ko je  $N$  velik, je glavna težava kako praktično vzorčiti primerke  $\vec{x}_j$  iz porazdelitve  $w(\vec{x})$ . Če bi konstruirali vzorčenje po principu „poskusi in zavrzi“ bi v režimu velikih  $N$  tipično sprejeli le eksponetno majhen delež poskusov. Uporabna ideja pa se porodi o razmišljanju o konstrukciji markovske verige, oziroma neke vrste naključnega sprehoda, po faznem prostoru, kjer z majhnimi koraki, potezami, skonstruiramo  $w(\vec{x})$  z deležem sprejetih potez, ki je tipično reda velikosti  $\sim 1$ , vendar za ceno dejstva, da so zaporedni poskusi precej slučajno odvisni. Najpreprostejši algoritem te vrste je ravno Metropolisov algoritem. Če imamo torej neko znano porazdelitev  $w(\vec{x})$ , po kateri želimo vzorčiti, velja, da je  $w(x \rightarrow x')$  t.i. prehodna verjetnost nekega markovskega procesa. Vzorcena porazdelitev je stacionarna natanko tedaj, ko prehodna verjetnost uboga princip *detajlnega ravnovesja* :

$$w(x)w(x \rightarrow x') = w(x')w(x' \rightarrow x), \quad (1)$$

kar pomeni, da mora biti verjetnost za prehod iz stanja  $\vec{x}$  v stanje  $\vec{x}'$  enaka verjetnosti za prehod iz stanja  $\vec{x}'$  v stanje  $\vec{x}$ . Sprehodi morajo torej biti reverzibilni. Boljšo predstavo dobimo pri konkretnih pregledih.

## Del I

# Molekularna verižica

## 1 Naloga

Imamo molekularno verigo, ki sestoji iz 17 členov, obešena pa je za oba konca. Vsak členek se lahko povesi od ničelne lege na poljubnega od nekega števila nivojev, kjer je število nivojev odvisno od razmerij gravitacijske konstante  $g$ , koeficienta vmesne vzmeti  $k$  in mase členkov  $m$ . S tem si verižica zmanjša potencialno energijo za eno enoto na nivo. Če pa s tem prenatagne vezi do sosedov, plača s prožnostno energijo, ki je za vsakega sosedu enaka kvadratu razlike v nivojskem številu. Določi ravnovesno energijo v odvisnosti od temperature. Za potezo lahko uporabiš spremembo za en nivo na izbranem slučajnem mestu.

## 2 Reševanje

Določiti želimo ravnovesno energijo v odvisnosti od temperature, kjer velja:

$$\begin{aligned} E_{pot} &= \sum_{i=1}^{17} mgh_i, \\ E_{el} &= \sum_{i=2}^{17} \frac{k}{2} (h_i - h_{i-1})^2, \\ E_0 &= E_{pot} + E_{el}, \end{aligned} \quad (2)$$

kjer sem si izbra koeficiente  $m = g = k = 1$ ,  $h_i$  pa predstavlja nivo posameznega členska.

Metropolisov algoritm uvedemo na naslednji način:

- naključno izžrebamo pozicijo  $i$
- predpostavimo potezo  $h_i \rightarrow h'_i$
- izračunamo razliko energije  $\Delta E = E'_0 - E_0$
- če velja  $\Delta E < 0$ , potezo sprejmemo
- če velja  $\Delta E > 0$ , potezo sprejmemo z verjetnostjo  $P = e^{-\frac{\Delta E}{E_T}}$ , kjer je  $E_T = \frac{1}{2}nk_B T$

Tudi tu sem brez izgube splošnosti predpostavil  $k_B = 1$ , saj smo tako netrivialno temperaturno območje postavili med 0 in 1. Opazimo, da z večanjem temperature povečujemo delež sprejetih potez in tako omogoča, da se rešimo kakšnega lokalnega minimuma. V limiti  $T \rightarrow \infty$  tako sprejmemo vse poteze in se sprehodimo po vseh možnih stanjih, v limiti  $T \rightarrow 0$  pa sprejmemo le najbolj optimalne poteze, oziroma tiste, ki zmanjšujejo energijo stanja proti osnovni energiji.

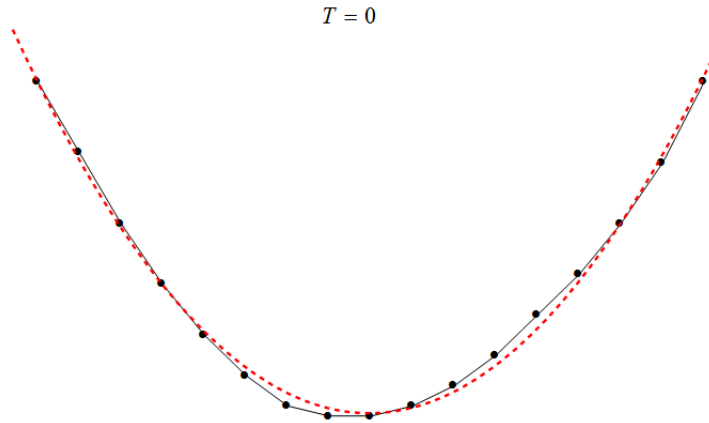
Zaradi spremembe le enega člena, lahko zapišemo razliko energij enostavneje kot:

$$\Delta E = E'_0 - E_0 = h'_i - h_i + \sum_{j \in [i-1, i+1]} \frac{1}{2} ((h'_i - h'_j)^2 - (h_i - h_j)^2), \quad (3)$$

saj tako ne računamo tistih energij, ki ostanejo nespremenjene in se na koncu odštejejo.

