

FYS4150 - Computational Physics

PROSJEKT 4

Isingmodellen, Monte Carlo og parallellisering

Kandidat 72

Dato: 7. november 2015

Sammendrag

Vi stifter nærmere bekjentskap med Isingmodellen for et todimensjonalt ferromagnetisk system, Metropolisalgoritmen og parallellisering av C++-kode. Vi ser på et system der spinnene i Isingmodellen kan ta to verdier, opp eller ned, og hvordan dette oppfører seg for forskjellige gitterstørrelser og temperaturer.

Vi bruker resultatene vi får til å estimere den kritiske temperaturen T_C for systemet og sammenlikner det med Onsagers resultat. Vi vil se at Isingmodellen og Metropolisalgoritmen gir en god aproksimasjon til den termodynamiske grensen selv for små gitte.

Prosjektets Git-domene: <https://github.com/jonvegards/FYS4150/tree/master/project4>

Introduksjon

I dette prosjektet skal vi se på den todimensjonale Isingmodellen ved hjelp av numeriske beregninger. Isingmodellen er en modell for et ferromagnetisk system der vi har diskrete variabler, som vi her kaller spinn, organisert i et gitter. Spinnene i systemet kan enten peke opp eller ned, som vi vil representere med henholdsvis 1 og -1 . Hvert spinn i gitteret vekselvirker med de fire næreste naboene.

For å simulere dette systemet på en datamaskin må vi bruke en egnet algoritme, i vårt tilfelle benytter vi Metropolisalgoritmen. Den er en ganske enkel, men kraftig algoritme. Kort sagt går den ut på å sammenlikne en prøvetilstand av systemet med en starttilstand og så akseptere eller avvise den nye tilstanden ved å bruke en akseptering- og avvisningsrutine.

Når vi bruker Metropolisalgoritmen så har vi behov for genererte tilfeldige tall, det kan vi få ved å bruke Monte Carlo-metoder. Vi har laget et program som løkker over alle matriseelementer for hver Monte Carlo-syklus, og for hvert matriseelement som blir løkket over så velger vi et tilfeldig element i matrisa som får spinnets sitt snudd, og deretter utsatt for Metropolistesten.

Til slutt i prosjektet så parallelliserer vi koden slik at vi kan kjøre den for flere temperaturer på en forholdsvis rask måte. Parallelliseringa er gjort på en måte slik at vi beregner simultant for så mange temperaturer som det er kjerner å kjøre på i maskinen vi bruker. Når vi kjører en slik beregning så kan vi plote de forskjellige verdiene som beskriver systemet, som en funksjon av temperatur. Vi vil da kunne finne hvor eventuelle faseoverganger skjer.

Teori

Vi kan bruke termofysikk til å finne egenskapene til systemet vårt, vi er først og fremst interessert i å finne partisjonsfunksjonen siden vi kan finne snittenergi, varmekapasitet, snittmagnetisering, og susceptibilitet når vi kjenner den. Partisjonsfunksjonen er generelt

$$Z = \sum_{i=1}^M e^{-\beta E_i},$$

hvor $\beta = 1/k_B T$, der k_B er Boltzmanns konstant som vi setter lik 1, for resten av dette prosjektet og vi summerer over alle mikrotilstander i i systemet. Energien

er enkel å finne, det er bare å telle opp alle vekselvirkningene spinnene har med hverandre og et eventuelt bidrag fra vekselvirkningene mellom et eksternt magnetfelt B og spinnene i systemet vårt. I vårt tilfelle er det ikke noe eksternt magnetfelt, så vi får,

$$E_i = -J \sum_{\langle j,k \rangle}^L s_j s_k,$$

hvor $\langle j,k \rangle$ betyr at vi summerer over de næreste naboene, *i.e.* spinnene rett over, under og på sidene av spinnet vi ser på. J er "koblingskonstanten" mellom spinnene, vi har $J = 1$. Siden gitteret vårt er endelig, så må vi bestemme oss for hvordan vi skal behandle endene. Vi er interessert i å ha en modell som simulerer virkeligheten best mulig, så vi velger å ha periodiske randbetingelser. Det betyr at vi kobler sammen endene på vårt to-dimensjonale gitter slik at vi effektivt får en torus.

Analytiske resultater For enkelhetsskyld så ser vi på et én-dimensjonalt system for å finne ut hvordan vi skal summere sammen energiene. Vi kan ha f.eks. tre spinn som peker opp,

$$\uparrow_1 \uparrow_2 \uparrow_3,$$

hvor vi da har tre vekselvirkninger, $\uparrow_1 \uparrow_2$, $\uparrow_2 \uparrow_3$ og $\uparrow_1 \uparrow_3$. Vi kan skrive dette enkelt som,

$$E_i = -J \sum_j^L s_j s_{j+1},$$

hvor $s_1 = s_{L+1}$, altså vår randbetingelse. Når vi så skal gjøre dette i to dimensjoner så kan vi tenke oss at den ene summen går radvis i gitteret, mens den andre går kolonnevis, slik at vi får to dobbeltsummer,

$$\begin{aligned} E_i &= -J \sum_j^L \sum_n^L s_{n,j} s_{n,j+1} - J \sum_j^L \sum_n^L s_{n,j} s_{n+1,j} \\ &= -J \sum_j^L \sum_n^L s_{n,j} (s_{n,j+1} + s_{n+1,j}). \end{aligned}$$

der vi har implementert randbetingelsene på samme måte som over. Vi kan nå enkelt finne partisjonsfunksjonen, i det minste for systemer med $L = 2$. Det blir fort mange mikrotilstander å telle over når L øker, men heldigvis for oss klarte Onsager å finne en analytisk løsning for partisjonsfunksjonen når $L \rightarrow \infty$, slik at vi kan sammenlikne vår numeriske løsning og hvordan den vil utvikle seg når vi øker L i beregningene våre.

Det viser seg at energidifferansen fra start- til slutttilstand bare kan ta fem verdier. Dette er en stor fordel! Vi slipper å tenke på alle mulige mikrotilstander, men kan bare se på energidifferansen mellom en foreslått ny tilstand og den gamle. Når vi regner ut energien til systemet, så ser vi på ett og ett spinn og dets vekselvirkning med dets fire naboer. Vi har altså fem spinn totalt. Alle spinnene kan peke opp, og så snur vi det ene, da har vi gått fra å ha en energi på $-4J$ til $4J$, altså $\Delta E = 8J$. Neste tilstand vi ser på er at alle spinn unntatt den ene naboen, peker opp, snur vi "hovedspinnets" da så får vi $\Delta E = 4J$, slik fortsetter det til vi har prekalkulert ΔE for situasjonen der alle spinnene peker ned til å starte med.

Randbetingelsene er også tatt hensyn til i programmet vårt. For å spare FLOPS har vi prekalkulert alle mulige naboer slik at når programmet er inne i Metropolisalgoritmen, så henter den bare ut indeksene til naboene ved å kalle på arrayen vi har laget i stedet for å kalle på funksjonen som regner ut hvor naboen er.

