



## Projeto 5: SIMULAÇÕES DE MONTE CARLO

Anderson Araujo de Oliveira 11371311

# Conteúdo

<b>1</b>	<b>introdução e Teória</b>	<b>3</b>
<b>2</b>	<b>Tarefa-A</b>	<b>4</b>
2.1	Discussões . . . . .	4
2.2	Código . . . . .	8
<b>3</b>	<b>Tarefa-B</b>	<b>10</b>
3.1	B1 . . . . .	10
3.1.1	Discursões . . . . .	10
3.1.2	Código . . . . .	12
3.2	B2 . . . . .	14
3.2.1	Discussões . . . . .	14
3.2.2	Código . . . . .	15
<b>4</b>	<b>Tarefa-C</b>	<b>17</b>
4.1	C1 . . . . .	17
4.1.1	Discussões . . . . .	17
4.1.2	Código . . . . .	21
4.2	C2 . . . . .	23
4.2.1	Discussões . . . . .	23
4.2.2	Código . . . . .	26
<b>5</b>	<b>Tarefa-D</b>	<b>27</b>
5.1	Discussões . . . . .	27
5.2	Código . . . . .	28

# 1 introdução e Teória

O modelo que utilizamos para realizar a simulação de Monte Carlo, em sistemas clássicos de spins, chamamos de modelo de Ising na rede quadrada.

O modelo de Ising é o modelo mais simples de muitos corpos que podemos imaginar. Nele as variáveis podem assumir apenas dois valores, e as interações direta entre as mesmas são apenas entre os vizinhos mais próximos. Apesar desta interação direta ser apenas entre os vizinhos mais próximos, eles interagem com os demais spins, via o efeitos que os vizinhos vão propagando aos seus vizinhos mais próximos, e assim sucessivamente.

As variáveis do sistema  $s(i, j)$ ;  $i = 1, \dots, L_x$ ,  $j = 1, \dots, L_y$ , estão localizadas nos sítios de uma rede retangular com  $L_x$  sítios na horizontal e  $L_y$  sítios na vertical, sendo o número total de sítios  $N = L_x \times L_y$ . Cada variável pode assumir dois valores possíveis, i. e.,  $s(i, j) = +1$  or  $s(i, j) = -1$ .

No modelo de Ising dá pesos diferentes para cada configuração. Ele quantifica esta energia dizendo que a energia de uma dada configuração seria dada por.

$$E = -\frac{J}{2} \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} s(i, j) [s(i-1, j) + s(i+1, j) + s(i, j-1) + s(i, j+1)] \quad (1)$$

Sendo  $J$  uma escala de energia.

Se pensarmos nas variáveis  $s(i, j)$  como sendo os valores de spins (relacionados a momentos magnéticos do sistema), então se tal sistema estiver no zero absoluto devemos esperar todo o sistema ordenado (todos  $+$  ou todos  $-$ ). Mas se o sistema estiver em uma temperatura absoluta  $T$ , esperamos que as flutuações térmicas permitam que configurações com maior energia possam ter uma probabilidade não nula de ocorrência. De fato, conforme vocês já devem ter visto, ou se não viram, verão, a probabilidade  $P(E)$  dos spins estarem em uma dada configuração de energia  $E$  será proporcional ao chamado fator de Boltzmann :

$$P(E) \approx e^{-\beta E} \quad (2)$$

Sendo  $\beta = \frac{1}{KT}$ , e  $K$  é a constante de Boltzmann. Para  $KT \ll J$ , onde a temperatura está baixa no sistema, a probabilidade de spin alterar no seu sistema é baixa ocorrência, para altas temperatura onde  $KT \gg J$ , a probabilidade de tal configuração é alta. O sistema estará em configurações típicas que dependerão da relação entre estas duas escalas que competem: a escala  $J$ , que tenta privilegiar as configurações que os spins estariam ordenados, e a escala  $KT$  que tenta desordenar o sistema. Uma grandeza que quantifica esta ordenação do sistema é a magnetização do sistema por spin. A magnetização, por spin, de uma dada configuração, é dada por:

$$m = \frac{M}{N} = \frac{1}{N} \sum_{i=1}^{L_x} \sum_{j=1}^{L_y} s(i, j) \quad (3)$$

Assim para a temperatura nula, onde o sistema estaria em apenas uma configuração totalmente ordenada teríamos  $m = 1$  ou  $m = -1$ , enquanto para que  $T \rightarrow \infty$ , todas as configurações teriam iguais chances e a magnetização dependeria da configuração que o sistema estivesse.

A probabilidade de um spin mudar é dada pela equação (2), a energia nesse sistema é  $E_i$ , se o spin mudar quando for escolhido a nova temperatura é  $s \rightarrow -s$ , a nova energia seria  $E_f = E_i - s(i, j)[s(i-1, j) + s(i+1, j) + s(i, j-1) + s(i, j+1)]$ . A probabilidade do spin não mudar, fica com a mesma energia  $E_i$  e  $P(E_i) \approx e^{-\beta E_i}$  e do sistema mudar seria  $P(E_f) \approx e^{-\beta E_f}$ , normalizando as probabilidades.

$$\begin{aligned} P(E_f) &= \frac{e^{-\beta E_f}}{e^{-\beta E_f} + e^{-\beta E_i}} \\ P(E_i) &= \frac{e^{-\beta E_i}}{e^{-\beta E_f} + e^{-\beta E_i}} \end{aligned} \quad (4)$$

Assim temos que a soma das probabilidades é 1, como deveria ser. Mas reparemos que  $E_i$  e  $E_f$  são dados por:

$$\begin{aligned} E_f &= E_0 - 2Js(i, j)[s(i-1, j) + s(i+1, j) + s(i, j-1) + s(i, j+1)] \\ E_i &= E_0 - Js(i, j)[s(i-1, j) + s(i+1, j) + s(i, j-1) + s(i, j+1)] \end{aligned} \quad (5)$$

Como  $E_0$  não envolve o spin  $s(i, j)$ . Se substituirmos a expressão acima em (4), temos que a probabilidade  $P(s)$  que o spin em  $s = s(i, j)$  permaneça com o seu valor  $s$ , e probabilidade  $P(-s)$  que ele mude serão dadas por:

$$\begin{aligned} P(s) &= \frac{e^{-\beta \Delta M}}{e^{\beta \Delta M} + e^{-\beta \Delta M}}, \quad P(-s) = \frac{e^{-\beta \Delta M}}{e^{\beta \Delta M} + e^{-\beta \Delta M}} \\ \Delta M &= Js(i, j)[s(i-1, j) + s(i+1, j) + s(i, j-1) + s(i, j+1)] \end{aligned} \quad (6)$$

Assim para fazermos uma simulação de Monte Carlo, em uma dada temperatura, usando a dinâmica acima, iniciamos com uma configuração inicial e vamos medindo por exemplo a magnetização do sistema, até que o sistema atinge o equilíbrio, quando a magnetização fica flutuando em torno de seu valor de equilíbrio. A unidade de tempo de Monte Carlo, que definimos, é aquela em que  $N$  spins são testados aleatoriamente.

Agora que temos conhecimento da teoria, podemos discutir os resultados obtidos nos programas.

## 2 Tarefa-A

### 2.1 Discussões

Nessa tarefa vamos verificar o magnetismo do sistema em duas temperaturas diferentes, quando os spins estão ordenados (todos estão com mesmo sinal). A temperatura que vamos verificar é para  $\beta=3$  e  $\beta=0.1$ . Vamos testar essas temperaturas para sistemas de tamanho diferente  $L=60, 100$ .

Para  $L=60$  e  $\beta=3$ . Magnetização e a ultima configuração.