### 3 Rezultati

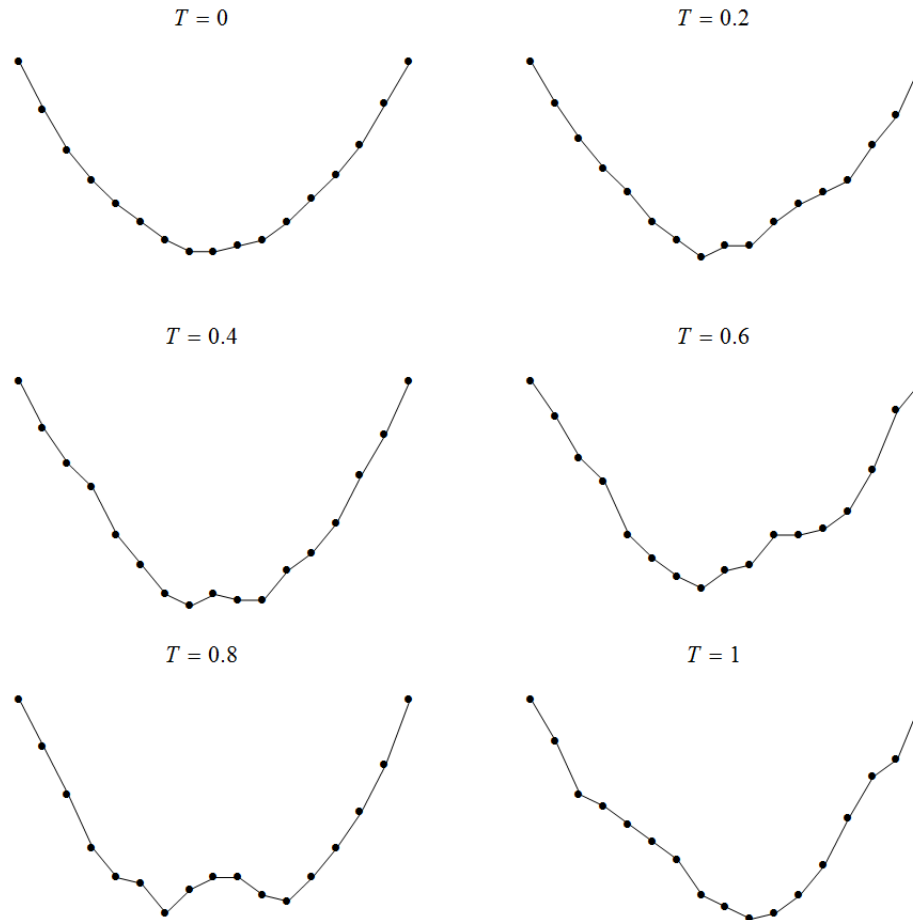
Ko takšen postopek večkrat ponovimo, na koncu dobimo rešitev. Pri temperaturi  $T = 0$  in številu potez  $N = 10000$  dobimo takšno osnovno stanje verige:



Vidimo, da je veriga res pripeta, največji odmik pa je na sredini. Takšno verižnico moramo v splošnem prestaviti in raztegniti v  $x$  in  $y$  smeri, tako da jo opisuje funkcija:

$$y = a + b \cosh(cx + d). \quad (4)$$

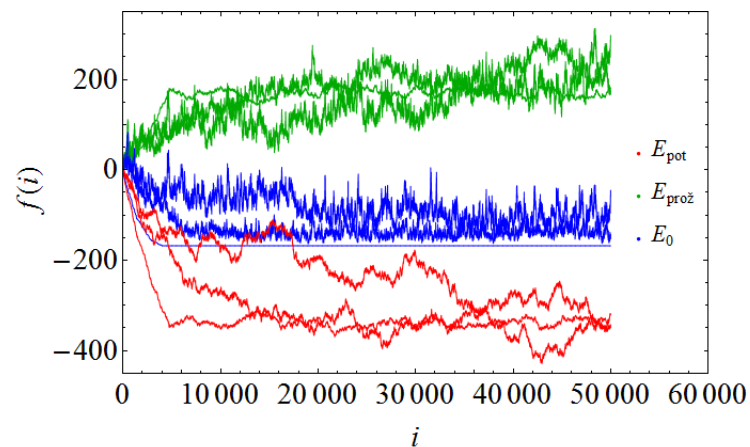
Če takšno verigo segrevamo, sprejemamo vedno več potez, ki ne predstavljajo znižanja energije. Takšne verige potem postajajo vedno bolj „divje“, kot je vidno na naslednjih slikah:



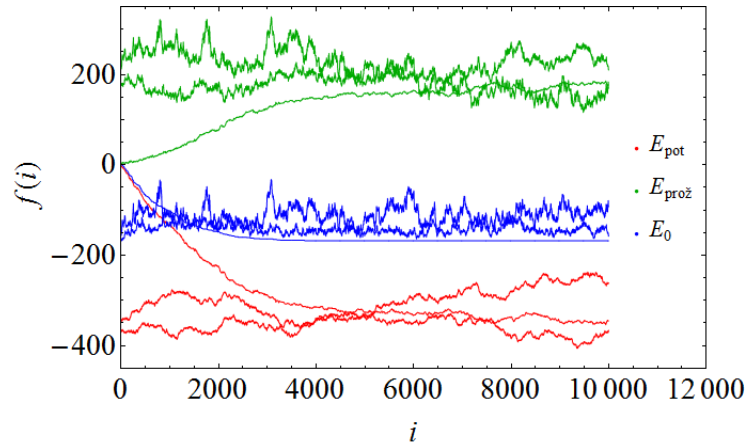
Vidimo, da so verige vedno bolj nepravilne oblike, kar je posledica višje temperature. Situacijo bolj nazorno prikazuje animacija, do katere vodi spodnja povezava:

<https://dl.dropboxusercontent.com/u/12747812/Animacije/veriga.gif>

Preden gremo na odvisnosti količin od temperature, si oglejmo odvisnos količin od števila iteracij v našem algoritmu, kar prikazuje naslednja slika, kjer sem vzel 50 000 iteracij:

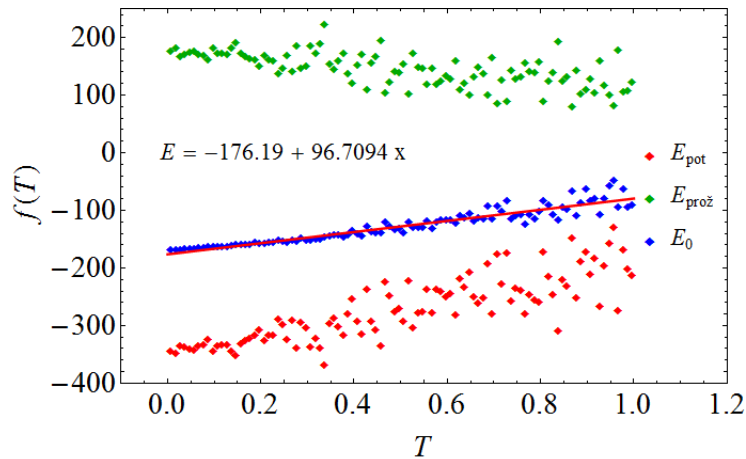


Zaradi večje nazornosti sem rezultate povezal s polno črto. Vidimo, da tiste krivulje, ki manj „divjajo“ pripadajo višjim temperaturam. Najbolj očitno je to pri temperaturi  $T = 0$ , kjer opazimo, da energija potrebuje nekaj časa, da pride do ravnovesja, potem pa v ravnovesju vztraja. Opazimo, da imamo torej nek termalizacijski čas, po katerem šele pride naša veriga v stacionarno stanje. Opazimo tudi, da se termalizacijski čas daljša z višjo temperaturo, kar pa je precej neugodno. Tu sem prišel na idejo, da vsakič, ko temperaturo zvišamo, vzamemo stanje od prej. Situacija je v tem primeru naslednja:



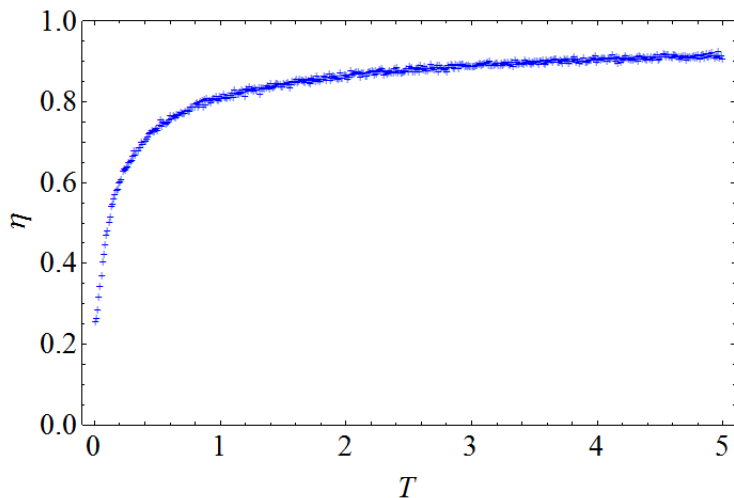
Vidimo, da rabimo termalizacijo le pri temperaturi  $T = 0$ , ki pa je nahitrejša, potem pa pri naslednji temperaturi uporabimo končno stanje prejšnje situacije za novo začetno stanje, ki je po vrednosti energije bližje, zato pa je tudi kraji relaksacijski čas. S tem nam je prikrajšana skrb o termalizacijskem času, naš program pa je bolj efektiven.

Sedaj smo pripravljeni, da izračunamo odvisnosti količin od temperature, kar sem storil za  $N = 10\,000$  potez in za 100 različnih temperatur na intervalu od 0 do 1. Naslednja slika prikazuje odvisnost potencialne, prožnostne in polne energije od temperature  $T$ :

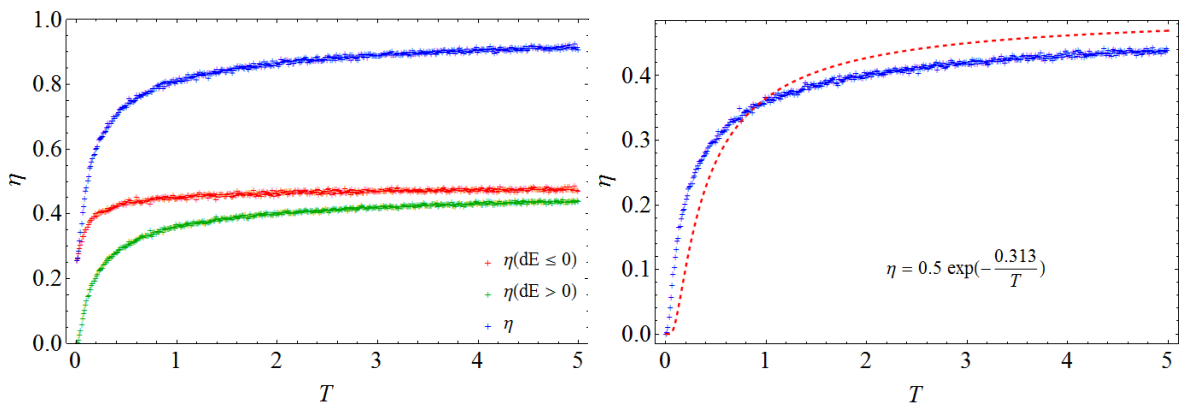


Na grafu je videti, da so vse količine linearno odvisne od temperature, le da z različnimi koeficienti. Opazimo, da se potencialna energija verige s temperaturo viša, kar je smiselno, saj se tudi težišče v povprečju višje pri višjih temperaturah, energija vezi pa je zaradi tega manjša, saj so si molekule vedno bližje skupaj. Prav tako potencialna in prožnostna veliko bolj nihata okoli srednje vrednosti, vidimo pa, da je njuna vsota veliko manj raztresena. Sklepamo, da je med prožnostno in potencialno energijo neka korelacija. Polna energija precej lepo sledi premici, katere enačba je napisana poleg.

Za konec si lahko ogledamo še delež sprejetih potez v odvisnosti od temperature. Že prej smo omenili, da z višanjem  $T$  omogočimo sprejetje vse manj ugodnih potez, torej pričakujemo, da se bo omenjena funkcija približevala vrednosti 1 v limiti  $T \rightarrow \infty$ . Poglejmo, če je to res:



Vidimo, da krivulja prikazuje to, kar smo pričakovali, hkrati pa smo se s tem tudi prepričali, da smo na pravem temperaturnem območju. Vidimo tudi, da v limiti  $T \rightarrow 0$  delež ne gre proti 0, saj smo računali tako, da smo vedno vzeli enako začetno stanje, celotni prispevek deleža pa je prišel na račun tistih, ki so energijo znižali ali pa pustili enako. V to se lahko prepričamo tako, da si ogledamo le odvisnost deleža tistih sprejetih potez, za katere velja  $\Delta E > 0$ :



Leva slika prikazuje odvisnost deleža optimalnih in neoptimalnih sprejetih potez, ter skupno število sprejetih potez, desna slika pa le neoptimalne sprejete poteze. Na levi sliki vidimo, da so bile naše domneve pravilne, saj je v limiti  $T = 0$  verjetnost za sprejem neoptimalnih potez enaka 0. Desna slika prikazuje še porazdelitveno funkcijo za dano odvisnost  $\eta(T)$ , vidimo, da je funkcija skoraj prava, vendar je potrebno opomniti, da imamo v algoritmu poleg spremenljivke  $T$  tudi spremenljivko  $\Delta E$ , tako da je zgornja krivulja vsota takšnih porazdelitev za različne fiksne parametre  $\Delta E$ .

## Del II

# Isingov model

## 1 Naloga

Isingov model feromagnetne in antiferomagnetne snovi v dveh dimenzijah opišemo s Hamiltonovim operatorjem

$$\mathcal{H} = -J \sum_{\langle i,j \rangle} s_i s_j - H \sum_i s_i, \quad J = \pm 1, \quad (5)$$

kjer je  $s_i = \pm 1$ . Prva vsota teče le po najbližjih sosedih. Če ni zunanega polja ( $H = 0$ ), kritična temperatura  $T_c$  faznega prehoda zadošča enačbi:

$$\sinh \frac{2J}{k_B T_c} \rightarrow T_c = 2.269185 \frac{J}{k_B}. \quad (6)$$

Določi povprečno energijo  $\langle E \rangle$  in lastno magnetizacijo  $\langle M \rangle$  v odvisnosti od temperature, kjer sta:

$$\langle E \rangle = \frac{1}{L^2} \langle \mathcal{H} \rangle, \quad (7)$$

$$\langle M \rangle = \frac{1}{L^2} \sum_{i=1}^n s_i, \quad (8)$$

kjer sta  $L$  dolžina spinske mreže in  $M$  magnetizacija sistema. Oglej si tudi spinsko susceptibilnost in specifično toploto pri različnih jakostih magnetnega polja:

$$\chi = \frac{\langle M^2 \rangle - \langle M \rangle^2}{k_B T}, \quad (9)$$

$$C_v = \frac{\langle E^2 \rangle - \langle E \rangle^2}{k_B T^2}. \quad (10)$$

## 2 Reševanje

Metropolisov postopek za Isingov model se glasi tako:

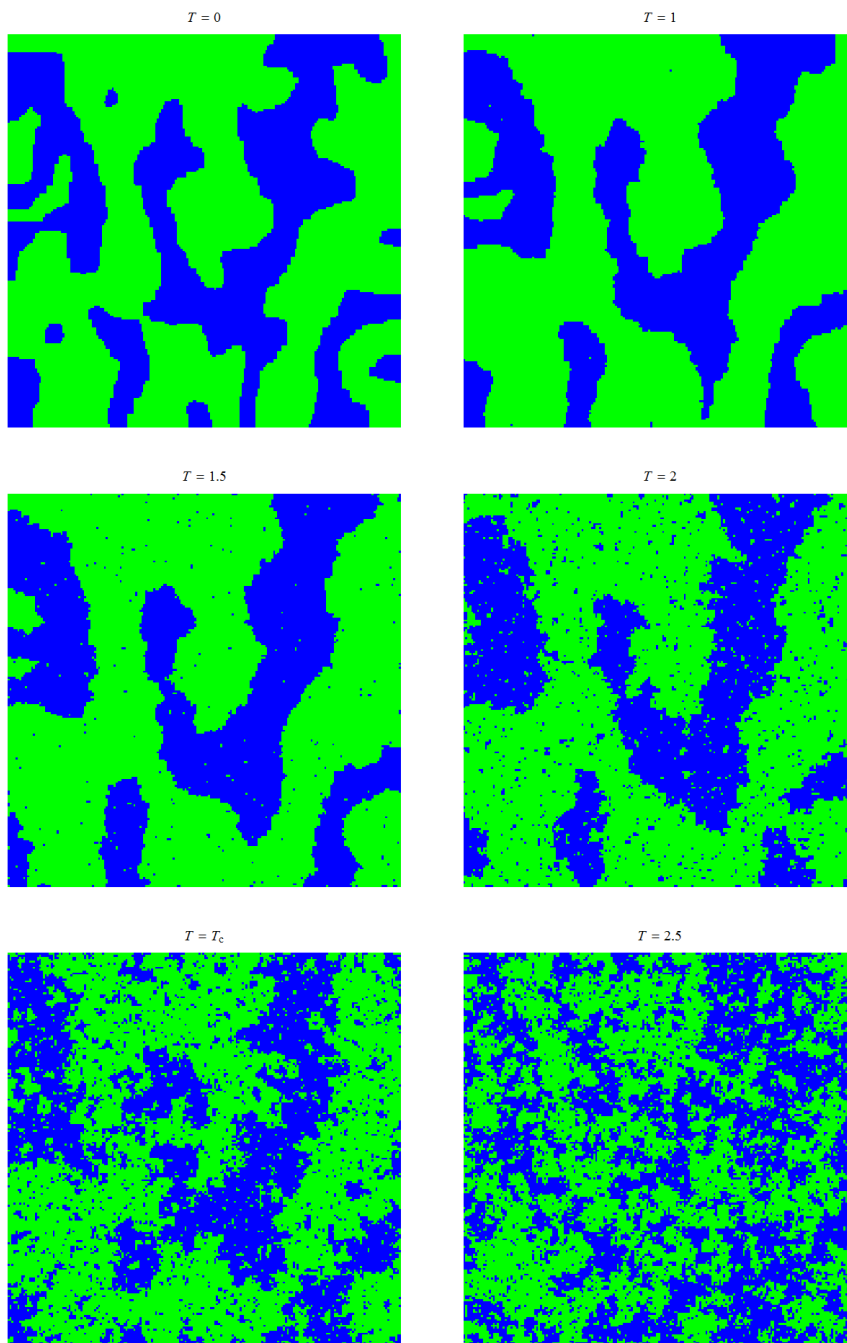
- izberemo naključno mesto na mreži  $(x, y)$
- izračunamo razliko v energiji med prvotno konfiguracijo, in tisto, v kateri je spin na mestu  $(x, y)$  obrnjen, preko formule:

$$\Delta E = 2s_i \left[ J \sum_{\langle i,j \rangle} s_j + H \right] \quad (11)$$

- potezo sprejmemo, če velja  $\Delta E < 0$
- če velja  $\Delta E > 0$ , sprejmemo potezo z verjetnostjo  $e^{-\beta \Delta E}$ , kjer je  $\beta = \frac{1}{k_B T}$ , kjer sem spet brez izgube splošnosti privzel  $k_B = 1$

V orodju MATLAB sem napisal algoritem za simulacijo Isingovega modela. Za občuek sem najprej narisal par mrež pri različnih  $T$  za primer feromagneta z uniformno in naključno inicializacijo ( $J = 1$ ), ter za primer antiferomagneta z naključno inicializacijo ( $J = -1$ ), vse pri odsotnosti polja.

Spodnje slike vsebujejo  $L^2 = 200 \times 200$  veliko kvadratno mrežo pri temperaturah  $T \in [0, 1, 1.5, 2, T_c, 2.5]$  in pri številu korakov  $N = 200L^2$ :