Vi kan nå finne partisjonsfunksjonen for $L = 2$ og deretter “enhetsteste” programmet vårt for å se om det gir riktig resultat. For $L = 2$ har vi $2^4 = 16$ mikrotilstander. De kan vi enkelt tegne opp,

$$\begin{array}{ccc} \uparrow\uparrow & \downarrow\downarrow & 4 \cdot \uparrow\downarrow \\ \uparrow\uparrow & \downarrow\downarrow & \downarrow\downarrow \end{array} \quad (1)$$

$$\begin{array}{ccc} 2 \cdot \uparrow\downarrow & 4 \cdot \downarrow\uparrow & 4 \cdot \uparrow\uparrow \\ \downarrow\uparrow & \uparrow\uparrow & \downarrow\downarrow \end{array} \quad (2)$$

Vi har ganget med degenerasjonsfaktoren for å vise hvor mange mikrotilstander som har samme energi (og antall spinn opp eller ned), vi ser at to mikrotilstander er like når naboene er til de samme spinnene i hver tilstand er like. Vi finner så energiene ved å bruke summasjonsformelen vi kom fram til i stad, vi ser at systemet enten har energi $E = 8J$ (mikrotilstanden med to spinn opp og like nabospinn) eller $E = -8J$ (alle spinn opp eller ned), partisjonsfunksjonen blir da,

$$Z = 2e^{-8\beta J} + 2e^{8\beta J} + 12e^{-\beta \cdot 0} = 4\cosh(8\beta J) + 12.$$

Det er nå enkelt å finne uttrykkene for snittenergi og varmekapasitet, de er definert som

$$\begin{aligned} \bar{E} &= -\frac{\partial \ln Z}{\partial \beta} \\ C_V &= \frac{1}{k_B T^2} \frac{\partial^2}{\partial \beta^2} \ln Z. \end{aligned}$$

Snittmagnetiseringa er simpelthen summen av alle spinnene delt på antall spinn, vi velger å se på absoluttverdien av magnetiseringa siden for små L så vil randbetingelsene medføre noen uregelmessigheter,

$$\langle M \rangle = \frac{1}{ZL^2} \sum_{i=1}^L |m_i| e^{-\beta E_i},$$

hvor vi har $m_i = \sum_{j=1}^{L^2} s_j$ der s_j er spinnets retning. Susceptibiliteten er variansen til magnetiseringa,

$$\chi = \frac{1}{k_B T} \left(\frac{\langle |M|^2 \rangle}{ZL^2} - \frac{\langle |M| \rangle^2}{Z^2 L^4} \right).$$

Vi deler på antall spinn slik at vi kan plotte resultatene fra kjøring med flere spinn i samme plott og sammenlikne. Det er nå ganske rett frem å regne ut de fire størrelsene vi er interessert i.

$$\begin{aligned} \bar{E} &= -\frac{\partial}{\partial \beta} (4\cosh(8\beta J) + 12) \\ &= -\frac{8J \sinh(8\beta J)}{\cosh(8\beta J) + 3}. \end{aligned}$$

Varmekapasiteten blir da,

$$\begin{aligned}
C_V &= \frac{1}{k_B T^2} \frac{\partial^2}{\partial \beta^2} (4 \cosh(8\beta J) + 12) \\
&= \frac{1}{k_B T^2} \frac{\partial}{\partial \beta} (-\bar{E}) \\
&= \frac{1}{k_B T^2} \frac{\partial}{\partial \beta} \frac{8J \sinh(8\beta J)}{\cosh(8\beta J) + 3} \\
&= \frac{1}{k_B T^2} \left(\frac{64J^2 \cosh(8\beta J)}{\cosh(8\beta J) + 3} - \frac{64J^2 \sinh(8\beta J)^2}{(\cosh(8\beta J) + 3)^2} \right) \\
&= \frac{64J^2}{k_B T^2} \left(\frac{\cosh(8\beta J)}{\cosh(8\beta J) + 3} - \frac{\sinh(8\beta J)^2}{(\cosh(8\beta J) + 3)^2} \right).
\end{aligned}$$

Videre har vi gjennomsnittlig magnetisering,

$$\begin{aligned}
\langle |M| \rangle &= \frac{1}{ZL^2} \sum_{i=0}^N |m_i| e^{-\beta E_i} \\
&= \frac{1}{ZL^2} (4e^{8\beta J} + 4 \cdot 2e^0 + 4 \cdot 2e^0 + 4e^{8\beta J}),
\end{aligned}$$

hvor vi da har talt opp én tilstand med alle spinn opp, en med alle spinn ned, fire med to spinn opp og to spinn ned, og to tilstander med ett spinn opp. Til slutt finner vi χ .

$$\begin{aligned}
\chi &= \frac{1}{k_B T} (\langle M^2 \rangle - \langle |M| \rangle^2) \\
&= \frac{1}{k_B T} \frac{1}{ZL^2} (4^2 e^{8\beta J} + 4 \cdot 2^2 e^0 + 4 \cdot 2^2 + 4^2 e^{8\beta J}) \\
&\quad - \frac{1}{k_B T} \left(\frac{1}{ZL^2} (4e^{8\beta J} + 4 \cdot 2e^0 + 4 \cdot 2e^0 + 4e^{8\beta J}) \right)^2 \\
&= \frac{1}{k_B T} \left(\frac{1}{ZL^2} (16e^{8\beta J} + 16) - \left(\frac{1}{ZL^2} (8e^{8\beta J} + 16) \right)^2 \right).
\end{aligned}$$

Disse uttrykkene kan vi nå sammenlikne med våre numeriske resultat, og hvis de numeriske resultatene stemmer overens med disse analytiske, så er det en indikasjon på at programmet vårt fungerer som det skal.

Metropolisalgoritmen og Monte Carlo Nå kan vi se litt nærmere på Metropolisalgoritmen, her så vil vi implementere algoritmen på følgende måte:

- 1: Generer en starttilstand
- 2: Generer en prøvetilstand, *i.e* snu ett spinn.
- 3: Sjekk energidifferansen mellom prøvetilstanden og starttilstanden mot en aksepteringsrutine, vi aksepterer den nye tilstanden hvis $\Delta E < 0$.
- 4: Regn ut $w = e^{-\beta \Delta E}$ og sammenlikne denne verdien med et tilfeldig tall r , hvis $r \leq w$, aksepterer vi den nye tilstanden, hvis ikke så beholder vi den gamle.
- 5: Finn de nye forventningsverdiene.
- 6: Repeter trinn 2-5 til en god nok tilnærming er oppnådd.

