

# Bepaling energie en soortelijke warmte 2D-atoomrooster m.b.v. de Metropolis Monte Carlo methode

Verslag Computational Physics

Sietze van Buuren

Begeleider:  
Prof.Dr. H. de Raedt

29 december 2005

## **Samenvatting**

Gepoogd is de energie en de soortelijke warmte van een 2D atoomrooster, dat voor de helft of een kwart is gevuld met atomen, uit te rekenen. Bij kleine atoomroosters kan dit vrij gemakkelijk handmatig of met de computer exact worden uitgerekend. Bij grote roosters wordt er gebruik gemaakt van de Metropolis Monte Carlo methode (MMC-methode), aangezien het aantal configuraties te groot is voor numerieke optelling.

# Inhoudsopgave

<b>1</b>	<b>Inleiding</b>	<b>3</b>
<b>2</b>	<b>Theorie</b>	<b>3</b>
<b>3</b>	<b>Handmatige bepaling</b>	<b>4</b>
<b>4</b>	<b>Geautomatiseerde numerieke optelling</b>	<b>5</b>
<b>5</b>	<b>Metropolis Monte Carlo methode</b>	<b>7</b>
<b>6</b>	<b>Conclusie</b>	<b>9</b>
<b>7</b>	<b>Dankwoord</b>	<b>9</b>
<b>A</b>	<b>Numerieke optelling</b>	<b>10</b>
<b>B</b>	<b>MMC-methode</b>	<b>11</b>
<b>C</b>	<b>Additionele grafieken</b>	<b>13</b>
	C.1 Numerieke optelling . . . . .	13
	C.2 MMC-methode . . . . .	14

# 1 Inleiding

Bij het vak computational physics krijgt iedere student een opdracht toegewezen m.b.t. tot dit vakgebied. In dit verslag is beschreven hoe van een tweedimensionaal atoomrooster de energie en de soortelijke warmte kunnen worden bepaald als functie van de temperatuur. Bij de exacte bepaling van deze waarden is het nodig om alle roosterconfiguraties te bepalen en door te rekenen.

Voor een rooster van  $2 \times 2$  atomen (dat voor de helft gevuld is met atomen) is het aantal configuraties (6) zo klein dat het mogelijk is om deze met de hand door te rekenen. Voor een rooster van  $4 \times 4$  atomen moet het aantal configuraties (12870) door worden gerekend m.b.v. een computer. Bij een rooster van  $8 \times 8$  atomen is het aantal configuraties ( $1,83 \times 10^{18}$ ) zo groot dat het praktisch onmogelijk is om met de computer al deze combinaties na te rekenen. Hiervoor wordt de MMC-methode gebruikt om de energie en soortelijke warmte te benaderen.

## 2 Theorie

Het rooster wordt als volgt voorgesteld. Het gaat om een tweedimensionaal rooster van  $L \times L$  atomen. We definiëren twee soorten atomen in dit rooster:

- Type A op plaats  $i$ :  $N_i = 1$
- Type B op plaats  $i$ :  $N_i = 0$

Voor het gemak zal worden aangenomen dat type A een atoom is en type B een lege plek.

Elke atoom heeft vier burens: boven, rechts, onder en links. Als een atoom zich op de rand van het rooster bevindt dan is het missende buuratom het atoom aan andere kant van het rooster. Dus

$$N_{i \pm L \hat{e}_x} = N_{i \pm L \hat{e}_y} = N_i \quad (1)$$

De energie van een roosterconfiguratie wordt bepaald door de som te nemen van  $N_i N_j$  over diens buuratomen ( $N_j$ ). Vervolgens wordt er gesommeerd over de atomen in het rooster ( $N_i$ ). De energie van één roosterconfiguratie  $k$  is dus:

$$E_k = J \sum_i \sum_j N_i N_j = J \sum_{i,j} N_i N_j \quad (2)$$

De letter  $J$  houdt verband met de constante van Boltzmann, welke voor het gemak in deze opdracht op één is gesteld.

Om nu de totale energie  $U$  uit te rekenen van een atoomrooster moeten alle  $E_k$  worden bepaald en in gevuld worden in:

$$U = \frac{1}{L^2} \frac{\sum_k E_k e^{-\beta E_k}}{\sum_k e^{-\beta E_k}} \quad (3)$$

Hier is  $\beta$  gedefiniëerd als  $\beta \equiv \frac{1}{T}$ .