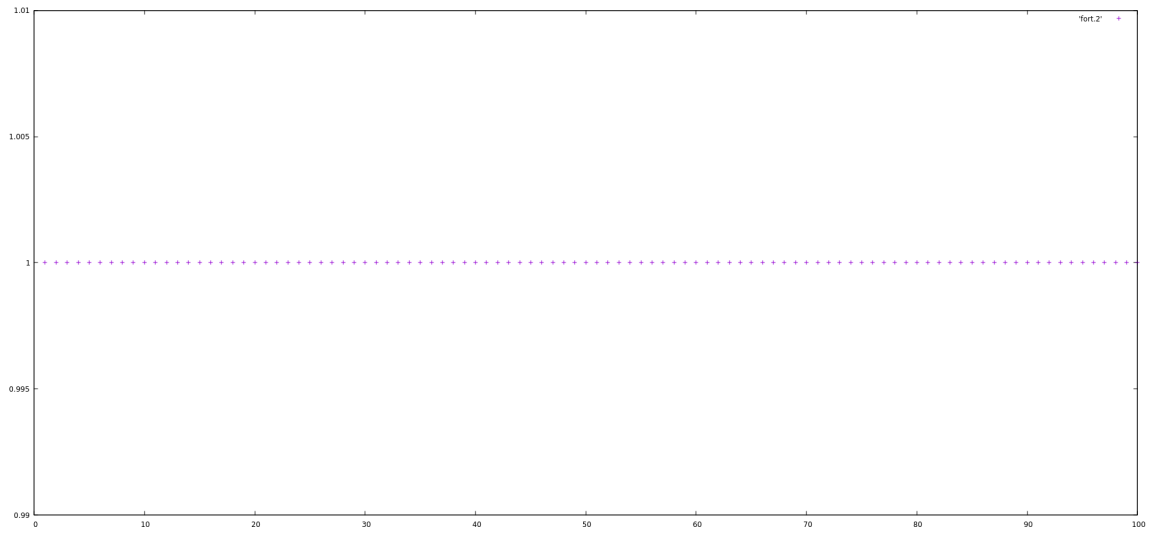


Figura 1: Magnetização versus interações

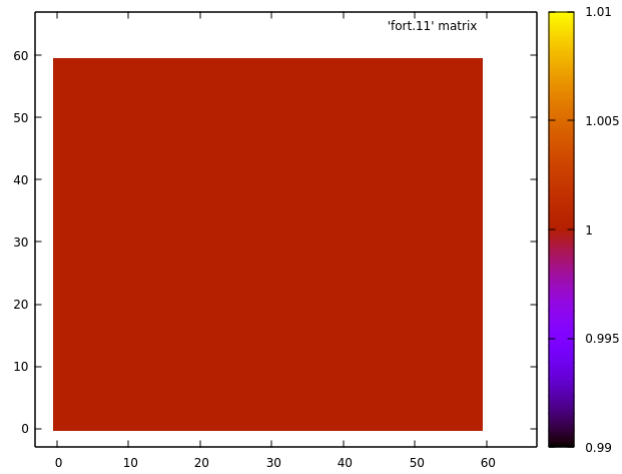


Figura 2: Organização dos spins, na ultima configuração

Para  $L=60$  e  $\beta=0.1$ . Magnetização e a ultima configuração.

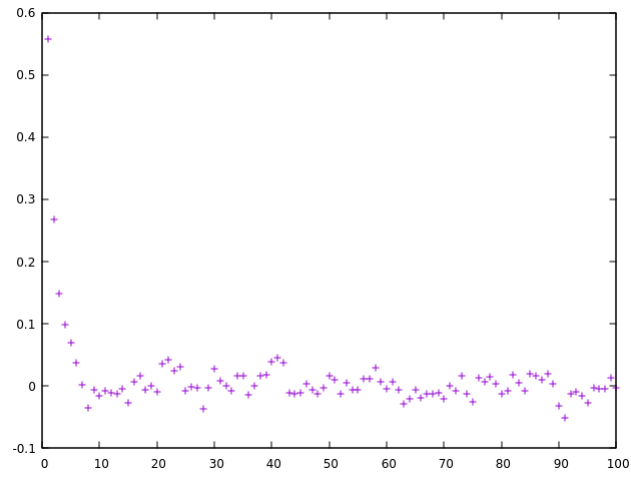


Figura 3: Magnetização versus interações

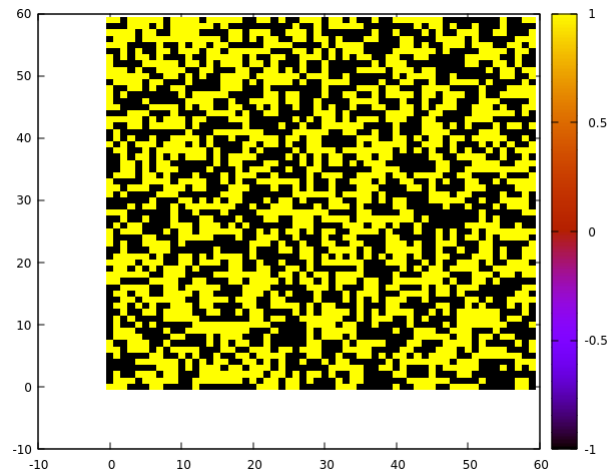


Figura 4: Organização dos spins, na ultima configuração

Para  $L=100$  e  $\beta=3$ . Magnetização e a ultima configuração.

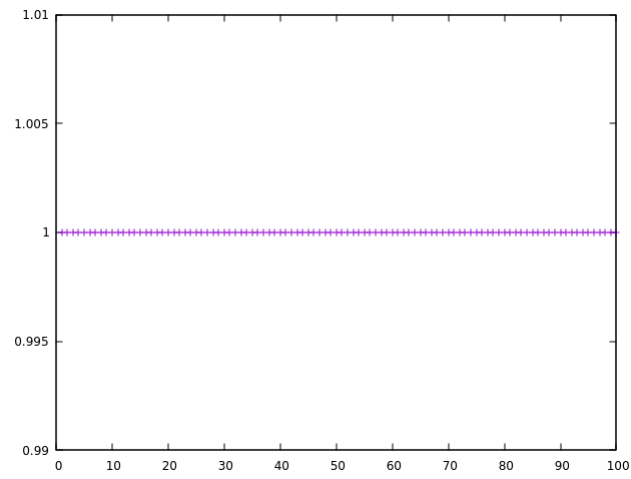


Figura 5: Magnetização versus interações

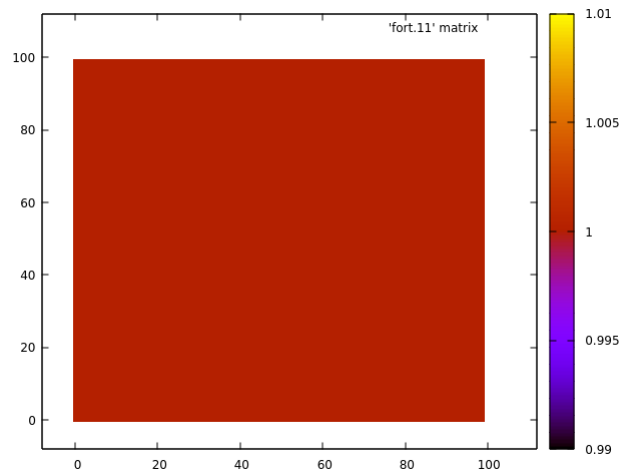


Figura 6: Organização dos spins, na ultima configuração

Para  $L=100$  e  $\beta=0.1$ . Magnetização e a ultima configuração.