En ting som vi har implementert i vår Metropolisalgoritme som egentlig ikke var nødvendig, er punkt 4 over. Hvis $\Delta E < 0$ følger det automatisk at $e^{-\beta\Delta E} > 1$, siden $r \in [0, 1] \Rightarrow r \leq w$. Vi kunne altså spart oss for en if-test og fått et litt raskere program.

Fordelen med Metropolisalgoritmen er at vi ikke trenger å tenke på hvordan vi skal kunne nå alle mulige mikrotilstander i systemet. Vi kan bare regne ut differansen mellom starttilstand og en ny foreslått tilstand, og så bruke den differansen mot en aksepteringsrutine for å bestemme om vi skal ha den eller ei. Det viser seg at energidifferansen mellom en tilstand og en ny kun kan ta fem verdier, disse forhåndskalkulerer vi og setter inn i en array slik at vi sparer noen FLOPS.

For å sjekke når vi har fått en likevektstilstand, så plotter vi hvordan verdiene vi er interessert i, utvikler seg som funksjon av antall MC-sykluser. Vi vil da få en graf som vil flate ut når vi har nådd likevekt. Dette er en ganske ad-hoc måte å sjekke om systemet er i likevekt på, men den er betraktelig enklere enn å regne ut korrelasjonsfunksjoner for å finne når systemet er helt ukorrelert.

Metode

Programmet vi har skrevet, benytter, som allerede nevnt, Metropolisalgoritmen for å finne variansen i energi (varmekapasiteten C_V), gjennomsnittlig energi $\langle E \rangle$, gjennomsnittlig magnetisering $\langle |M| \rangle$ og variansen av magnetiseringa (susceptibilitet χ). Før vi kan se ordentlig på disse verdiene, må vi lage et program som vi vet at virker. Vi ser derfor på 2×2 -tilfellet av et gitter med spinn som kan peke opp eller ned. I forrige seksjon fant vi de analytiske svarene på verdiene vi er interessert i, så vi kan nå enkelt finne ut hvor mange Monte Carlo-sykluser som er nødvendig før vi når et svar som er nøyaktig nok for oss. Vi vil også se at tilfellet med $T = 1$ vil vise seg å være viktig for å sjekke at programmet produserer pålitelige resultater.

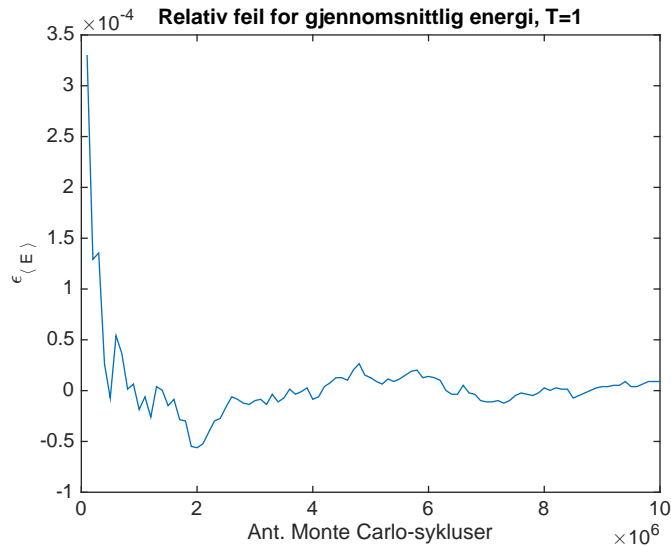
Project4 er bygget opp slik at vi kjører for veldig mange MC-sykluser og plotter deretter resultatene som funksjon av antall sykluser. Når vi har et gitter med kun fire spinn så går utregningene ganske raskt, så det er ikke noe problem å ha et høyt antall sykluser, vi brukte $N_{MC} = 10^7$. Etter at vi har funnet ut hvor mange sykluser som er nødvendig for å nå likevekt, så kan vi øke størrelsen på gitteret og begynne å samle data etter at vi har nådd likevekt.

For å sjekke om starttilstand har noe å si for konvergenshastigheten for systemet, så vil vi også se på hvor raskt de konvergerer fra en starttilstand der alle spinn peker opp kontra en tilfeldig starttilstand, dette gjør vi for temperaturene $T = 1$ og $T = 2.4$. Vi vil da plote gjennomsnittlig energi og magnetisering samt hvor mange aksepterte tilstander som funksjoner av MC-sykluser.

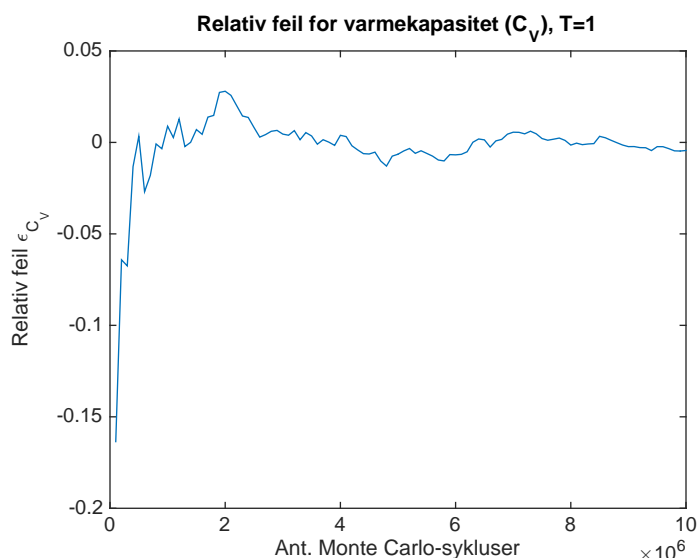
project4_TemperatureRuns kjører for flere temperaturer, vi har da så mange MC-sykluser at termaliseringa av systemet ikke bidrar nevneverdig til det endelige resultatet vårt. Støyen fra termaliseringa blir ytterligere redusert ved at vi gjenbruker tilstandene. Når vi går ett temperatursteg ΔT opp så vil termaliseringa ta kortere tid siden starttilstanden er ganske nære den nye likevektstilstanden for $T_i = T_{i-1} + \Delta T$, det betyr at beregninga for den første temperaturen i hver tråd vil være litt mer unøyaktig enn de påfølgende. Men gir dette utslag på plottene? Følg med i neste seksjon!

Numerisk stabilitet og presisjon

Før vi ser på resultatene, så tar vi en titt på numerisk presisjon målt opp mot analytiske løsninger. I teori-delen fant vi analytiske uttrykk for gjennomsnittlig energi, spesifikk varmekapasitet, gjennomsnittlig magnetisering og susceptibilitet. I figurene (1), (2), (3), og (4) ser vi de relative feilene mellom numerisk og analytisk løsning for et 2×2 -gitter ved $T = 1$. Vi satte programmet til å regne ut verdiene for hver 10^4 Monte Carlo-syklus, men det ser ut til at vi ikke hadde behov for så mange sykluser, vi kunne antakeligvis gått ned flere størrelsesordener og fremdeles hatt et rimelig anslag på systemets verdier, men da måtte vi tenkt mer på termaliseringsprosessen.