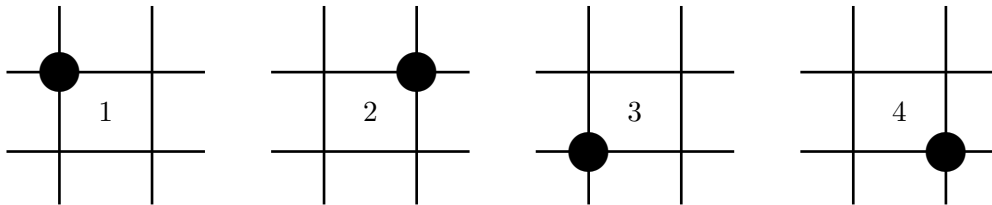
De soortelijke warmte  $C$  wordt bepaald met:

$$C = \frac{\beta^2}{L^2} \left( \frac{\sum_k E_k^2 e^{-\beta E_k}}{\sum_k e^{-\beta E_k}} - L^2 U^2 \right) \quad (4)$$

### 3 Handmatige bepaling

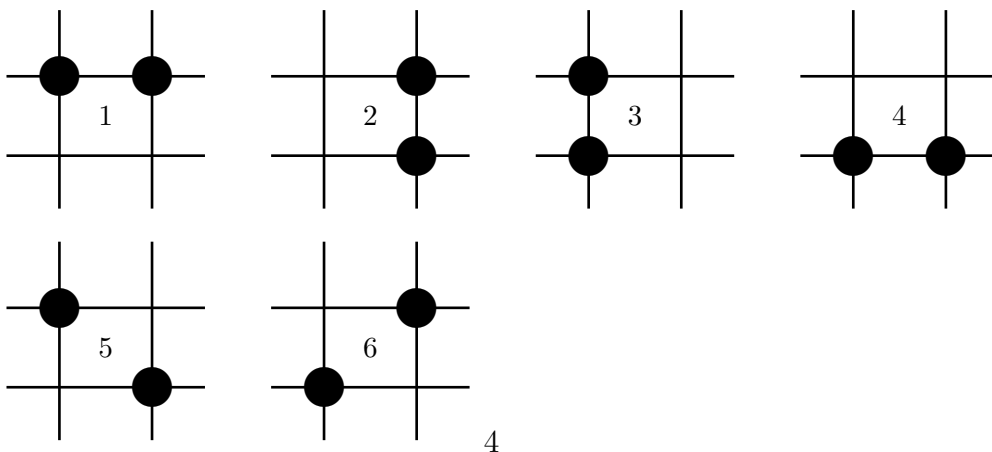
Voor een  $2 \times 2$  rooster kunnen betrekking (3) en (4) met de hand worden bepaald. Net zoals als bij het  $4 \times 4$  en het  $8 \times 8$  rooster zal dit worden gedaan voor  $\sum_i N_i = L^2/n_0$ , waarbij  $n_0 = 2, 4$ .

Bij een  $2 \times 2$  zijn dit respectievelijk 1 en 2 voor  $n_0 = 4$  en  $n_0 = 2$ . Met één plek bezet (dus  $n_0 = 4$ ) kunnen vier combinaties worden gemaakt:



Voor elke configuratie als  $n_0 = 4$  is de energie nul, aangezien het enige atoom nooit een naburig atoom heeft. Daarom is  $U = 0$  en ook  $C = 0$ .

Voor  $n_0 = 2$  is het echter wat interessanter:



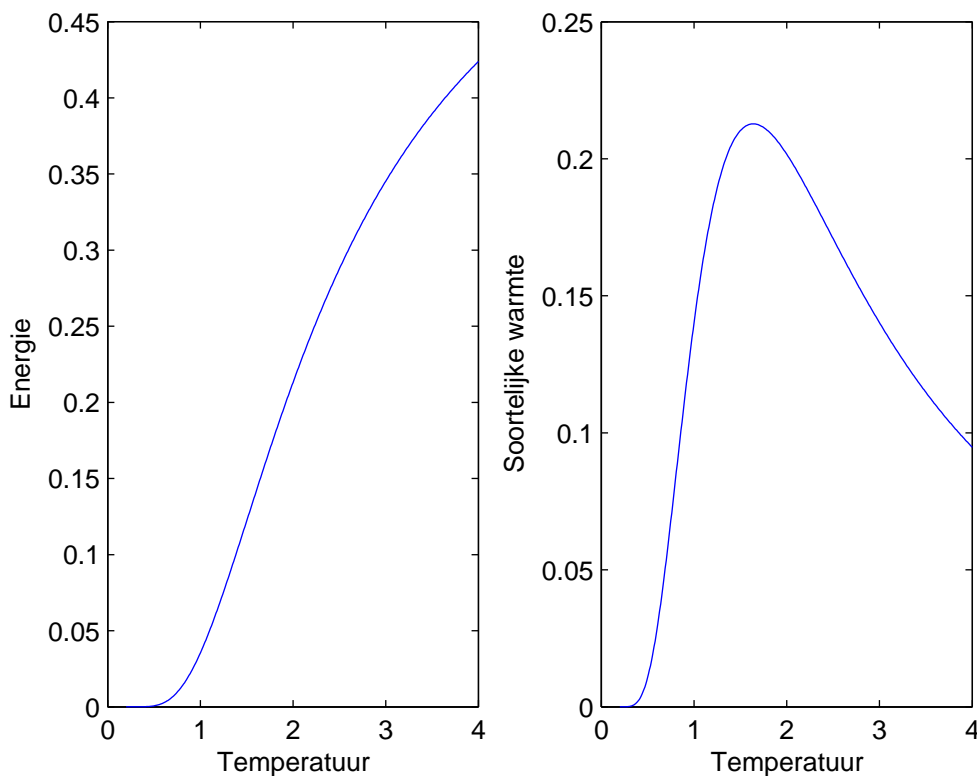
Als we voor elke configuratie (2) uitrekenen, zien we dat  $E_1 = E_2 = E_3 = E_4 = 4J$  en  $E_5 = E_6 = 0$ . Worden deze waarden ingevuld in (3) en (4) dan krijgen we hiervoor respectievelijk de functies:

$$U = \frac{2Je^{-4\beta J}}{2e^{-4\beta J} + 1} \quad (5)$$

en

$$C = \beta^2 \left( \frac{8e^{-4\beta J}}{2e^{-4\beta J} + 1} - \frac{4J^2 e^{-4\beta J}}{(2e^{-4\beta J} + 1)^2} \right) \quad (6)$$

Rekenen we deze waarden uit als functie van de tijd, dan krijgen we de energie en de soortelijke warmte te zien. Zie figuur 1.



Figuur 1: De energie en de soortelijke warmte van een  $2 \times 2$  rooster als  $n_0 = 2$ .

## 4 Geautomatiseerde numerieke optelling

Bij een  $4 \times 4$  rooster is handmatig alle configuraties uitrekenen praktisch onmogelijk. Daarom is hier een computerprogramma voor geschreven. Dit computerprogramma bepaalt eerste alle mogelijke configuraties. Daarna rekt het programma per configuratie (2) uit voor een gegeven temperatuur

en kan zo (3) en (4) bepalen voor deze temperatuur.  
 Het programma is geschreven in Matlab.

Eerste genereert het programma alle mogelijk configuraties die er zijn:

```
k = 1;
for i = 1:2^(L^2)
    N = rem(floor(i * pow2(1-L^2:0)),2);
    if sum(N) == L^2/no
        M(k,:) = N;
        k = k + 1;
    end
end
```

Dan rekent het van elke configuratie de energie (2) uit.

```
for i = 1:L
    for j = 1:L
        U = U + N(i,j) .* (N(mod(i, L) + 1, j) +
            N(mod(i - 2, L) + 1, j) + N(i, mod(j, L) + 1) +
            N(i, mod(j - 2, L) + 1));
    end
end
```