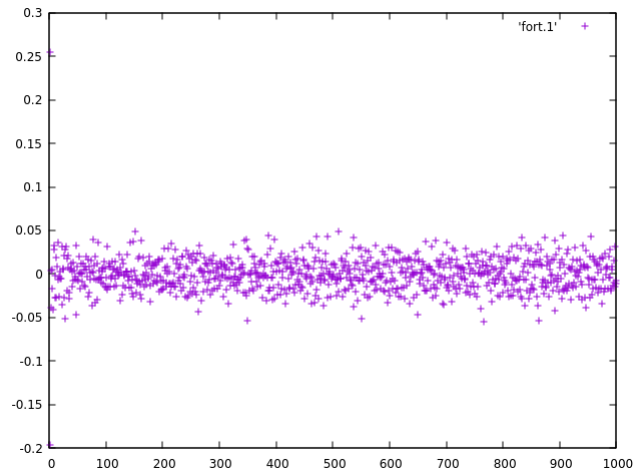


Figura 7: Magnetização versus interações

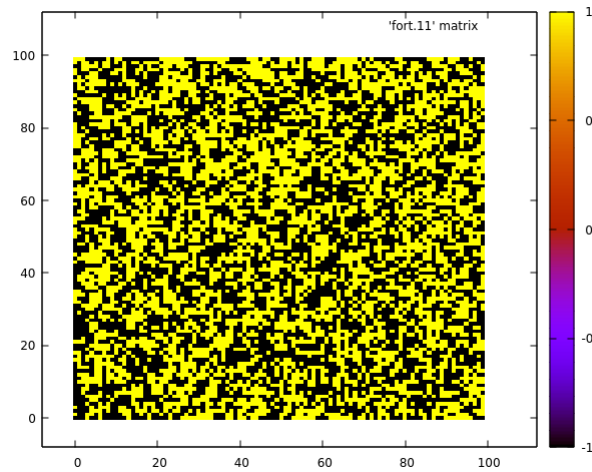


Figura 8: Organização dos spins, na ultima configuração

Vemos que para sistema onde o valor de  $\beta = 0.1$  o sistema fica desordenado, a temperatura no sistema é alta o sistema tende a desordem, isso ocorre que energia cinética nas moléculas é maior que a potencial, e para  $\beta = 3$  o sistema se ordena, isso ocorre por causa que estamos em temperatura baixa, a potencial ganha da cinética, portanto, as partículas se organizam melhor.

## 2.2 Código

```

1 program carlo
2   integer :: l
3   !criar um vetor para exponencial
4   real*8 :: magne, magnean
5   real, dimension(-4:4) :: pma, pme
6   integer, dimension(1000) :: plus, minus

```



```

7  byte s(1000,1000)!gerando a matriz spin s(i,j)
8  character*1 isimb(-1:1)
9  integer,parameter :: seed = 15000!gerando uma semente
10 call srand(seed)
11 nao=rand(seed)!ignorar essa linha
12
13
14 isimb(1) = "+"
15 isimb(-1) = "-"
16
17 !inserindo dados
18 write(*,*)'escreva o valor de beta'
19 read(*,*)b
20 write(*,*)'escreva o tamanho do sistema'
21 read(*,*)l
22 magne=0!zerando valor media magn tica
23
24 do j=-4,4,2!vetores dos expoentes
25   pma(j)=exp(j*b)
26   pme(j)=exp(-j*b)
27 enddo
28
29 do k=1,l!gerando o vetor minus e plus
30   plus(k)=k+1
31   minus(k)=k-1
32 enddo
33 plus(1)=1
34 minus(1)=1
35
36
37 !gerando o sistema de spin
38 do i=1,l
39   do j=1,l
40     s(i,j)=1
41   enddo
42 enddo
43
44 do i=1,l!primeira configuracao
45   write(10,*)(s(i,j),j=1,l)
46 enddo
47
48 do ia=1,100!intera es
49
50   do ic=1,(l)**2 !contador de monte carlos l**2
51
52     !escolhendo uma coordenada aleatorio.
53     i=(l)*rand()+1
54     j=(l)*rand()+1
55
56     !calculo da probabilidade
57     M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
58     prob=pma(M)/(pma(M)+pme(M))
59
60     if(rand()>=Prob)then!probabilidade do spin mudar

```

```

61         s(i,j)=s(i,j)*(-1)
62     endif
63 enddo
64
65 do i=1,l!somando os spin para calcular o magnetismo
66     do j=1,l
67         magne=magne+s(i,j)
68     enddo
69 enddo
70
71 write(2,*)ia,magne/(l)**2!media magnetica
72 magne=0
73 enddo
74 do i=1,l!ultima configuracao
75     write(11,*)(s(i,j),j=1,l)
76 enddo
77 do i=1,l!ultima configuracao, usando txt
78     write(12,*)(isimb(s(i,j)),j=1,l)
79 enddo
80 end program

```

## 3 Tarefa-B

### 3.1 B1

#### 3.1.1 Discursões

Existem dois processos térmicos no mundo da metalurgia, tais processos determina a dureza do ductilidade do aço. Esses processos são recozimento e a têmpera. Assim vamos colocar nosso sistema variar sua temperatura  $\Delta\beta \ll 1$ , de forma que praticamente que cada variação o sistema esteja mais próximo do equilíbrio.

Para tarefa b1 começamos o sistema  $\beta = 0 (T \rightarrow \infty)$ , alteramos  $\Delta\beta = 0.001$ , até chegar o valor de  $\beta=3$ , quando a temperatura tiver baixa, sendo esse processo o recozimento.

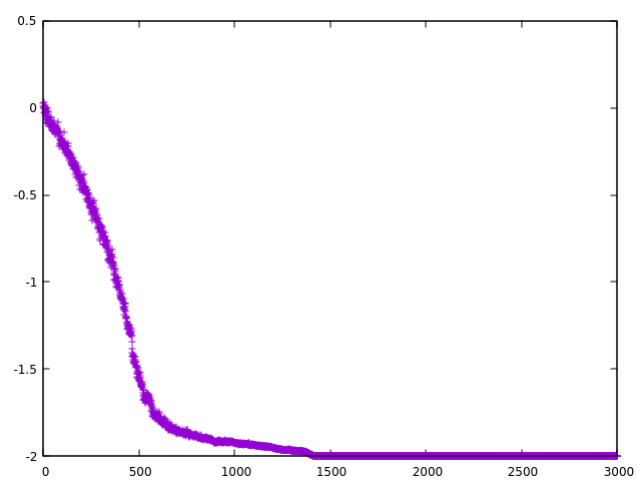


Figura 9: Energia do sistema

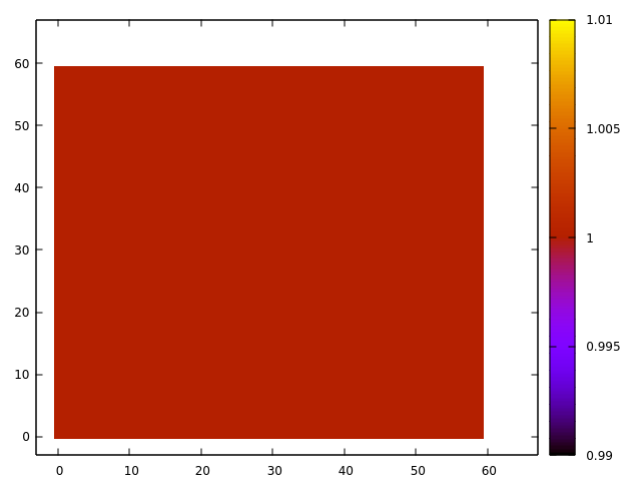


Figura 10: Configuração do sistema

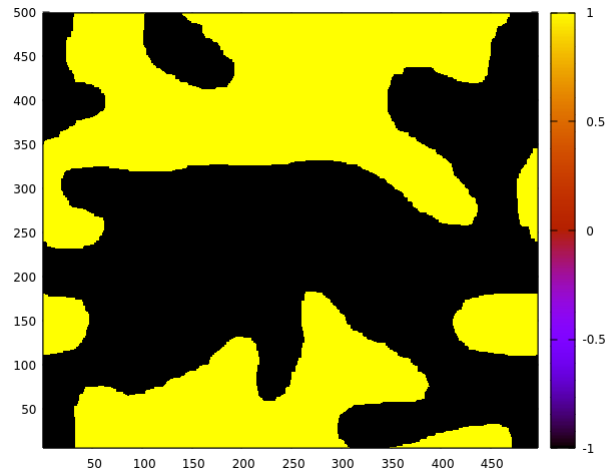


Figura 11: Rede L=500

Conforme a  $\beta$  aumenta, assim temperatura vai abaixando ( $\beta$  é definido  $\frac{1}{KT}$ ) o sistema vai ordenando como vemos na figura 10, no fim esperávamos um sistema ordenado, para rede maiores o sistema com alguma regiões positiva e negativa como vemos na figura 11.