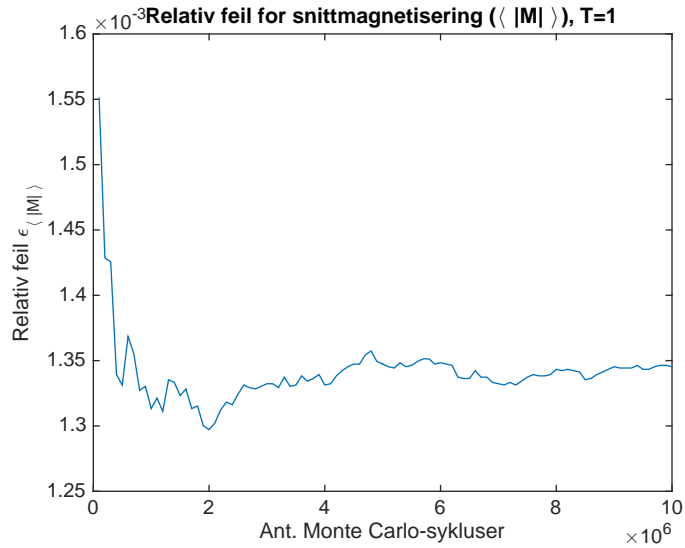


Figur 1: Relativ feil for snittenergi i et 2×2 -gitter. Vi ser at feilen aldri er spesielt stor, den er i størrelsesorden 10^{-4} eller mindre, noe som er akseptabelt for oss. Når systemet har nådd likevekt er feilen cirka i størrelsesorden 10^{-6} , noe som er ganske bra. Programmet ble satt til å regne ut verdiene for hver 10^4 Monte Carlo-syklus, men ut i fra dette plottet ser det ut til at vi kunne gått mye lavere og fremdeles hatt en akseptabel relativ feil.

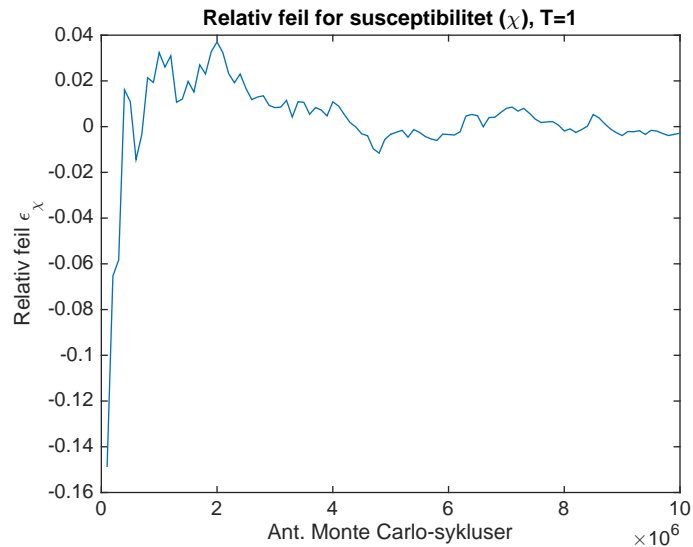


Figur 2: Her ser vi relativ feil for spesifikk varmekapasitet. Programmet bommer mest på denne i starten, og det kan være fordi temperaturen veldig lav. Se tekst for ytterligere detaljer. ($L = 2$)

I fig. (2) så vi at den relative feilen var ganske stor i starten sammenlignet med de andre feilene. Det kommer av at temperaturen er ganske lav og at vi da får $\langle E^2 \rangle$ og $\langle E \rangle^2$ nesten helt like hverandre og noen små feil i programmet kan da gjøre store utslag. En feil vi hadde i programmet befant seg i for-løkken for MC-sykluserne. Vi hadde først satt at den skulle gå fra $n = 0$ til $n \leq N_{MC}$, men da gjorde vi i realiteten $N_{MC} + 1$ MC-sykluser, noe som er greit, men problemet oppstår når vi skal normalisere verdiene i det vi regner ut C_V , siden normaliseringa (i funksjonen `ExpectationValues`) da var ett siffer feil, så fikk det veldig mye å si når T var lav. Vi løste problemet ved å sette løkken til å gå opp til $n < N_{MC}$.



Figur 3: Den relative feilen for snittmagnetisering er liten hele veien, i størrelsesorden 10^{-3} . ($L = 2$)



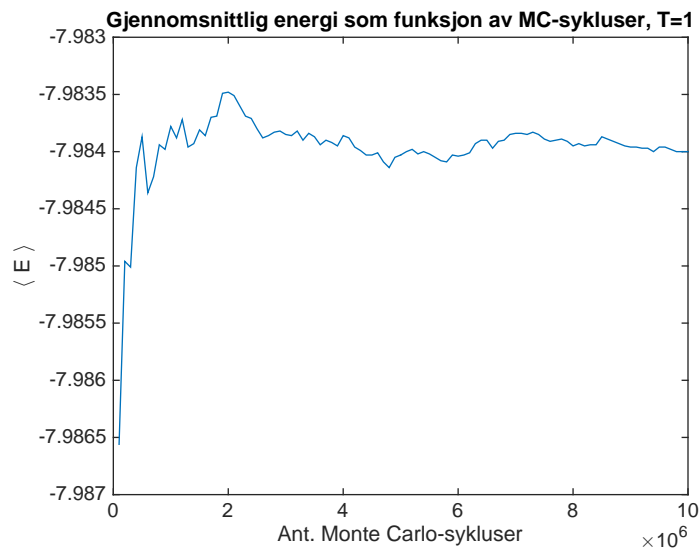
Figur 4: Susceptibiliteten har også en stor relativ feil i starten, men tar seg fort inn og kommer ned i en feil på 10^{-2} ser det ut til. ($L = 2$)