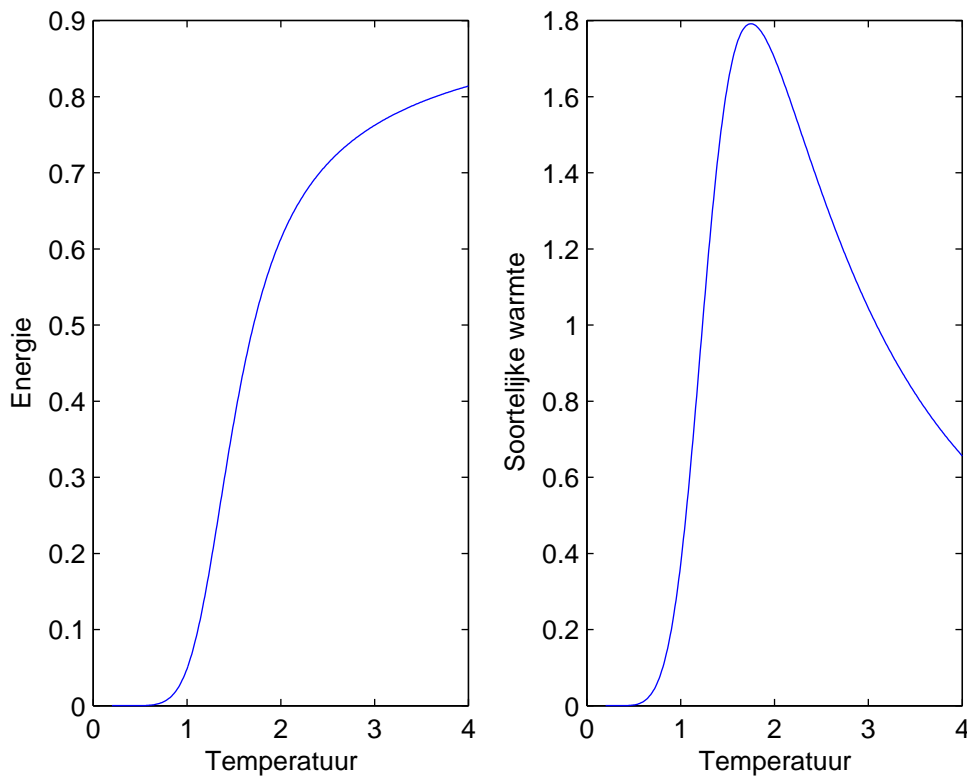
Hierna bepaalt het programma betrekking (3) en (4).

```
for i = 1:combs
    E = energ(M(i,:), L);
    U1 = U1 + E*exp(-E./T);
    U2 = U2 + exp(-E./T);
    U3 = U3 + E.^2*exp(-E./T);
end
U = 1/L^2.*U1./U2;
C = (1./(L*T)).^2.*(U3./U2 - L^2*U.^2);
```

De volledige code van het programma valt te vinden in appendix A.

De energie en soortelijke warmte van een  $4 \times 4$  rooster, en  $n_0 = 2$ , staan in figuur 2. De uitkomst voor  $n_0 = 4$  is terug te vinden in appendix C.

Zoals te zien is, is figuur 2 redelijk analoog aan de gegevens van figuur 1 op pagina 7.



Figuur 2: De energie en de soortelijke warmte van een  $4 \times 4$  rooster als  $n_0 = 2$ .

## 5 Metropolis Monte Carlo methode

Omdat het aantal configuraties bij een  $8 \times 8$  rooster te groot wordt, is de MMC-methode gebruikt om gunstige configuraties te kiezen. Dit gebeurt op de volgende wijze.

Eerst wordt een willekeurige configuratie gegenereerd, waarna deze met een willekeurig kleine gewijzigde configuratie wordt vergeleken. Is de energie van de nieuwe configuratie lager dan de oude, dan wordt de nieuwe configuratie onmiddellijk aangenomen en opgeteld bij de som  $E = \frac{1}{M} \sum_k E_k$  (waarbij  $M$  het aantal gegenereerde configuraties is). Als de energie hoger is dan wordt gekeken of er wordt voldaan aan  $e^{-\Delta E/kT} \geq r$ , waarbij  $r$  een random getal is in het interval  $[0, 1]$ . Is dit zo dan wordt alsnog de nieuwe configuratie meegenomen in de eerder genoemde som, maar als dit niet zo is dan wordt de oude configuratie meegenomen in deze som. Hierna begint het proces opnieuw en wordt het net zo vaak herhaald als is aangegeven.

In Matlab ziet dit er ongeveer zo uit:

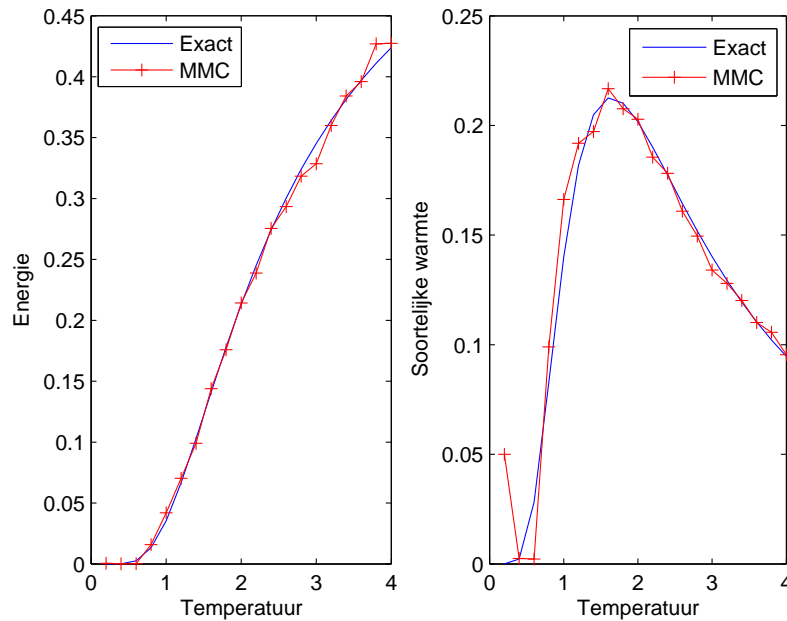
```
for i = 1:p
    s = round((L^2 - 2)*rand(1) + 1);
    g = round(rand(1)*(s - 1) + 1);
```

```

h = round((L^2 - s - 1)*rand(1) + s + 1);
[Nj, Ej] = intenerg(N, E, g, h, L, s);
if Ej < E
    U1 = U1 + Ej;
    U2 = U2 + Ej^2;
    E = Ej;
    N = Nj;
elseif exp(-(Ej - E)/T) >= rand(1)
    U1 = U1 + Ej;
    U2 = U2 + Ej^2;
    E = Ej;
    N = Nj;
else
    U1 = U1 + E;
    U2 = U2 + E^2;
end
end
end
U = U1/(p*L^2);
C = (1./(L*T)).^2.*(U2/p - L.^2.*U.^2);

```

De volledige code van het MMC-algoritme valt terug te vinden in appendix B.

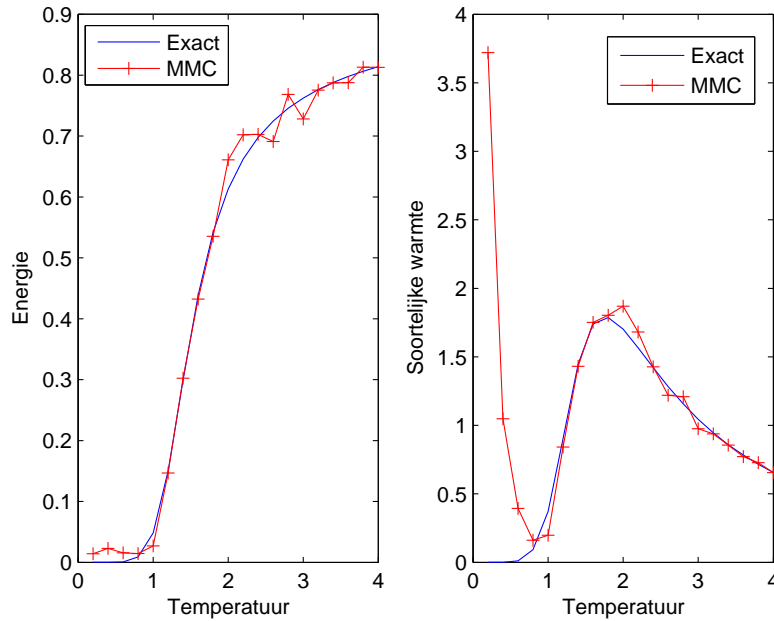


Figuur 3: De MMC-methode vergeleken met de exacte resultaten bij een  $2 \times 2$  rooster als  $n_0 = 2$ .

De resultaten van de MMC-methode worden met eerdere data van het  $2 \times 2$  en het  $4 \times 4$  rooster vergeleken. Dit is te zien in figuur 3 en 4. In beide



figuren is  $n_0 = 2$ , de grafiek voor  $n_0 = 4$  is terug te vinden in appendix C.



Figuur 4: De MMC-methode vergeleken met de exacte resultaten bij een  $4 \times 4$  rooster als  $n_0 = 2$ .