### 3.1.2 Código

```

1 program carlo
2   integer :: l,a,c
3   !criar um vetor para exponencial
4   real*8::magne,magnean
5   real,dimension(-4:4)::pma,pme,ale
6   integer,dimension(1000)::plus,minus
7   byte s(1000,1000)!spin s(i,j)
8   integer,parameter :: seed = 15400!para salva a seed
9   call srand(seed)
10  nao=rand(seed)!ignorar essa linha
11
12  character*1 isimb(-1:1)
13  isimb(1) = "+"
14  isimb(-1) = "-"
15
16  !entrada de dados
17  write(*,*)'escreva o tamanho do sistema '
18  read(*,*)l
19
20  b=0
21  ener=0!zerando valores
22  magne=0
23  ale(1)=-1
24  ale(2)=1
25
26
27  do j=-4,4,2!vetores dos expoentes
28    pma(j)=exp(j*b)

```

```

29     pme(j)=exp(-j*b)
30 enddo
31
32 do k=1,l!gerando o vetor minus e plus
33     plus(k)=k+1
34     minus(k)=k-1
35 enddo
36 plus(1)=1
37 minus(1)=1
38
39 !gerando o sistema de spin
40 do i=1,l!media magn tica
41     do j=1,l
42         ir=2*rand()+1
43         s(i,j)=ale(ir)
44     enddo
45 enddo
46
47 do i=1,l!primeira configura o
48     write(10,*)(s(i,j),j=1,l)
49 enddo
50
51
52 do a=1,3000
53     do ic=1,(l)**2 !contador de monte carlos l**2
54
55         !escolhendo uma coordenada aleatorio.
56         i=(l)*rand()+1
57         j=(l)*rand()+1
58
59         !calcula da probabilidade
60         M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
61         prob=pma(M)/(pma(M)+pme(M))
62
63         if(rand()>=Prob)then!probabilidade do spin mudar
64             s(i,j)=s(i,j)*(-1)
65         endif
66     enddo
67
68     do i=1,l!energia
69         do j=1,l
70             ener=ener+s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,
71             minus(j)))
72         enddo
73     enddo
74
75     write(4,*)a,-ener*0.5d0/(l)**2!media da energia por spin
76     ener=0
77
78     b=b+0.001!aumentando o beta
79     do j=-4,4,2!reajustando vetores dos expoentes
80         pma(j)=exp(j*b)
81         pme(j)=exp(-j*b)
82     enddo

```

```

82     do i=1,1!calculo da magnetiza o
83         do j=1,1
84             magne=magne+s(i,j)
85         enddo
86     enddo
87     write(7,*)a,magne*1.0d0/(1)**2!magnetiza o m dia
88     magne=0
89
90 enddo
91 do i=1,1!plot da ultima configura o
92     write(2,*)(s(i,j),j=1,1)
93 enddo
94 do i=1,1!ultima configuracao, usando txt
95     write(12,*)(isimb(s(i,j)),j=1,1)
96 enddo
97 end program

```

## 3.2 B2

### 3.2.1 Discussões

O processo chamado têmpera, onde submetemos o sistema em uma variação drástica de temperatura. Na nossa tarefa começamos sistema com temperatura  $\beta = 0$  e resfriamos instantaneamente. Queremos ver como sistema se comporta, se irá para o equilíbrio com essa mudança brusca.

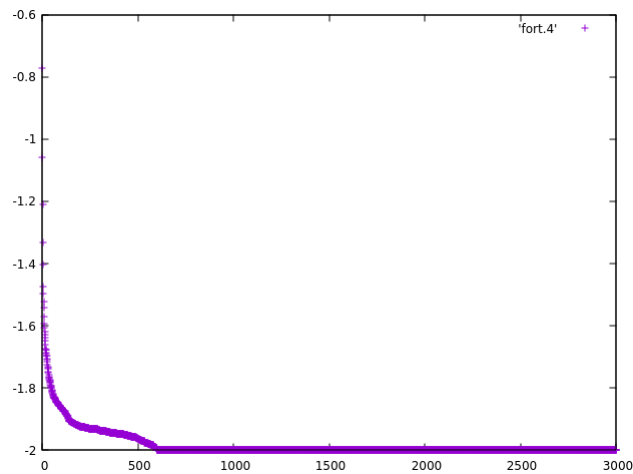


Figura 12: Energia do sistema

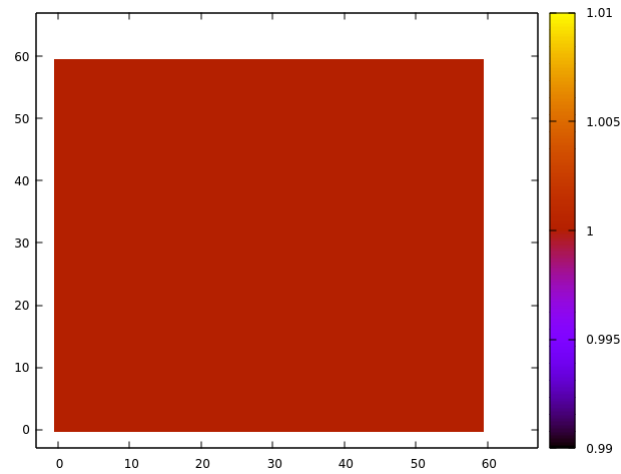


Figura 13: Configuração do sistema

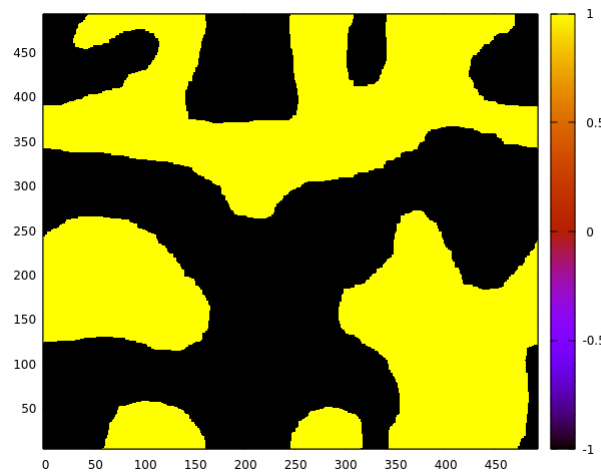


Figura 14: Rede L=500

Na vemos na figura 14, o sistema cria regiões de positivo e negativo, pela mudança brusca de temperatura sistema acaba criando essa região, assim, não irá completamente para o equilíbrio.

