

# 計算物理学 第四単元レポート

61908697 佐々木良輔

## 1 5-A.

### 1.1 プログラムについて

プログラムはサンプルコードを元に各ブロック毎に計算された物理量をさらに平均化するように修正したものを用いている。変数 `sume`, `sumc`, `summ`, `sumx` は各ブロックで計算された内部エネルギー, 比熱, 磁化, 帯磁率の合計値を格納し, 最後にブロック数で割ることで物理量の平均値を計算している。プログラムの修正部分をソースコード A.1 に示す。

### 1.2 結果

#### 1.2.1 乱数の検証

乱数が十分高いエントロピーを持っていることを確認するため, いくつかのシード値に対してシミュレーションを行いシード値と物理量 (内部エネルギー) との相関を調べた。相関係数は `gnuplot` の `stats` コマンドを用いて計算した。計算条件は表 1 の通りである。シード値は 3 から 601 までの全奇数 (300 点) とした。結果は図 1 のようになった。シード値と物理量の相関係数は  $R = 0.02$  であった。また標準偏差は

$$\text{sume} = -1.9511(1 \pm 0.0002) \quad (1)$$

となった。

表 1 計算条件

無次元量	値
格子サイズ	$8 \times 8$
温度	1.5
$M_i$	10000
$M_f$	20000
ブロック数	10

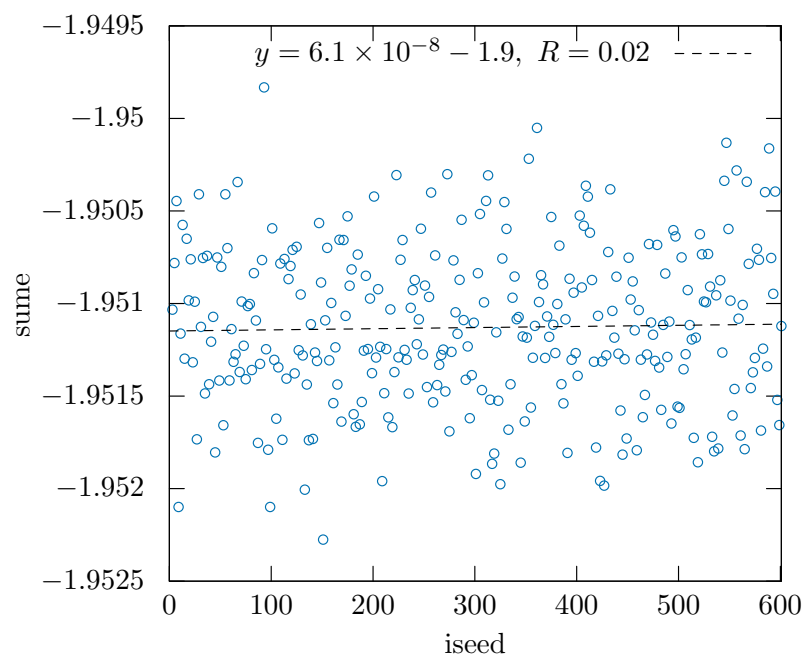


図 1 シード値と内部エネルギーの関係

### 1.2.2 シミュレーション結果

いくつかの温度で内部エネルギーと比熱のシミュレーションを行った。また同じ温度での厳密解の値を計算した。その結果は表 3 のとおりである。また計算条件は表 2 の通りである。

表 2 計算条件

無次元量	値
格子サイズ	$8 \times 8$
$M_i$	10000
$M_f$	20000
ブロック数	10
シード値	601

表 3 計算結果

温度	スピン初期状態	シミュレーション		厳密解	
		内部エネルギー	比熱	内部エネルギー	比熱
1.5	all up	-1.951	0.1952	-1.951	0.1972
2.3	all up	-1.452	1.176	-1.456	1.170
2.3	random	-1.453	1.174	-1.456	1.170
3.0	random	-0.8420	0.4850	-0.8413	0.4840

## 1.3 考察

### 1.3.1 乱数の検証

前節で述べた通り乱数のシード値と計算される物理量の間のコ相関係数は  $R = 0.02$  と非常に小さく、シード値は条件を満たす限りにおいて任意に取って良いものと考えられる。

またシード値の変化による物理量の標準偏差も  $0.02\%$  と非常に小さく、このことから計算結果に与えるシード値の影響は十分小さいと考えられる。

以上から今後の計算においては乱数は十分高いエントロピーを持ち、計算結果にシード値による誤差などは現れていないものとする。

### 1.3.2 シミュレーション結果

各条件でのシミュレーション結果の厳密解との相対誤差は表 4 の通りである。このことから全ての条件でシミュレーションの結果と厳密解が良く一致しており、シミュレーションの結果が妥当であると考えられる。

またシミュレーションの結果から比熱は臨界温度付近で大きくなっていることがわかる。これについては次節の課題 5-B. で詳細に検討する,

また臨海温度 (2.3) の 2 つシミュレーションではスピン初期状態によらずどちらも厳密解に良く一致しており, 初期条件によらず平衡状態に収束していると考えられる.