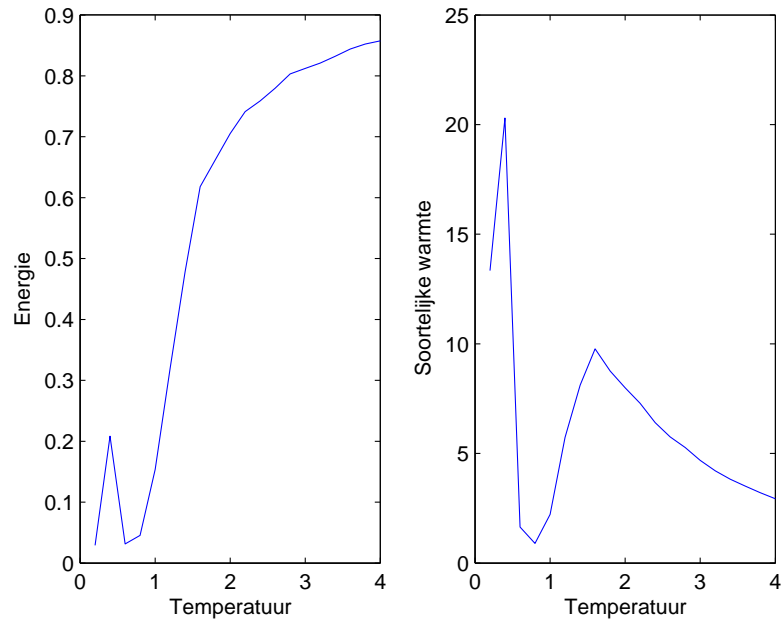
In figuur 3 en 4 benaderen de laagste temperaturen de uitkomsten van de MMC-methode slecht. Dit is altijd het geval en dit heeft waarschijnlijk te maken met de acceptatiefactor  $e^{-\Delta E/kT}$ , welke temperatuurafhankelijk is. Hetzelfde effect doet zich voor bij het  $8 \times 8$  rooster, zoals te zien is in figuur 5 op pagina 10. Voor de overige temperaturen komen de benaderingen in figuur 3 en 4 van de MMC-methode redelijk tot goed overeen met de exacte waarden.

## 6 Conclusie

De conclusie die kan worden getrokken is dat de MMC-methode een goede methode is om de energie en de soortelijk warmte van atoomroosters te bepalen. Echter voor erg lage temperaturen, werkt de methode niet goed.

## 7 Dankwoord

Graag zou ik Prof.Dr. H. De Raedt willen bedanken voor de assistentie bij dit vak.



Figuur 5: De resultaten van de MMC-methode bij een  $8 \times 8$  rooster als  $n_0 = 2$ .

## Appendices

### A Numerieke optelling

```
function [U, C] = enercon(L, no, T)
%-----
% Matlab functie Computational Physics
% Functie voor het exact bepalen van de energie en soortelijke warmte van
% een rooster met een gegeven aantal atomen.
%-----
% U    Energie van alle roosterconfiguraties
% C    Soortelijke warmte van alle roosterconfiguraties
% N    Roostersamenstelling (matrix)
% L    Formaat rooster
% T    Temperatuur
% no   bepaalt hoeveelheid atomen, hoeveelheid = L^2/no, no = 2, 4
%-----

%Initialisatie
U1 = 0; U2 = 0; U3 = 0; k = 1;

%Hoeveelheid configuraties
combs = nchoosek(L^2, L^2/no);

%Genereer matrix (veel te langzaam)
for i = 1:2^(L^2)
    N = rem(floor(i * pow2(1-L^2:0)), 2);
    if sum(N) == L^2/no
        M(k,:) = N;
        k = k + 1;
    end
end
end
```

```

%Reken energien uit
for i = 1:combs
    E = energ(M(i,:), L);
    U1 = U1 + E*exp(-E./T);
    U2 = U2 + exp(-E./T);
    U3 = U3 + E.^2*exp(-E./T);
end

%Bepaling Energie en Soortelijke warmte
U = 1/L^2.*U1./U2;
C = (1./(L*T)).^2.*(U3./U2 - L^2*U.^2);

%end.

function U = energ(N, L)
% -----
% Matlab functie Computational Physics
% Metropolis Monte Carlo: Energie van een 2D-kristalrooster
% Functie voor de energie van n roostercombinatie
% -----
% U    Energie v/e rooster
% N    Roostersamenstelling (matrix)
% L    Formaat rooster
% beta 1/Temperatuur
% -----

% Declaratie
U = 0;
N = sparse(mod([1:L^2]-1,L) + 1, floor(1:L:L+1-1/L), N, L, L);

%Bepaling energie
for i = 1:L
    for j = 1:L
        U = U + N(i,j) .* (N(mod(i, L) + 1, j) + N(mod(i - 2, L) + 1, j) +
            N(i, mod(j, L) + 1) + N(i, mod(j - 2, L) + 1));
    end
end
end