### 3.2.2 Código

```

1 program b2
2   integer :: l
3   !criar um vetor para exponencial
4   real*8::magne,magnean
5   real,dimension(-4:4)::pma,pme,ale
6   integer,dimension(1000)::plus,minus
7   byte s(1000,1000)!gerando a matriz spin s(i,j)
8   character*1 isimb(-1:1)
9   integer,parameter :: seed = 1500

```

```

10  call srand(seed)
11  nao=rand(seed)!ignorar essa linha
12  isimb(1) = "+"
13  isimb(-1) = "-"
14
15  l=500
16  ener=0!zerando valores
17  magne=0
18  ale(1)=-1
19  ale(2)=1
20  b=3
21
22  do j=-4,4,2!vetores dos expoentes
23      pma(j)=exp(j*b)
24      pme(j)=exp(-j*b)
25  enddo
26
27  do k=1,l!gerando o vetor minus e plus
28      plus(k)=k+1
29      minus(k)=k-1
30  enddo
31  plus(1)=1
32  minus(1)=1
33
34  !gerando o sistema de spin
35  do i=1,l
36      do j=1,1
37          ir=2*rand()+1
38          s(i,j)=ale(ir)
39      enddo
40  enddo
41  do i=1,l
42      write(3,*)(s(i,j),j=1,1)
43  enddo
44  do i=1,l
45      write(4,*)(s(i,j),j=1,1)
46  enddo
47  do ia=1,1000
48      do ib=1,3
49          do ic=1,(l)**2 !contador de monte carlos l**2
50
51              !escolhendo uma coordenada aleatorio.
52              i=(l)*rand()+1
53              j=(l)*rand()+1
54
55              !calcula da probabilidade
56              M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
57              prob=pma(M)/(pma(M)+pme(M))
58
59              if(rand()>=Prob)then!probabilidade do spin mudar
60                  s(i,j)=s(i,j)*(-1)
61              endif
62          enddo
63

```



```

64      do i=1,l!energia do sistema e magnetismo no sistema
65          do j=1,l
66              ener=ener+s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,
minus(j)))
67              magne=magne+s(i,j)
68          enddo
69      enddo
70      !energia media
71      write(4,*)ia,-ener*0.5d0/(l)**2
72      ener=0
73      write(7,*)ia,(magne)*1.0d0/(l)**2!media do magnetismo
74      magne=0
75  enddo
76  do i=1,l!ultima configuracao
77      write(10+ia,*)(s(i,j),j=1,l)
78  enddo
79  enddo
80  do i=1,l!ultima configuracao
81      write(2,*)(s(i,j),j=1,l)
82  enddo
83  do i=1,l!ultima configuracao, usando txt
84      write(7,*)(isimb(s(i,j)),j=1,l)
85  enddo
86  end program

```

## 4 Tarefa-C

### 4.1 C1

#### 4.1.1 Discussões

Nessa tarefa faremos o loop térmico. Chamamos o loop térmico o processo, onde passamos o sistema de uma temperatura infinita até um valor próximo de zero absoluto, quando atingi essa temperatura próxima de zero absoluto, aquecemos o sistema até a temperatura infinita. Vamos realizar essa tarefa usando interações de Monte Carlo.

Começamos o sistema na temperatura infinita e varia até  $\beta = 0$  até 1.75, quando beta atingir esse valor aquecemos novamente o sistema, para procurar o intervalo de histerese que é região onde está a temperatura crítica. Vamos simular três tamanho de rede diferente  $L=60,80$  e  $100$  e vamos variar o  $\Delta\beta$  entre  $0.001$  e  $0.0001$ .