Resultat

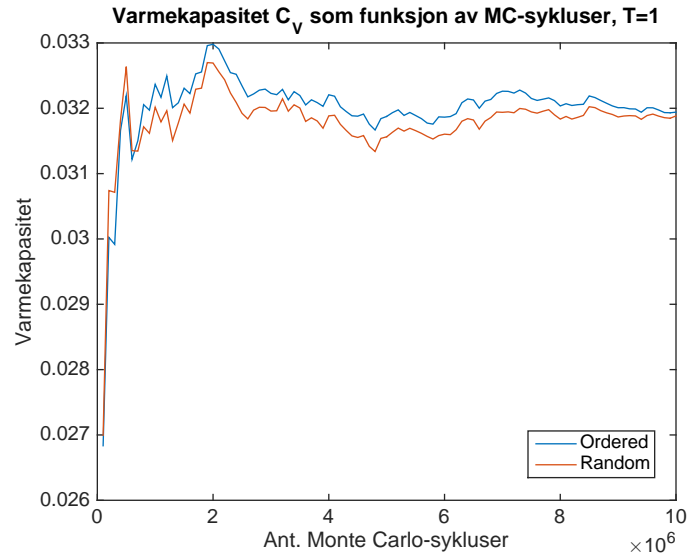
Vi ser først på resultatene fra “enhetstestinga” av programmet. Vi simulerte et gitter med 2×2 spinn for temperatuene $T = 1$ og $T = 2.4$, begge med ordnet og uordnet starttilstander. Resultatene er plottet som verdier per spinn slik at det er enkelt å sammenlikne resultater for flere gitterstørrelser. I figurene (5), (6), og (7) ser vi resultatene av kjøring med 10 millioner MC-sykluser for $T = 1$. Vi ser at

verdiene konvergerer raskt mot en likvektstilstand og at det ikke er nødvendig med 10 millioner MC-sykluser. Vi kan tillate oss å gå helt ned til én million MC-sykluser når vi skal kjøre for flere temperaturer etter hverandre, som nevnt tidligere vil de påfølgende temperaturstegene ha en kort termaliseringsprosess siden vi kommer til å sette temperatursteget til $\Delta T = 0.01$.

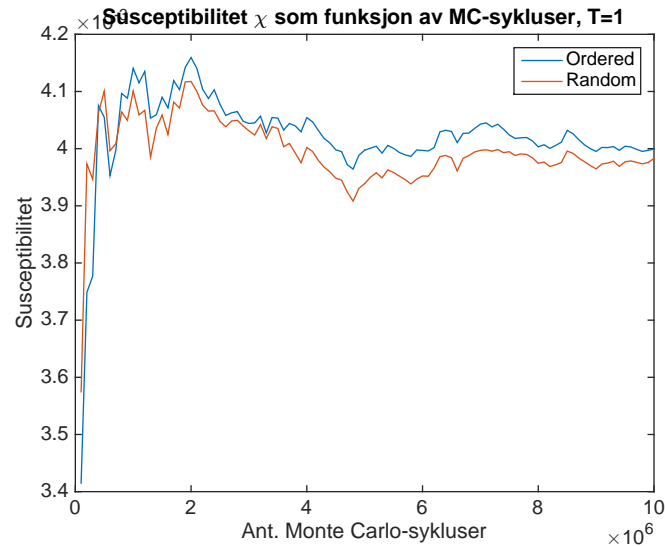
Vi ser også at første verdien som beregnes i alle plottene treffer på første desimal eller bedre, noe som er ganske bra. Det kan se ut som om vi bommer godt på likevektstilstanden i starten, men realiteten er at programmet faktisk treffer ganske bra. La oss se nærmere på fig. (5). Første beregna verdi der er $\langle E \rangle = -7.9866$, mens den analytiske verdien er $\langle E \rangle = -7.9839$, noe som gir en relativ feil på $\epsilon = 0.3296 \cdot 10^{-3}$, som vi tør påstå er ganske bra.



Figur 5: Her er $\langle E \rangle$ plottet som funksjon av antall MC-sykluser for ordnet og uordnet starttilstand. Vi ser at første verdi for snittenergien er ganske bra, den treffer på andre desimal, og tar seg raskt inn til likevektstilstanden. Det vil ikke være noe problem å se bort i fra termaliseringsprosessen så lenge vi kjører en million sykluser. ($L = 2$)



Figur 6: Vi ser at varmekapasiteten blir estimert litt dårligere i starten enn snittenergien ble, slik vi også så i numerisk presisjon-seksjonen ovenfor. ($L = 2$)

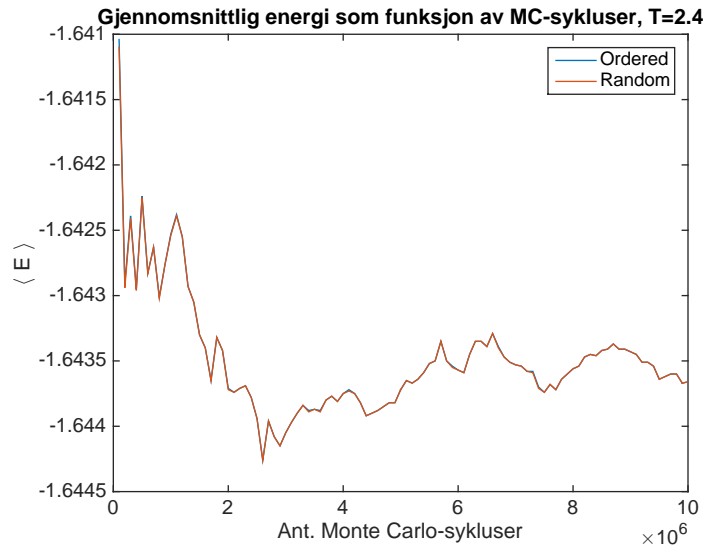


Figur 7: I forrige seksjon fant vi at beregnet verdi for susceptibiliteten bommet litt i starten, men tok seg fint inn, vi ser det samme her også. Vi legger også merke til at starttilstanden har litt å si for hvilken verdi likevektstilstanden tar, den ordnete starttilstanden legger seg på en litt høyere susceptibilitet enn den uordnete. ($L = 2$)

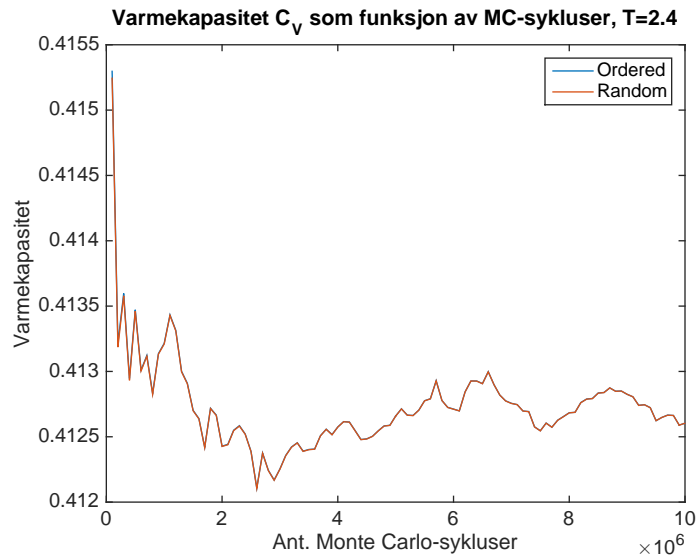
Vi kan så øke temperaturen til $T = 2.4$ og se hvordan forskjellene mellom ordnet uordnet starttilstand er da. I figurene (8), (9), og (10) ser vi resultatene. Vi ser at grafene ligger helt oppå hverandre, i noe de ikke gjorde for $T = 1$.