```

## B MMC-methode

```

function [U, C] = moncar(L, no, T, p)
%-----
% Matlab functie Computational Physics
% Metropolis Monte Carlo: Energie van een 2D-kristalrooster
% Functie voor het bepalen van een schatting van de energie en soortelijke
% warmte van een rooster met een gegeven aantal atomen.
%-----
% U    Energie van alle roosterconfiguraties
% C    Soortelijke warmte van alle roosterconfiguraties
% L    Formaat rooster
% no   bepaalt hoeveelheid atomen, hoeveelheid = L^2/no, no = 2, 4
% T    Temperatuur
% p    Aantal iteraties
% s    Snijpunt roosterverdeling
%-----

s = L^2/no;

%Initialisatie
N = floor(((1-1/(L^2)):-1/(L^2):0)/(1-1/no));
E = energ(N, L);
U1 = E;
U2 = E^2;

```

```

%Metropolis Monte Carlo Methode
for i = 1:p
    g = round(rand(1)*(s - 1) + 1);
    h = round((L^2 - s - 1)*rand(1) + s + 1);
    [Nj, Ej] = intenerg(N, E, g, h, L, s);
    if Ej < E
        U1 = U1 + Ej;
        U2 = U2 + Ej^2;
        E = Ej;
        N = Nj;
    elseif exp(-(Ej - E)/T) >= rand(1)
        U1 = U1 + Ej;
        U2 = U2 + Ej^2;
        E = Ej;
        N = Nj;
    else
        U1 = U1 + E;
        U2 = U2 + E^2;
    end
end

U = U1/(p*L^2);
C = (1./(L*T)).^2.*(U2/p - L.^2.*U.^2);

function [Nj, Ej] = intenerg(N, E, i, j, L, s)
%-----
% Matlab functie Computational Physics
% Metropolis Monte Carlo: Energie van een 2D-kristalrooster
% Functie voor het random omwisselen van twee atomen in een atoomrooster.
% Het het bereken van de energie adhv van de vorige configuratie
%-----
% N    Roostersamenstelling (matrix)
% E    Energie rooster N
% Nj   Gewijzigde rooster
% Ej   Energie van het gewijzigde rooster
% L    Lengte array
%-----

%Bepaling trivialiteiten
k = L^2;

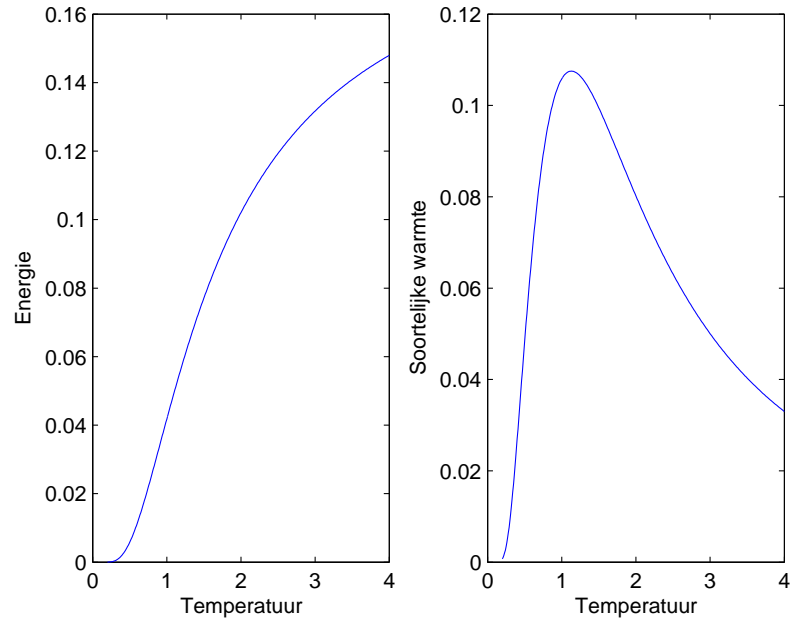
%Omwisseling
Nj = N;
Nj(i) = N(j);
Nj(j) = N(i);

%Bepaling Ej
M = sparse(mod([1:L^2]-1,L) + 1, floor(1:L:L+1-1/L), N, L, L);
Mj = sparse(mod([1:L^2]-1,L) + 1, floor(1:L:L+1-1/L), Nj, L, L);
p = mod(i - 1, L) + 1;
q = floor((i - 1)/L + 1);
v = mod(j - 1, L) + 1;
w = floor((j - 1)/L + 1);
Ej = E + 2*( (Nj(i) - Nj(j))*(M(mod(p, L) + 1, q) + M(mod(p - 2, L) + 1, q) + M(p, mod(q, L) + 1) + M(p, mod(q - 2, L) + 1)) + (Nj(j) - Nj(i))*(Mj(mod(v, L) + 1, w) + Mj(mod(v - 2, L) + 1, w) + Mj(v, mod(w, L) + 1) + Mj(v, mod(w - 2, L) + 1)));