Rede  $L=100$

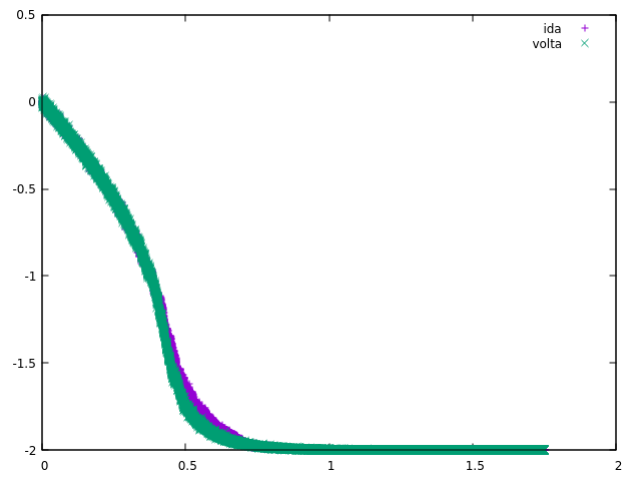


Figura 15:  $\Delta\beta = 0.0001$

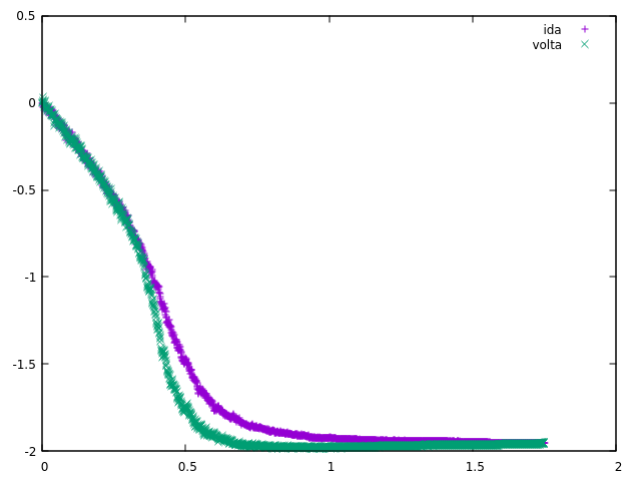


Figura 16:  $\Delta\beta = 0.001$

Rede L=80

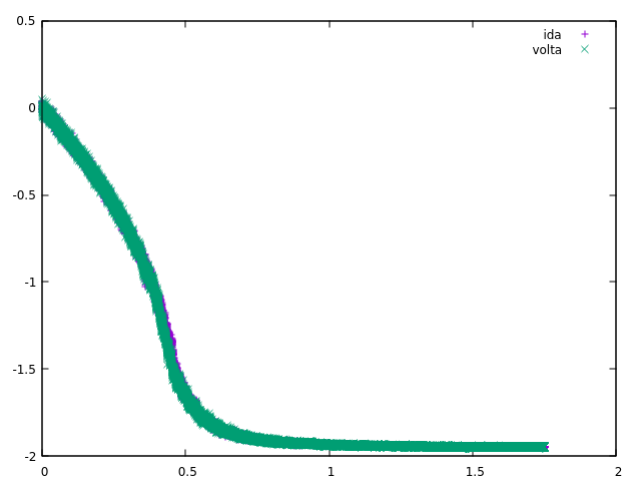


Figura 17:  $\Delta\beta = 0.0001$

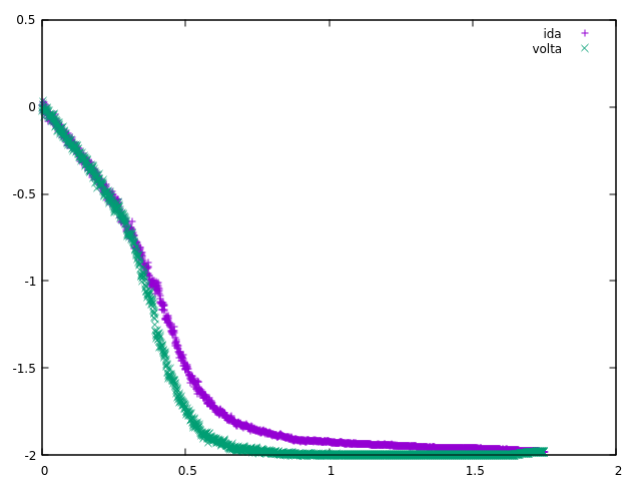


Figura 18:  $\Delta\beta = 0.001$

Rede=60

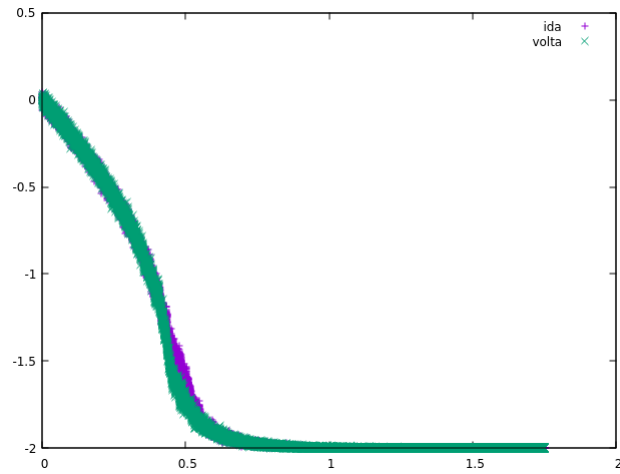


Figura 19:  $\Delta\beta = 0.0001$

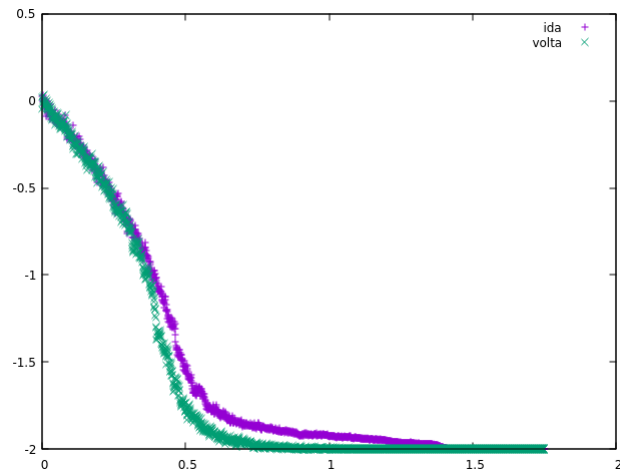


Figura 20:  $\Delta\beta = 0.001$

A região que a histerese fica entre  $\beta=0.35$  e  $0.5$ , podemos ver o resultado melhor com a imagem abaixo, onde foi dado um zoom, portanto, vemos que em todos os casos a histerese não muda muito a região de seu intervalo.

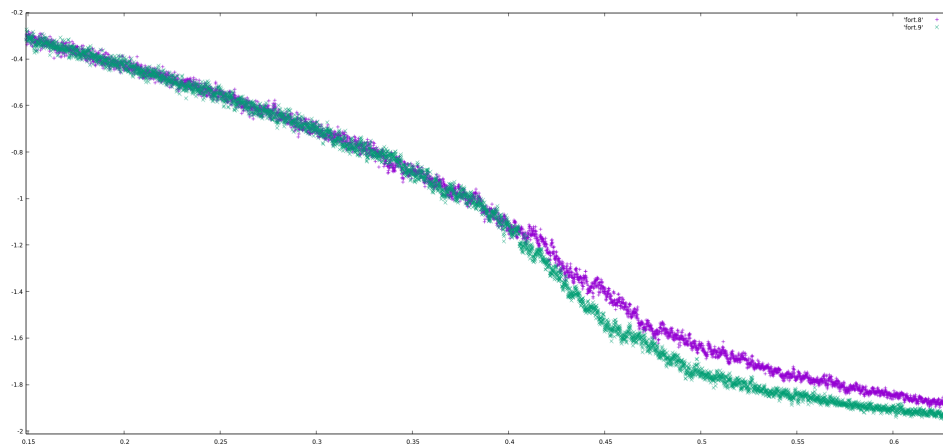


Figura 21

#### 4.1.2 Código

```

1 program carlo
2   integer :: l
3   !criar um vetor para exponencial
4   real*8::magne,magnean
5   real,dimension(-4:4)::pma,pme,ale,beta
6   integer,dimension(1000)::plus,minus
7   byte s(1000,1000)!spin s(i,j)
8   integer,parameter :: seed = 15400!gerando seed aleatoria.
9   call srand(seed)
10  nao=rand(seed)!para salva a seed
11  write(*,*)'escreva o tamanho do sistema (l)'
12  read(*,*)l
13  write(*,*)'escolha o valor da variancia da temperatura'
14  read(*,*)db
15  ener=0!zerando valores
16  magne=0
17  b=0
18  a=0
19  !valor para colocar spins aleatorio
20  ale(1)=-1
21  ale(2)=1
22  !vetor para mudan a de temperatura
23  beta(1)=db
24  beta(2)=-db
25
26  ialpha=1
27  do j=-4,4,2!vetores dos expoentes
28    pma(j)=exp(j*b)
29    pme(j)=exp(-j*b)
30  enddo
31
32  do k=1,l!gerando o vetor minus e plus
33    plus(k)=k+1
34    minus(k)=k-1

```

```

35  enddo
36  plus(1)=1
37  minus(1)=1
38
39  !gerando o sistema de spin
40  do i=1,1!media magnética
41      do j=1,1
42          ir=2*rand()+1
43          s(i,j)=ale(ir)
44      enddo
45  enddo
46
47  do i=1,1!primeira configuração
48      write(3,*)(s(i,j),j=1,1)
49  enddo
50
51
52  do while(b>=0)
53
54      do ic=1,(1)**2 !contador de monte carlos 1**2
55
56          !escolhendo uma coordenada aleatorio.
57          i=(1)*rand()+1
58          j=(1)*rand()+1
59
60          !calculando a probabilidade
61          M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
62          prob=pma(M)/(pma(M)+pme(M))
63
64          if(rand()>=Prob)then!probabilidade do spin mudar
65              s(i,j)=s(i,j)*(-1)
66          endif
67      enddo
68      a=a+1
69
70      do i=1,1!energia média
71          do j=1,1
72              ener=ener+s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,
73              minus(j)))
74          enddo
75      enddo
76
77      b=b+beta(ialpha)!alterando a temperatura
78      do j=-4,4,2!reajustando vetores dos expoentes
79          pma(j)=exp(j*b)
80          pme(j)=exp(-j*b)
81      enddo
82
83      !write(*,*)magne*1.0d0/(1)**2,abs(magne-maganean)*1.0d0/(1)**2
84      write(1,*)a,-ener*0.5d0/(1)**2
85      write(7+ialpha,*)b,-ener*0.5d0/(1)**2!energia em função da
86      temperatura
87      ener=0

```

```

87
88     do i=1,l!media magn tica
89         do j=1,l
90             magne=magne+s(i,j)
91         enddo
92     enddo
93
94     write(7,*)a,magne*1.0d0/(l)**2!escrevendo magnetiza o
95     magne=0
96
97
98     if (b>=1.75)then!verificando se beta maior que 1.75, caso for
99     aquecer o sistema novamente.
100         ialpha=2
101         do i=1,l
102             write(10,*)(s(i,j),j=1,l)
103         enddo
104     endif
105
106     do i=1,l
107         write(2,*)(s(i,j),j=1,l)
108     enddo
109 end program

```

## 4.2 C2

### 4.2.1 Discussões

Vamos agora procurar o valor da temperatura crítica da rede, utilizamos as informações que obtivemos anteriormente onde descobrimos a região da histerese que é 0.35 a 0.5. Para encontrar a temperatura crítica, temos que encontrar um  $\beta$  onde sistema não tende a para desordenado ou ordenado, assim vamos dividir o sistema entre desordem ( $T \rightarrow \infty$ ) e ordem ( $T \rightarrow 0$ ), portanto caso, a temperatura não seja crítica a energia irá reduzir ou aumentar, caso seja a energia irá se manter o mesmo valor.