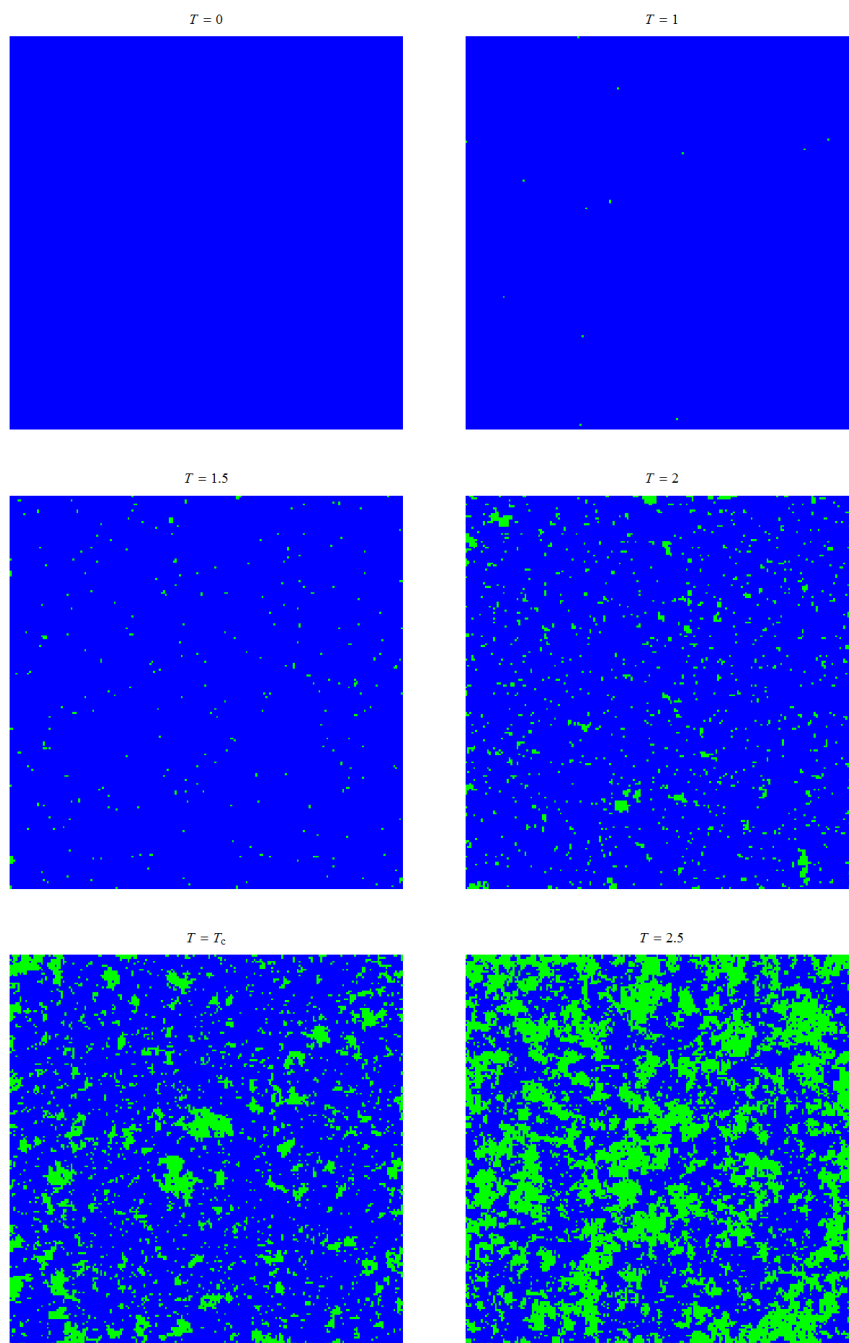


Vidimo, da se na začetku ustvarijo spinske domene in imamo neko spontano magnetizacijo. Ko temperaturo višamo, se domene postopoma kvarijo in premikajo, znotraj pa se začenjajo spini obračati. Simulacija nazorno prikazuje fazni prehod v okolici temperature prehoda. Pri temperaturah  $T > T_c$  vidimo, da se magnetizacija mreže postopoma približuje vrednosti 0. Kratka animacija zgornjega primera je dostopna na spodnji povezavi:

[https://dl.dropboxusercontent.com/u/12747812/Animacije/mat\\_ising.gif](https://dl.dropboxusercontent.com/u/12747812/Animacije/mat_ising.gif)

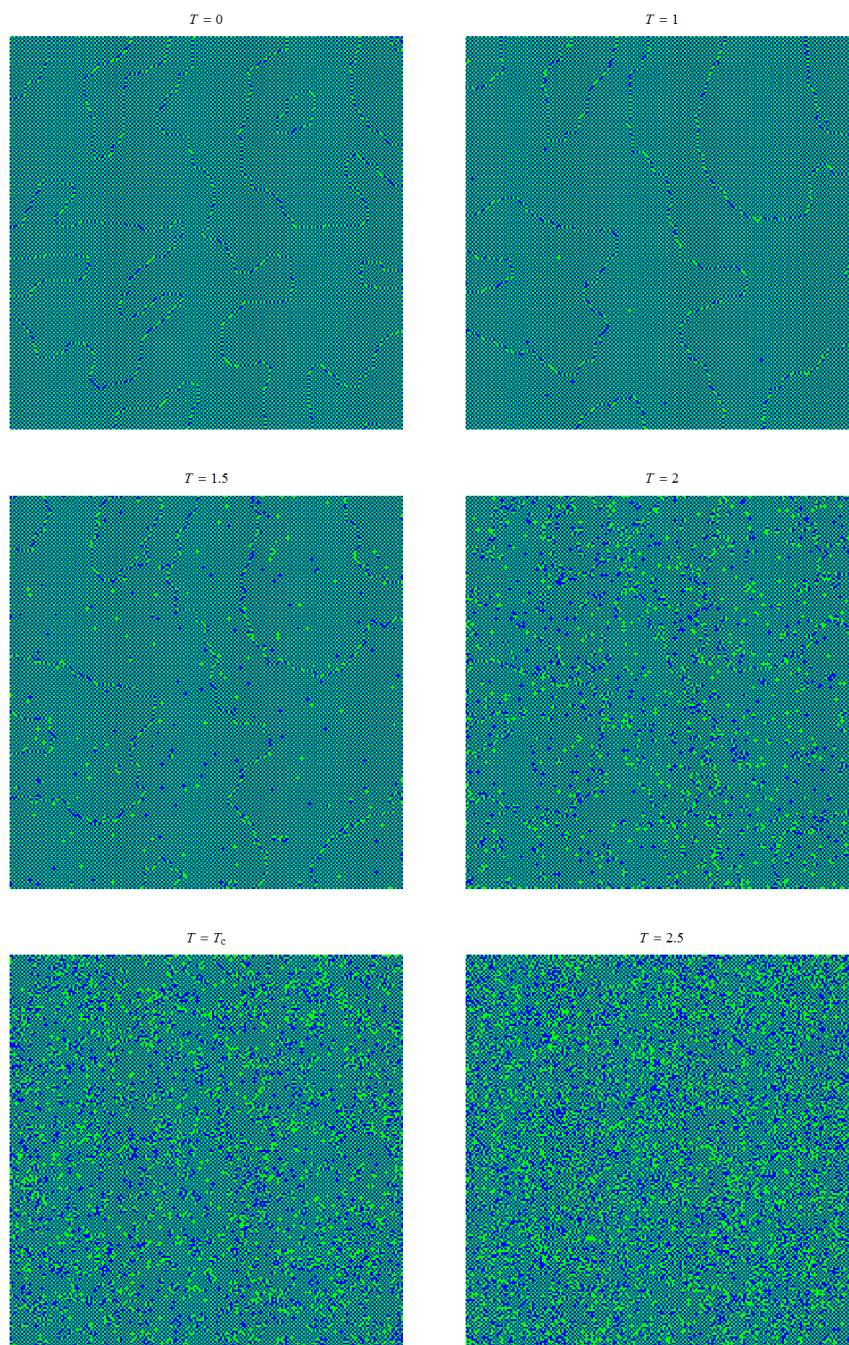


Situacija je drugačna, če je naš začetni pogoj uniformen. Spodnje slike prikazujejo simulacijo pri istih temperaturah, kjer smo na začetku imeli vse spine orientirane v isto smer, kar predstavlja osnovno stanje mreže:



Vidimo, da v tem primeru spini pri  $T = 0$  vztrajajo v tem stanju, v katerega smo jih postavili. Ko višamo temperaturo, se bližamo situaciji iz prejšnjega primera, kjer se spini naključno obračajo. Fazni prehod je zopet očiten pri  $T = T_c$ . Kratka animacija zgornjega primera je dostopna na spodji povezavi: [https://dl.dropboxusercontent.com/u/12747812/Animacije/mat\\_ising2.gif](https://dl.dropboxusercontent.com/u/12747812/Animacije/mat_ising2.gif)

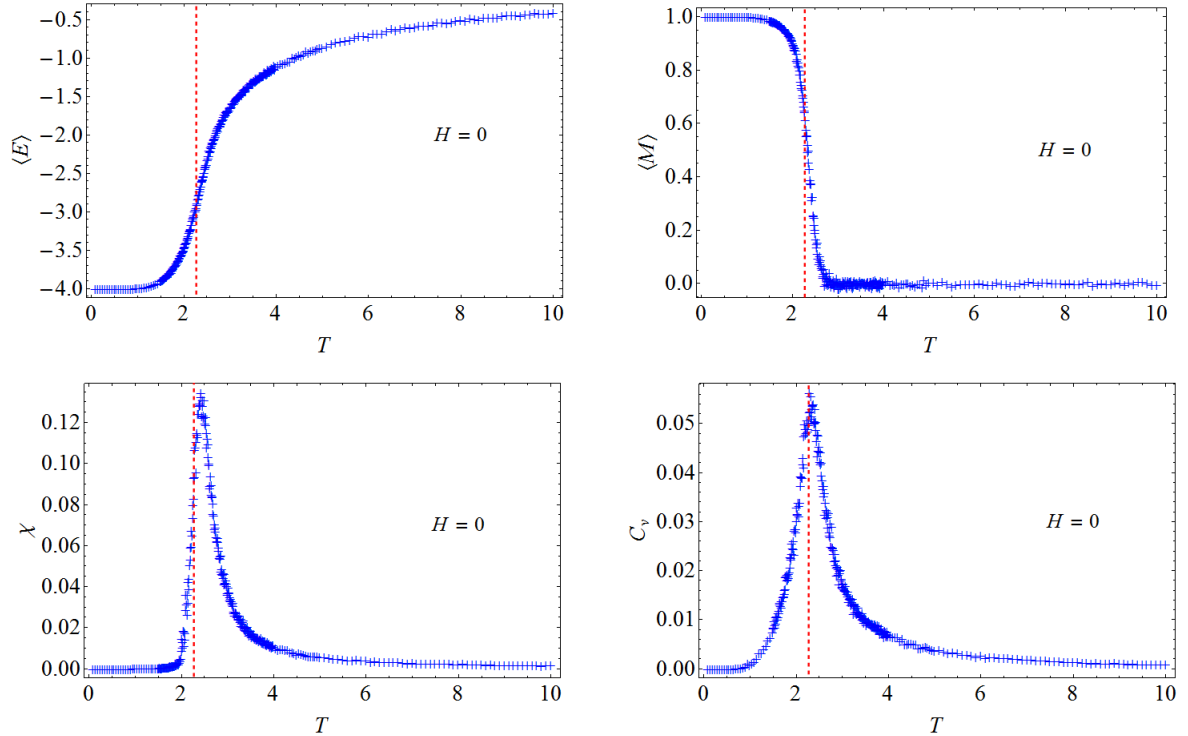
Kot zanimivost si oglejmo še primer, ko je  $J = -1$ , torej model antiferomagneta, pri naključni začetni postavitvi spinov:



Tokrat se domene pojavijo v obliki težko vidljivih mej, ki se premikajo in pačijo na isti način, kot pri feromagnetu z naključno inicializacijo. Kratka animacija zgornjega primera je dostopna na spodji povezavi: [https://dl.dropboxusercontent.com/u/12747812/Animacije/matant\\_ising.gif](https://dl.dropboxusercontent.com/u/12747812/Animacije/matant_ising.gif)

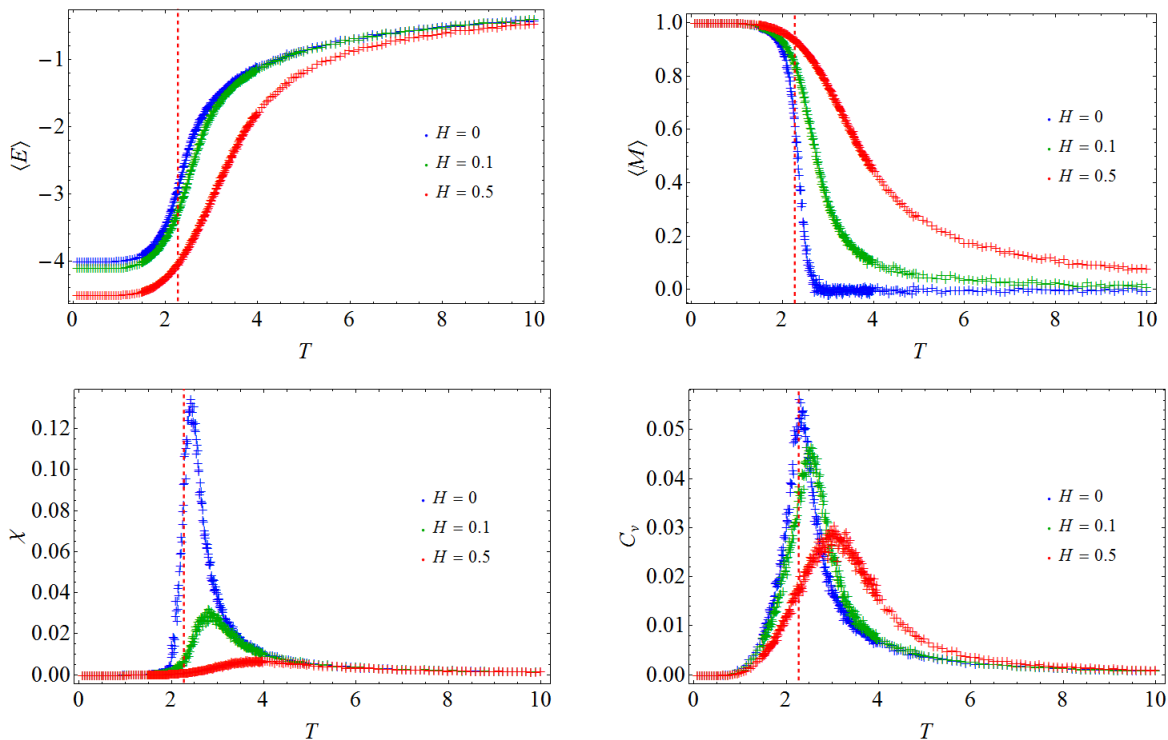
### 3 Rezultati

Preko zgoraj definiranih formul lahko izračunamo relevantne količine. Vse račune sem naredil za primer uniformne začetne postavitve spinov, kjer sem vzel  $10 \times 10$  veliko matriko spinov, saj velikost matrike ni bila tako pomembna kot število korakov in ponovitev. Pri posamezni temperaturi sem naredil  $200L^2$  potez, vsako te poteze pa sem povprečil po 1000 ponovitvah. Temperaturo sem gledal v območju  $T \in [0, 5]$ , območje pa sem razdelil na 100 korakov. Oglejmo si zgoraj definirane količine za primer izključenega magnetnega polja  $H = 0$ :