```

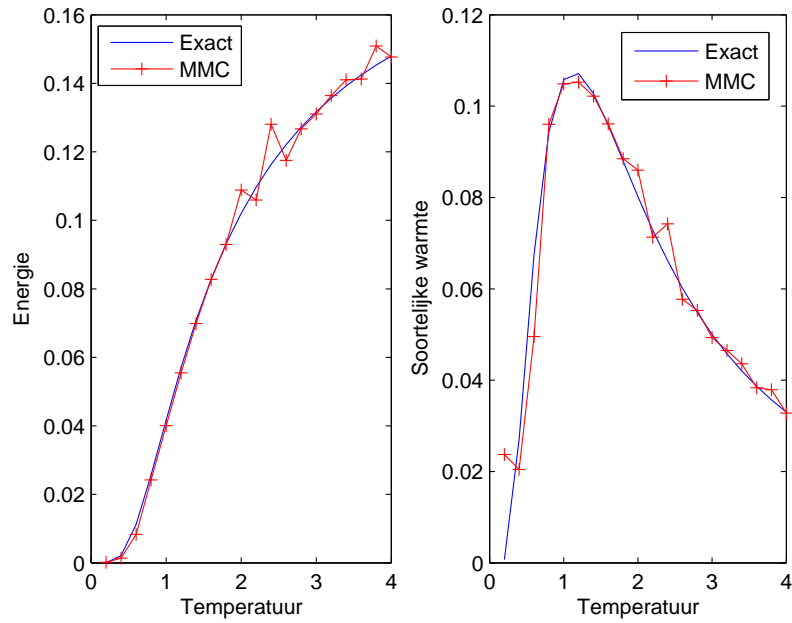
## C Additionele grafieken

### C.1 Numerieke optelling

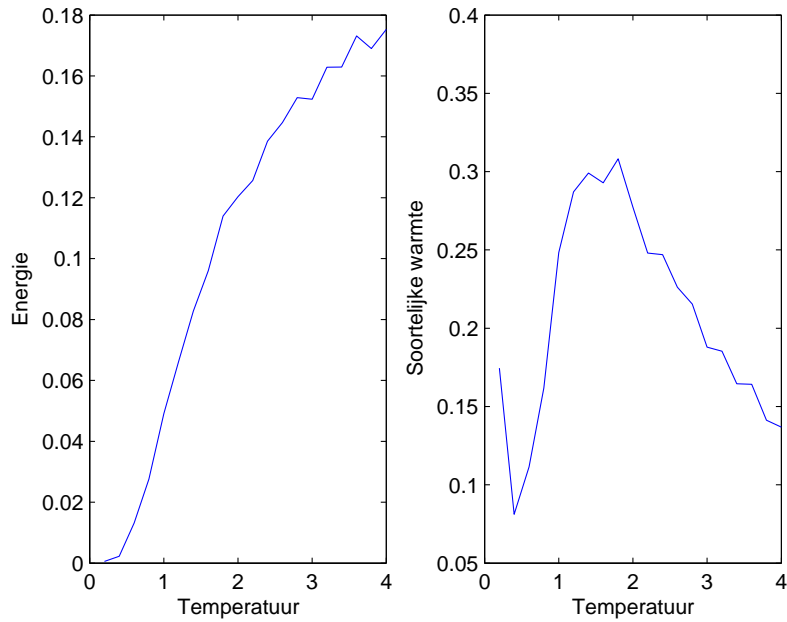


Figuur 6: De energie en de soortelijke warmte van een  $4 \times 4$  rooster als  $n_0 = 4$ .

## C.2 MMC-methode



Figuur 7: De MMC-methode vergeleken met de exacte resultaten bij een  $4 \times 4$  rooster als  $n_0 = 4$ .



Figuur 8: De resultaten van de MMC-methode bij een  $8 \times 8$  rooster als  $n_0 = 4$ .