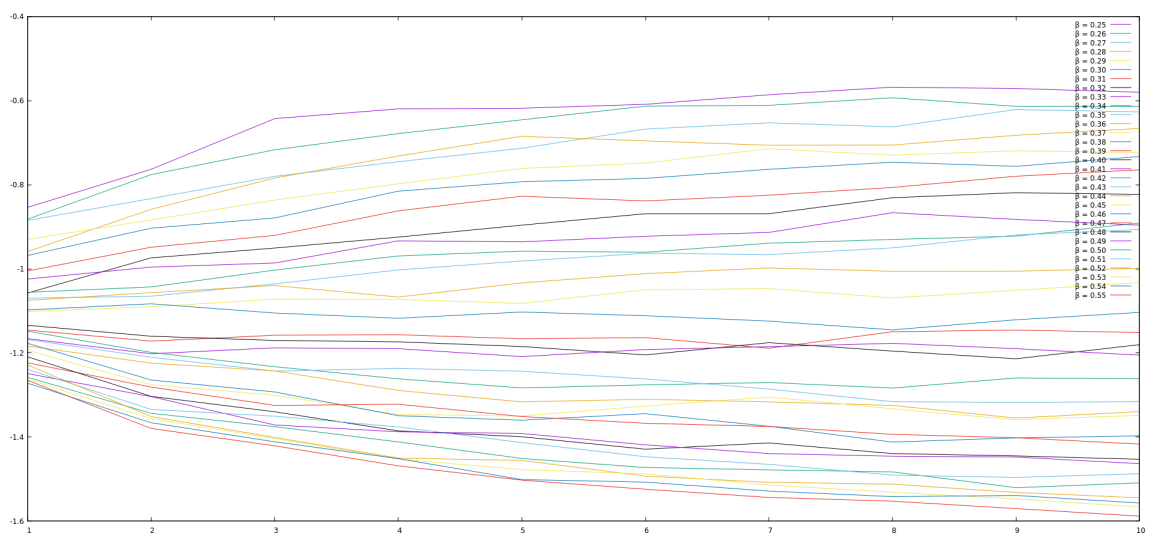


Figura 22: Rede de Tamanho  $L=100$

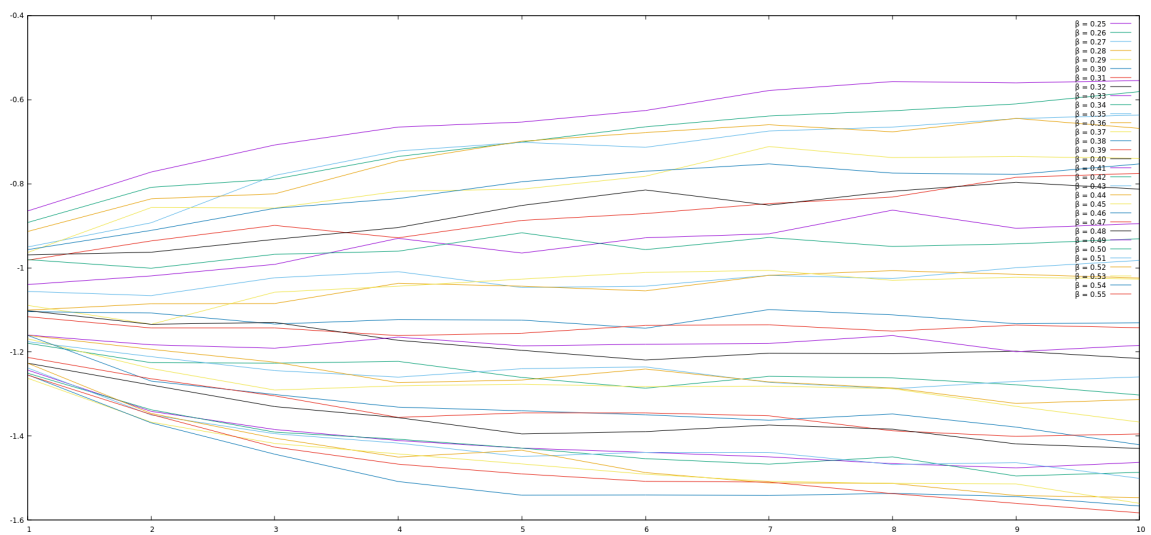


Figura 23: Rede de Tamanho  $L=80$



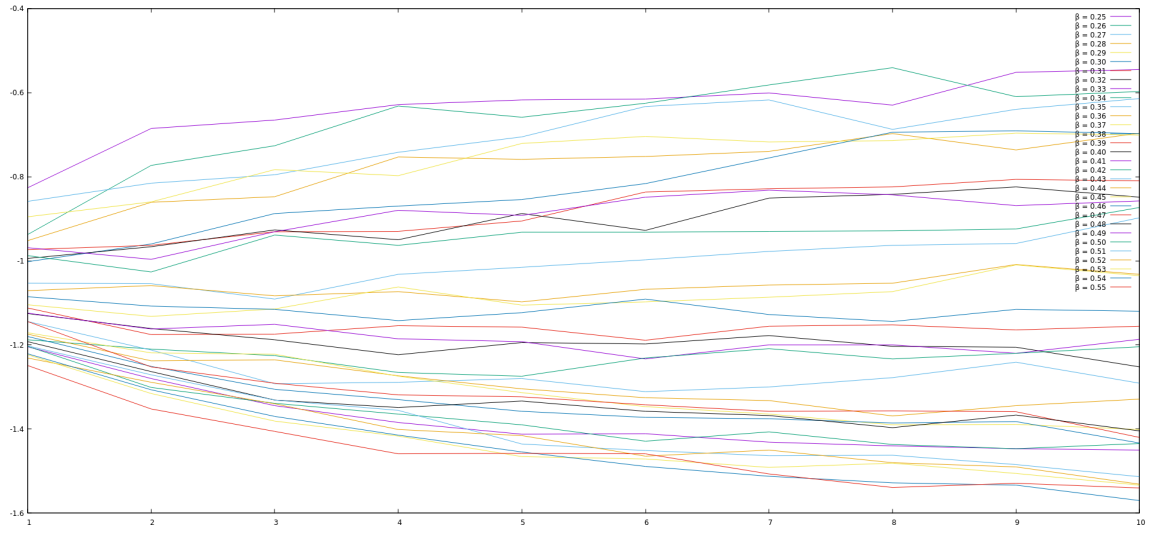


Figura 24: Rede de Tamanho L=60

Se darmos um zoom em uma das imagens, ficamos com a seguinte imagem abaixo, onde chegamos que  $\beta$  crítico 0.38 a 0.41, podemos verifica que  $\frac{1}{KT_c} = \beta = 0.39$  é valor que menos varia da sua energia original.

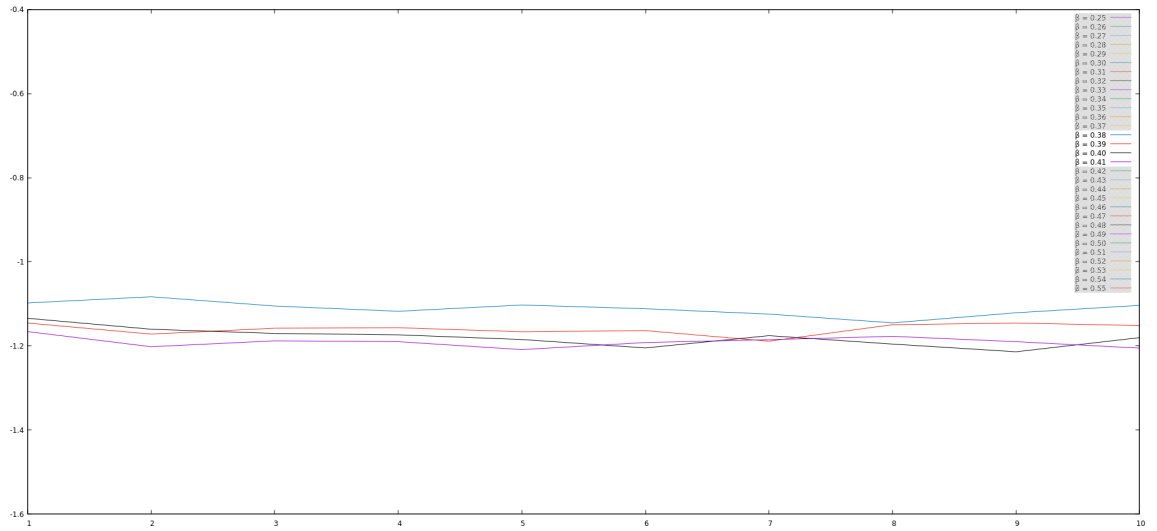


Figura 25

1	-1.1456000000000000
2	-1.1719999999999999
3	-1.1579999999999999
4	-1.1568000000000001
5	-1.1664000000000001
6	-1.1636000000000000
7	-1.1888000000000001
8	-1.1496000000000000
9	-1.1456000000000000
10	-1.1516000000000000