For $T = 2.4$ har vi ikke regnet ut relativ feil, men vi kan se på hvor man-

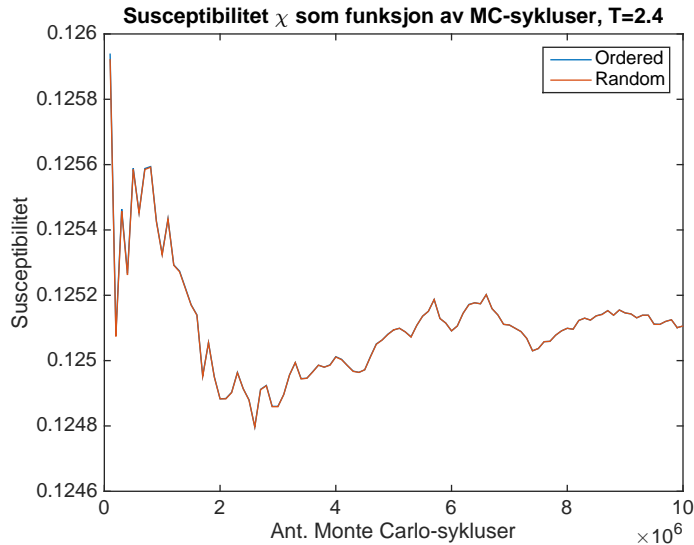
ge desimaler unna første beregnet verdi er fra likevektsverdien. Snittenergien treffer på andre desimal, varmekapasitet på første desimal, og ditto med susceptibilitet. Vi kan også merke oss at for $T = 2.4$ så ser det ut til at systemet gjør litt større svingninger enn for $T = 1$ før det når likevektstilstanden.



Figur 8: Her er $\langle E \rangle$ plottet som funksjon av antall MC-sykluser for ordnet og uordnet starttilstand. Vi ser at første verdi for snittenergien er ganske bra, den treffer på andre desimal, og tar seg raskt inn til likevektstilstanden. ($L = 2$)



Figur 9: Vi ser at varmekapasiteten blir estimert litt dårligere i starten enn snittenergien ble, men standardavviket er $\sigma_{C_V} = 3.6538 \cdot 10^{-4}$. Vi kan altså se bort fra termaliseringa her også. ($L = 2$)

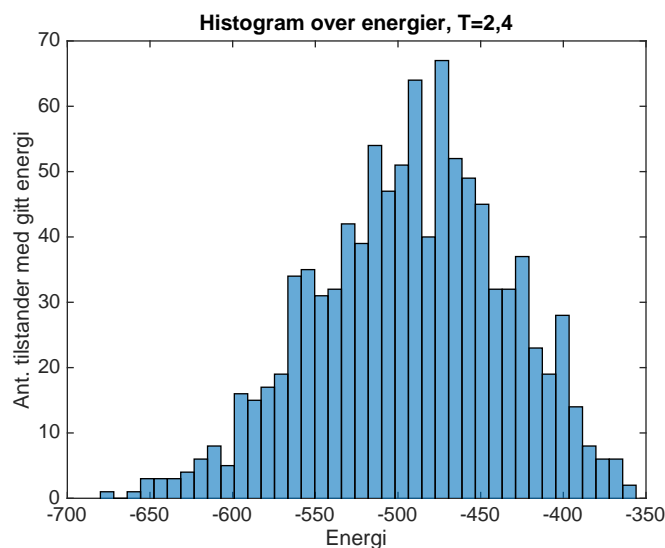


Figur 10: Susceptibiliteten får et estimat som er et stykke unna likevektsverdien, men etter to millioner MC-sykluser er den så å si i likevekt. ($L = 2$)

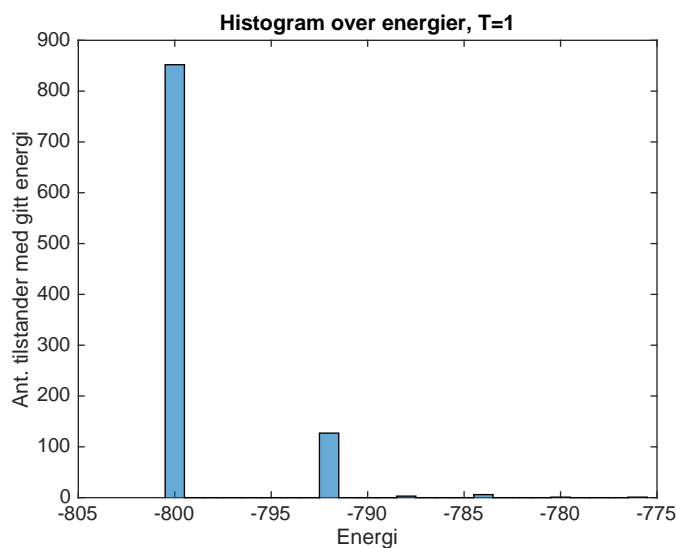
Vi kan så ta en titt på fordelinga av energitilstander. I fig. (11) så ser vi histogrammet over energitilstander etter å ha oppnådd likevekt. Histogrammet senterer seg omkring -500 . Vi har samlet opp data for hver 10000 MC-syklus, men det ville kanskje vært bedre å gjort det oftere slik at vi hadde fått mer data å se på.

I fig. (12) ser vi histogrammet for $T = 1$, ikke overraskende er nesten alle tilstandene i spinn ned, altså de har verdi -1 .

Det som er interessant nå er å sammenlikne variansen til resultatene vi har plottet i nevnte figurer med variansen som vi har regnet ut og sett på tidligere i rapporten, altså C_V . For $T = 1$ er variansen til dataene i fig. (12) 0.0244, mens $C_{V,T=1} = 0.0235$, for $T = 2.4$ har vi fra fig. (11) variansen 1.4010, mens $C_{V,T=2.4} = 1.3955$. Våre utregninger stemmer med andre ord overens, og det er betryggende.



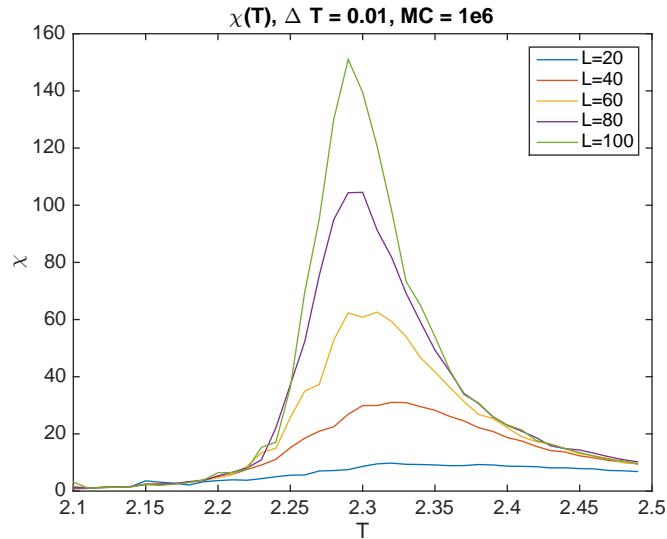
Figur 11: Vi ser her histogrammet som viser oss hvor ofte vi ser tilstander med visse energier for $T = 2.4$. Fordelinga nærmer seg en gaussisk fordeling, for større gitter enn her vil fordelinga nok bli enda glatter. ($L = 20$)