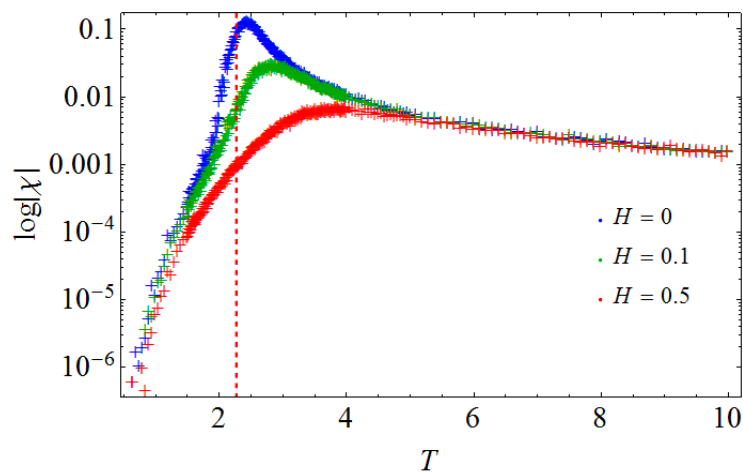


Pri energiji ni potrebo omenjati nič posebnega. Logično je, da energija narašča, saj višamo temperaturo. Opazimo le hiter porast energije v okolici kritične temperature. Pri magnetizaciji vidimo, da imamo na začetku maksimalno možno, ki predstavlja začetno stanje, kjer so vsi enako orientirani. Z višanjem temperature se je magnetizacija manjšala, saj so se spini začeli obračati. Opazimo, da v okolici kritične temperature magnetizacije hitro pade na 0 in tam tudi ostane. Magnetna susceptibilnost in specifična toplota imata obe vrh približno pri kritični temperaturi, kjer tudi dosežeta maksimalno vrednost, potem pa se vrmeta proti vrednosti 0.

Sedaj si lahko pogledamo še primer, kjer imamo magnetno polje vključeno. Teorija za neničelno magnetno polje ne obstaja, zato v tem primeru kritične temperature ne moremo določiti, v kolikor je ta drugačna. Če je drugačna ali ne, pa nam povedo naslednji grafi:



Res opazimo spremembo kritične temperature pri  $H \neq 0$ , saj se v vseh primerih z večjo vrednostjo unanjega magnetnega polja  $H$  k višjim temperaturam prestavi tudi kritična temperatura  $T_c$ . Za bolj nazoren prikaz prilagam še magnetno susceptibilnost v logaritemski skali:



## Del III

# Trgovski potnik

## 1 Naloga

Analiziraj zanesljivost Metropolisovega algoritma na modelu trgovskega potnika, tako da rezultate primerjaš s točnimi rezultati pri majhnem številu mest.

Problem je torej podoben prejšnjim, le da je energija sistema v tem primeru obseg lika, ki ga opiše naključna zbirka točk, oziroma ajkrajša razdalja, ki jo prepotujemo, ko vsako mesto obiščemo le enkrat. V tem primeru se naloga nekoliko rešuje od prejšnjih, saj moramo tu uporabiti t.i. *metodo simuliranega ohlajanja*, kjer je predvsem pomembno to, da začetno s sistemom pri visoki temperaturi, saj tako omogočimo „skakanje“ po celotnem faznem prostoru. Potem sistem postopoma ohlajamo, kjer je začetno stanje pri hladnejši temperaturi vedno končno stanje, ki smo ga dobili pri prejšnjem koraku z višjo temperaturo. Na tak način skušamo z visokimi temperaturami onemogočiti, da bi se rešitev zataknila v lokalnem minimumu, saj je to tu precej verjetno. Pri hladnejših temperaturah nato upamo, da smo se izognili vsem lokalnim minimumom in da se z našimi potezami bližamo globalnemu.

Točno rešitev takšnega problema poiščemo tako, da preverimo prepotovano razdaljo za vse možne kombinacije, katerih je  $n!$ , kjer je  $n$  število mest. Točno rešitev je zelo težko najti že pri  $n > 10$ , kar pa je precej majhno število. Z metodo simuliranega ohlajanja tako zelo poceni pridemo do končnega rezultata. Kako zelo je ta metoda zanesljiva, bomo pa preverili v tem delu naloge.

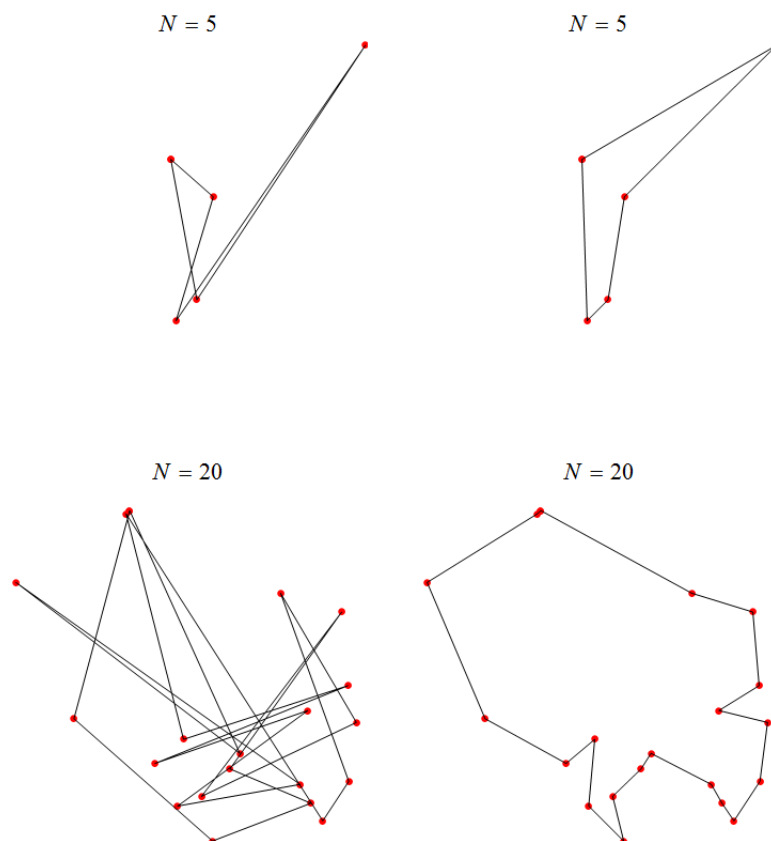
## 2 Reševanje

V Mathematici sem napisal algoritem za reševanje tega problema, kjer sem za vsako temperaturo naredil  $N = 100n$  potez, algoritem pa je potekal nekako tako:

- naključno izžrebam 2 indeksa, ki pripadata dvema različnima mestoma
- indeksa zamenjam, s čimer zamenjam vrstni red obiskanih mest
- preverim razliko prepotovane poti  $\Delta L$
- če je  $\Delta L < 0$  potezo sprejemem, sicer jo sprejemem z verjetnostjo  $p = e^{-\frac{\Delta L}{k_B T}}$ , kjer je  $k_B = 1$
- štejem koliko je takšnih sprejetih potez
- če je število potez večje od 0, zmanjšam temperaturo in ponovim postopek

Postopek sem tako ponavljal, dokler ni bilo pri neki temperaturi več nobene možne uspešne poteze, trenutno konfiguracijo pa sem razglasil za rešitev. Izkazalo se je, da moja metoda deluje v redu za nižje vrednosti  $n$ , za kaj več pa sem uporabil vnaprej definirano funkcijo `FindShortestTour[]`, ki jo vsebuje matematično orodje Mathematica.

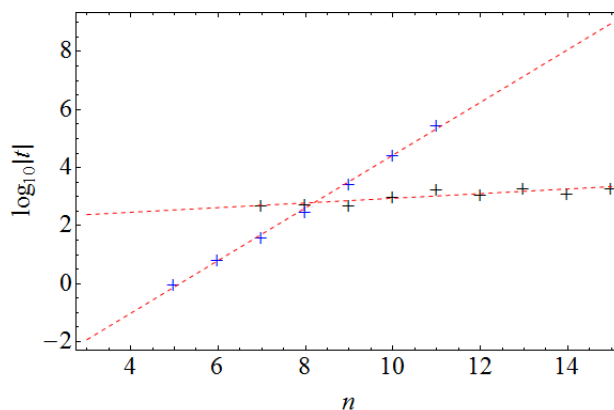
Oglejmo si nekaj možnih postavitev mest s prvo naključno izbrano konfiguracijo in z najbolj optimalno konfiguracijo, ki jo je podala metoda, ki sem jo napisal sam:



Vidimo, da metoda dobro razvozla začetno stanje. Na naslednji povezavi je tudi animacija, ki prikazuje delovanje takšne metode:

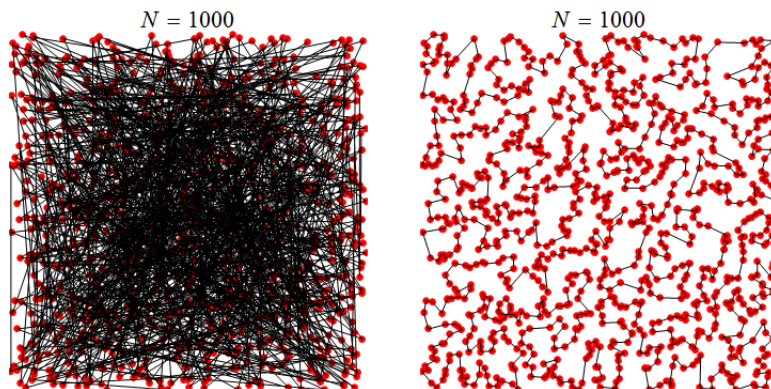
<https://dl.dropboxusercontent.com/u/12747812/Animacije/salesman.gif>

V tem območju lahko rešitve dobimo tudi točno. Poglejmo si, kakšna je časovna zahtevnost obeh metod reševanja:

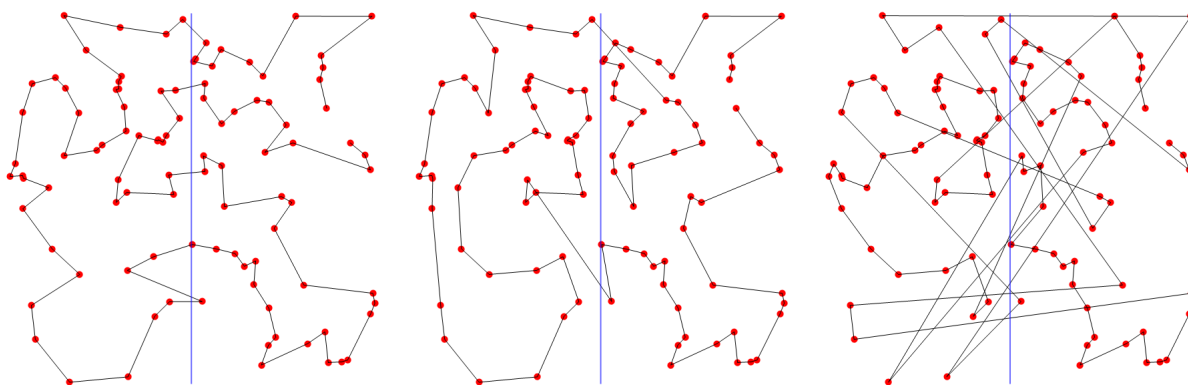


Vidimo da časovna potratnost obeh metod narašča eksponentno ali celo hitreje, vendar je naraščanje porabljenega časa precej hitrejšo pri točni metodi, saj imamo tudi  $n!$  kombinacij, ki jih moramo preveriti. Obstajajo tudi druge metode, poleg metode simuliranega ohlajanja, kot npr. "AllTours", "CCA",

"Greedy", "GreedyCycle", "IntegerLinearProgramming", "OrOpt", "OrZweig", "RemoveCrossings", "Space-FillingCurve" in "TwoOpt" o katerih se nismo učili in jih tudi sicer ne poznam, opazil pa sem, da so precej hitreje tudi od metode simuliranega ohlajanja, kjer slednja za 1000 točk porabi 520 s, privzeta metoda pa le okoli 70 s. S takšnimi metodami lahko zlahka rešimo problem trgovskega potnika tudi za 1000 mest ali več, kjer imajo rešitve naslednjo obliko:



Možno je posamezne poti tudi ustrezno utežiti, s čimer omogočimo, da so nekatere poti bolj zaželeno kot druge, recimo zavijanje v levo, ali prečkanje reke. Oglejmo si primer s 100 mesti, kjer po sredini teče dolga ravna reka. Spodaj so prikazani trije primeri, kjer leva slika ne vključuje restrikcij, sredinska slika kaznuje prečkanje reke, desna slika pa nagrajuje. Sredinska slika tako prikazuje mačko, ki se ne mara zmočiti, desna pa tihotapca, ki tihotapi blago čez reko:



Res opazimo, da leva slika ne prikazuje nobenih očitnih restrikcij na prečkanje čez reko, medtem ko jih sredinska slika očitno kaznuje in desna slika očitno nagrajuje.