Figura 26: Variação da energia

#### 4.2.2 Código

```

1 program carlo
2   integer :: l, a, alpha
3   !criar um vetor para exponencial
4   real*8 :: magne, magnean
5   real, dimension(-4:4) :: pma, pme, ale, beta
6   integer, dimension(1000) :: plus, minus
7   byte s(1000,1000) !spin s(i,j)
8   write(*,*) 'escreva o tamanho do sistema'
9   read(*,*) l
10  ener=0 !zerando valores
11  magne=0
12  ale(1)=-1
13  ale(2)=1
14
15  do k=1,l !gerando o vetor minus e plus
16    plus(k)=k+1
17    minus(k)=k-1
18  enddo
19  plus(1)=1
20  minus(1)=1
21
22  do ic=1,31
23    b=0.24+ic*0.01 !mudan a do beta indo 0.35      0.45
24
25    do j=-4,4,2 !vetores dos expoentes
26      pma(j)=exp(j*b)
27      pme(j)=exp(-j*b)
28    enddo
29
30    !gerando o sistema de spin
31    do i=1,l
32      do j=1,l
33        if(i>=l/2) then
34          ir=2*rand()+1
35          s(i,j)=ale(ir)
36        else
37          s(i,j)=1
38        endif
39      enddo
40    enddo

```

```

41
42 do a=1,10
43
44 do id=1,(l)**2 !contador de monte carlos l**2
45
46 !escolhendo uma coordenada aleatorio.
47 i=(l)*rand()+1
48 j=(l)*rand()+1
49
50 !calculo da probabilidade
51 M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
52 prob=pma(M)/(pma(M)+pme(M))
53
54 if(rand()>=Prob)then!probabilidade do spin mudar
55 s(i,j)=s(i,j)*(-1)
56 endif
57 enddo
58
59 do i=1,l!energia
60 do j=1,l
61 ener=ener+s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,
minus(j)))
62 enddo
63 enddo
64
65
66 write(24+ic,*)a,-0.5d0*ener/(l)**2!energia media
67 ener=0
68
69
70 do i=1,l!magnetiza o
71 do j=1,l
72 magne=magne+s(i,j)
73 enddo
74 enddo
75
76 write(7,*)a,magne*1.0d0/(l)**2!media da magnetiza o
77 magne=0
78
79 enddo
80 enddo
81 end program

```

## 5 Tarefa-D

### 5.1 Discussões

Aqui vamos verificar, um conceito importante chamado quebra espontânea de simetria.

O modelo de Ising tem uma simetria por inversão global dos spins, também chamada por simetria global  $Z(2)$ . Essa simetria se dá no caso de inversão de todos os spins  $s \rightarrow -s$  a

energia é mesma. A probabilidade de inversão de dada pela energia no sistema, assim as duas configurações possuem a mesma chance de ocorrer. Em sistemas maiores probabilidade de que inversão fica cada vez menor, mas Na realidade, se estivermos com uma das ordenações e se esperarmos por tempo suficiente, verificaremos que o sistema vai para configurações de outra ordenação, e mais adiante viria para a ordenação antiga e assim sucessivamente. Se  $L \rightarrow \infty$ , o chance da rede sair do estado ordenado  $s \rightarrow -s$  para o inverso dela fica nula.

Então vamos apreciar esse conceito usando, as seguintes redes  $L=4,5,6,7,8,9$  e  $10$ , fazendo uma média de quantas vezes o sistema se inverte, assim mostrando que para rede maior chance diminuir exponencialmente.

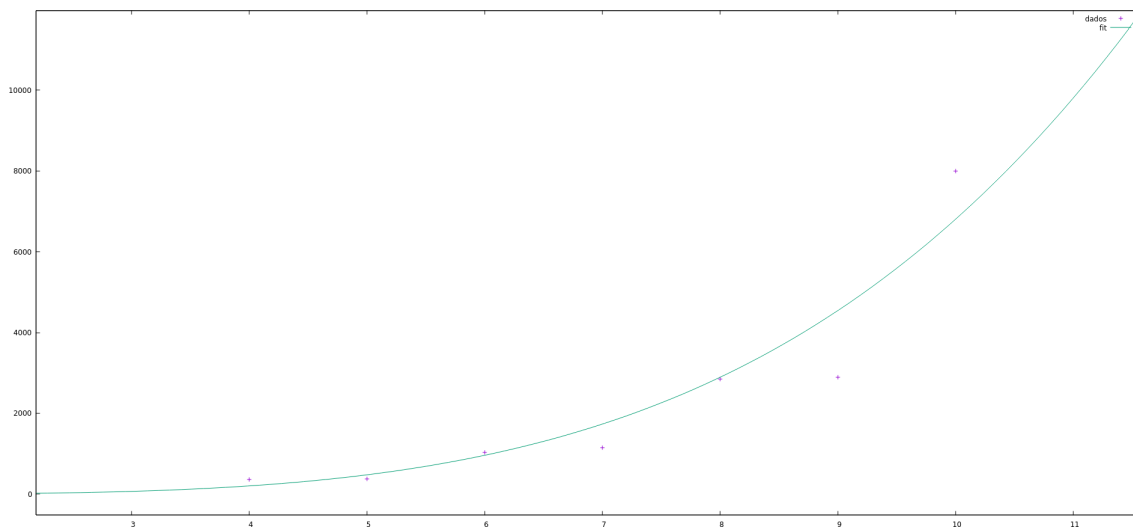


Figura 27

Para o fit  $x^d$  conseguimos um valor  $d = 3.83 \pm 0.04$ , assim vemos que  $L \rightarrow \infty$  (que pode ser comparada a rede muito grande, como a vida real que é  $10^{23}$ ), vai dar zero rapidamente.

## 5.2 Código

```

1 program spind
2   integer :: l,a,c,af,be
3   !criar um vetor para exponencial
4   real*8::magne,magnean
5   real,dimension(-4:4)::pma,pme
6   integer,dimension(1000)::plus,minus,ale
7   byte s(1000,1000)!spin s(i,j)
8   integer,parameter :: seed = 14506
9   call srand(seed)
10  nao=rand(seed)!fim
11  b=0.5d0
12  magne=0!zerando valor media magn tica
13  magnea=0
14  ale(1)=1

```

```

15 ale(2)=-1
16 Ti=0
17 do j=-4,4,2!vetores dos expoentes
18   pma(j)=exp(j*b)
19   pme(j)=exp(-j*b)
20 enddo
21
22 do l=4,10,1
23
24   do k=1,l!gerando o vetor minus e plus
25     plus(k)=k+1
26     minus(k)=k-1
27   enddo
28   plus(1)=1
29   minus(1)=1
30
31   !gerando o sistema de spin
32   do i=1,l
33     do j=1,l
34       ir=2*rand()+1
35       s(i,j)=ale(ir)
36     enddo
37   enddo
38
39
40   do a=1,200000!escolhendo uma coordenada aleatorio.
41     do c=1,(l)**2
42       i=(l)*rand()+1
43       j=(l)*rand()+1
44
45       M=s(i,j)*(s(plus(i),j)+s(minus(i),j)+s(i,plus(j))+s(i,minus(j)))
46       prob=pma(M)/(pma(M)+pme(M))
47
48       if(rand()>=prob)then
49         s(i,j)=s(i,j)*(-1)
50       endif
51     enddo
52
53     do i=1,l!media magn tica
54       do j=1,l
55         magne=magne+s(i,j)
56       enddo
57     enddo
58
59     if(magne*magnea<0)then!quando magnetismo trocar de sinal
contabilizar
60       Ti=Ti+1
61     endif
62     write(10+l,*)a,magne/(l)**2
63     magnea=magne!magnea representa a magnetiza o anterior
64     magne=0
65
66   enddo
67   !salvando os dados

```

```
68     write(2,*)1,a/Ti
69     write(3,*)1,log(a/Ti)
70     Ti=0
71     enddo
72 end program
```