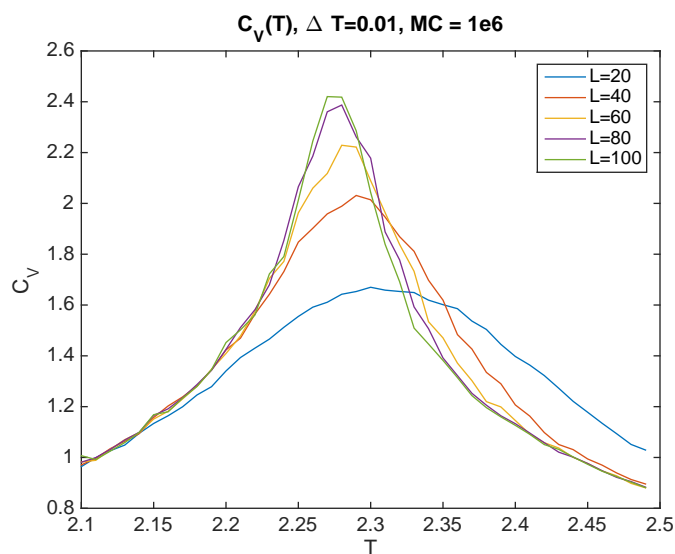
Figur 12: For $T = 1$ ser vi at nesten alle tilstander har alle spinnene sine vendt ned. Dette er tilstanden som gir lavest energi, og siden temperatur og energi er to sider av samme sak, så vil systemet ha den laveste energien. ($L = 20$)

Nå har vi blitt såpass godt kjent med systemet at vi kan bruke det vi vet til å bestemme oss for antall MC-sykluser og temperatursteg slik at vi kan regne ut spesifikk varmekapasitet og susceptibilitet til å estimere en verdi for den kritiske temperaturen T_C . Vi har kjørt med én million MC-sykluser og temperatursteg $\Delta T = 0.01$ for alle gitterstørrelser, $L = [20, 40, 60, 80, 100]$.

Resultatene ser vi i fig. (14) og (13). Vi ser klart og tydelig at noe foregår rundt $T \approx 2.3$. Dette er faseovergangen vi skal se etter. Det er kanskje naturlig å tenke seg at det er lurt å bruke data fra χ_{num} til å finne et estimat for T_C , men det er faktisk litt lurere å bruke $C_{V,num}$ som datagrunnlag. Grunnen til det er at grafen for χ_{num} blir veldig spiss og datapunktene kan derfor sprike litt mye når vi skal finne toppunktene, valget av temperatursteg vil spille en veldig stor rolle og velger man "galt" så kan estimatet bli dårlig. Bruker vi derimot $C_{V,num}$ så vil toppunktet ikke ligge langt over resten av punktene og vi får et mer stabilt resultat.

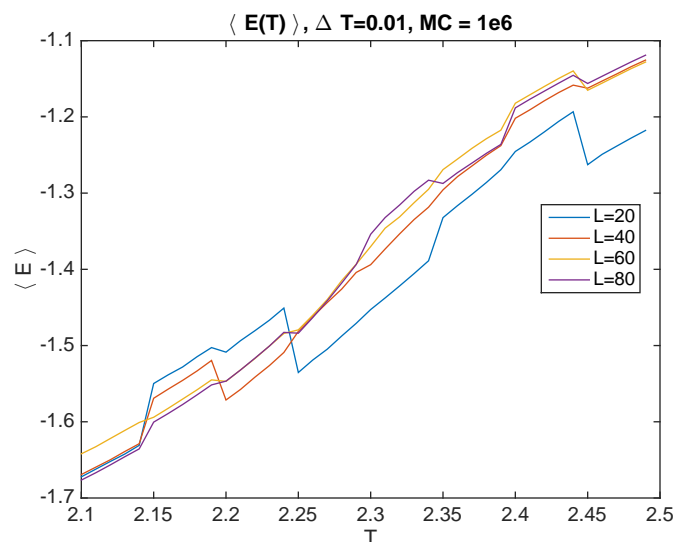


Figur 13: Her har vi plottet susceptibiliteten som funksjon av temperatur for flere gitterstørrelser. Vi ser klart at det skjer noe for $T \approx 2.3$. Dette noe er faseovergangen vi leter etter.



Figur 14: Varmekapasiteten ser ut til å konvergere mot en verdi etter som gitterstørrelsen øker, vi ser også her en indikasjon på at noe skjer i området mellom $T = 2.25$ og $T = 2.30$.

Vi burde ha plottet \bar{E} også, men det ble først glemt vekk og da vi forsøkte å implementere det fikk vi noen numeriske feil som gjorde at resultatet ble det vi ser i fig. (15). På grunn av tidsnød fikk vi ikke tid til å ordne opp i det, men i det minste ser vi at snittenergien som forventet øker med temperaturen.



Figur 15: Vi ser at beregningen av snittenergien ikke gikk helt bra. Dette kan ha å gjøre med parallelliseringa, at det skjer noe galt når vi skal flette sammen arrayene som de forskjellige prosessene har produsert.

For å estimere T_C bruker vi,

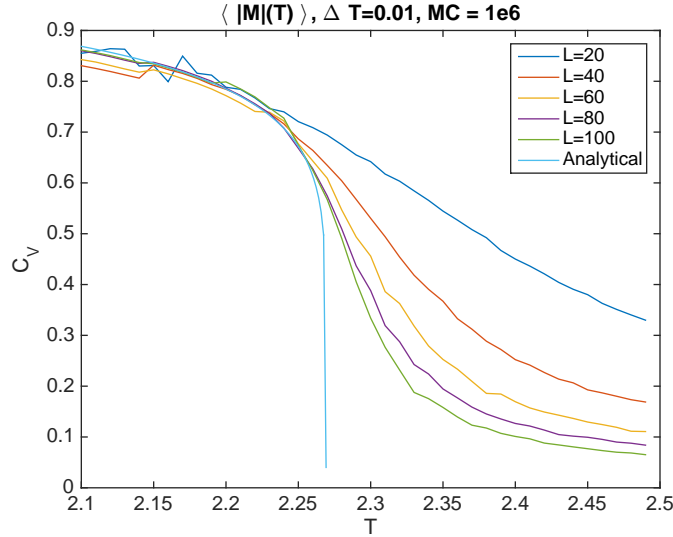
$$T_C(L) - T_C(L \rightarrow \infty) = aL^{-1/\nu},$$

hvor $\nu = 1$, og $T_C(L \rightarrow \infty)$ og a er størrelsene vi skal finne. a er en konstant. Vi løser likningen flere ganger for a på denne måten $a = \frac{T_C(L=100) - T_C(L=40)}{(L=100)^{-1} - (L=40)^{-1}}$, vi tar så snittet av alle verdiene vi regner ut og får $a = 1.17$, noe som gir oss

$$T_C(L \rightarrow \infty) = T_C(L=100) - 1.17 \cdot 100^{-1} = 2.2653,$$

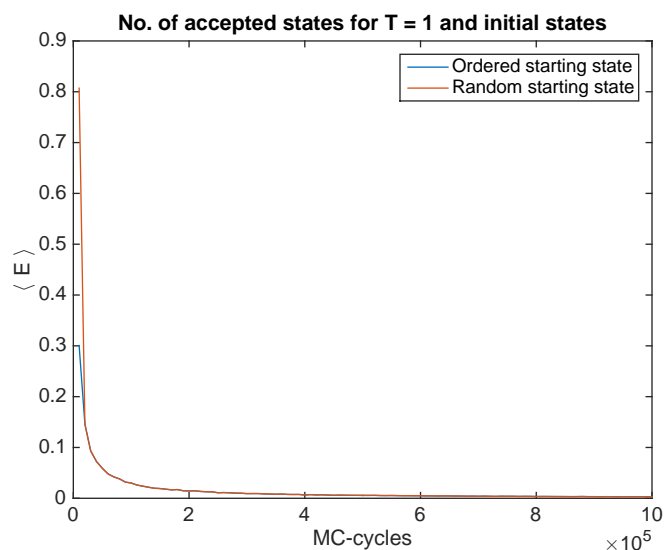
som ikke er så veldig langt fra Onsagers analytiske resultat $T_C \approx 2.269$.

I fig. (16) har vi plottet analytisk og numeriske verdier av snittmagnetiseringa til systemet. Vi ser at Onsagers løsning viser at magnetiseringa blir 0 når vi når den kritiske temperaturen T_C og at våre numeriske beregninger legger seg inntil det analytiske resultatet ettersom gitterstørrelsen øker.

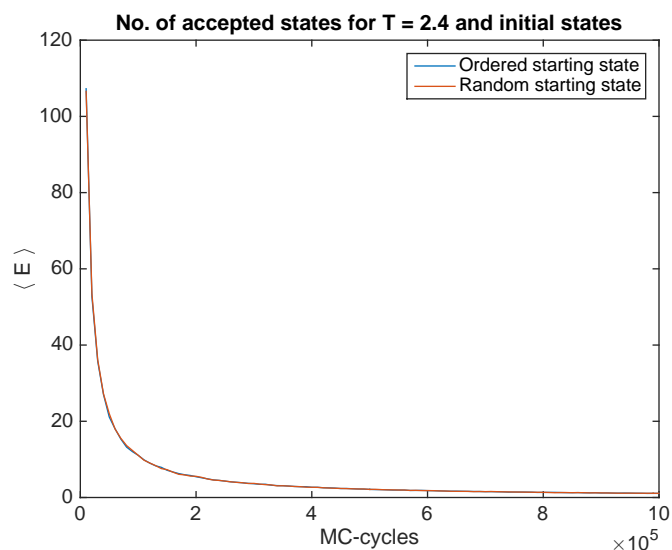


Figur 16: Onsagers analytiske løsning er her plottet sammen med våre numeriske løsninger, vi ser at våre numeriske løsninger konvergerer fint mot den analytiske. Vi ser at over den kritiske temperaturen T_C så har systemet null magnetisering.

Det er også interessant å se på antall aksepterte tilstander som funksjon av MC-sykluser. I fig. (17) og (18) ser vi hvordan systemet aksepterer nye tilstander ettersom syklusene går. Antallet aksepterte tilstander synker drastisk. Vi har plottet resultatet som aksepterte tilstander per antall MC-sykluser, og det er tydelig at for $T = 2.4$ så skjer det mye oftere at nye tilstander blir akseptert enn for $T = 1$. Dette kommer av at sistnevnte helst vil gå til en tilstand der alle spinn peker ned, mens førstnevnte har mer energi og vil gjerne mange spinn som peker opp også. Men begge systemene går mot null nye aksepterte tilstander når likevekt nås, dette skjer fordi systemet har lite å vinne på å akseptere nye tilstander siden det er i en tilstand som har “perfekt” energi i forhold til temperaturen vi har valgt.



Figur 17: Antall aksepterte tilstander er alltid lavt for $T = 1$ og går mot null når likevekt er nådd.



Figur 18: Setter vi $T = 2.4$ aksepterer systemet mange nye tilstander og etter hvert også likevekt slik at antall aksepterte tilstander går mot null. Vi ser at starttilstand har ingenting å si, mens for $T = 1$ så var det en liten forskjell i starten.

Vi har i denne seksjonen sett hvordan en magnet simulert som et 2D gitter oppfører seg for forskjellige temperaturer. Antall MC-sykluser i kombinasjon med gitterstørrelse er kritisk for hvor raskt programmet gjør utregningene, vi prøvde å kjøre en utregning på et 400×400 -gitter med en million MC-sykluser, men det viste seg å ta for lang tid til å ta med i denne rapporten, kjøretiden var på over et døgn. Ser vi på alle figurene der vi har plottet inn to starttilstander,

ordnet og uordnet, så ser vi at det hadde lite å si for hvordan systemet utviklet seg.

Konklusjon

Dette prosjektet har lært oss en del om hvor fort tidsbruken til et program eskalerer når man driver med matriser og Monte Carlo-metoder. Vi fant også at Onsagers resultater ser ut til å stemme i den termodynamiske grensen, det hadde dog vært artig interessant å gjort ordentlige kjøring for større gitter enn vi har gjort her. Kjøringene som er gjort i dette prosjektet kunne også vært forbedret, i noen plott her (f.eks. fig. (15)). Tiden strakk dessverre ikke til for å rette opp på slike ting.

Vi har også sett at Metropolisalgoritmen er en enkel og kraftig algoritme som gir gode resultater selv for veldig små systemer. Et gitter på $L = 80$ er veldig smått hvis vi skal sammenlikne det med en fysisk magnet, den består av millionvis av atomer. Eller hvis vi sammenlikner gitrene våre med den termodynamiske grensen og Onsagers resultat, så er det god grunn til å være imponert over presisjonen til algoritmen vi har brukt.

Kommentarer til prosjektet

Til slutt vil vi nevne at prosjektet har vært morsomt å jobbe med, men har slukt unødvendig mye tid. Programvaren på labmaskinen bør bli testet ut *før* vi starter på prosjektene slik at vi slipper å bruke masse tid på å lete oss fram til en løsning (som viser seg å være veldig enkel) siden vi ikke alltid er så drevne på QT Creator, MPI og C++.