表 4 相対誤差

温度	スピン初期状態	相対誤差 / %	
		内部エネルギー	比熱
1.5	all up	0.0	1.0
2.3	all up	0.2	0.5
2.3	random	0.2	0.4
3.0	random	0.1	0.2

## 2 5-B.

### 2.1 プログラムについて

プログラムは 5-A のプログラムを元に温度を 4.0 から 0.0 まで 0.1 刻みで掃引しながらシミュレーションを行うように修正したものを用いている. また変数  $\text{sume2}$ ,  $\text{sumc2}$ ,  $\text{summ2}$ ,  $\text{sumx2}$  はそれぞれ内部エネルギー, 比熱, 磁化, 帯磁率の 2 乗平均を格納しており, 最後に平均の 2 乗を引き, 平方根を取ることで標準偏差を計算している. プログラムでは温度と各物理量の平均値, 標準偏差を出力している. プログラムの修正部分をソースコード A.2 に示す.

### 2.2 結果

温度を 4.0 から 0.0 まで 0.1 刻みで掃引しながら内部エネルギー, 比熱, 磁化, 帯磁率の温度依存性を計算した. 図 2 から図 5 にその結果を示す. また表 5 に計算条件を示す.

表 5 計算条件

無次元量	値
格子サイズ	$16 \times 16$
$M_i$	10000
$M_f$	20000
ブロック数	10
シード値	601

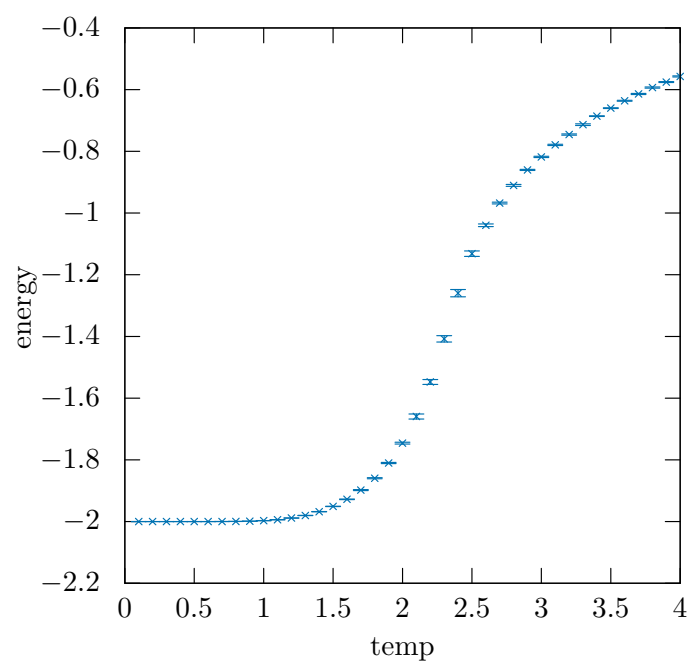


図 2 エネルギーの温度依存性

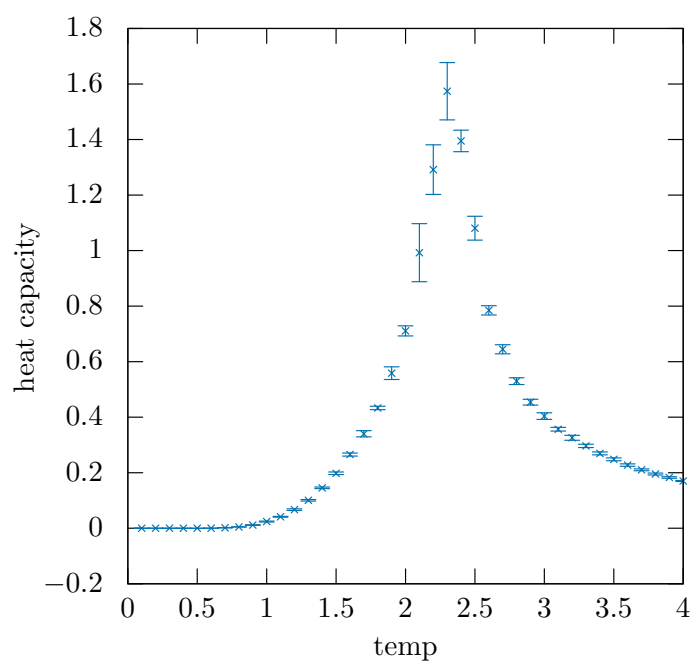


図 3 比熱の温度依存性

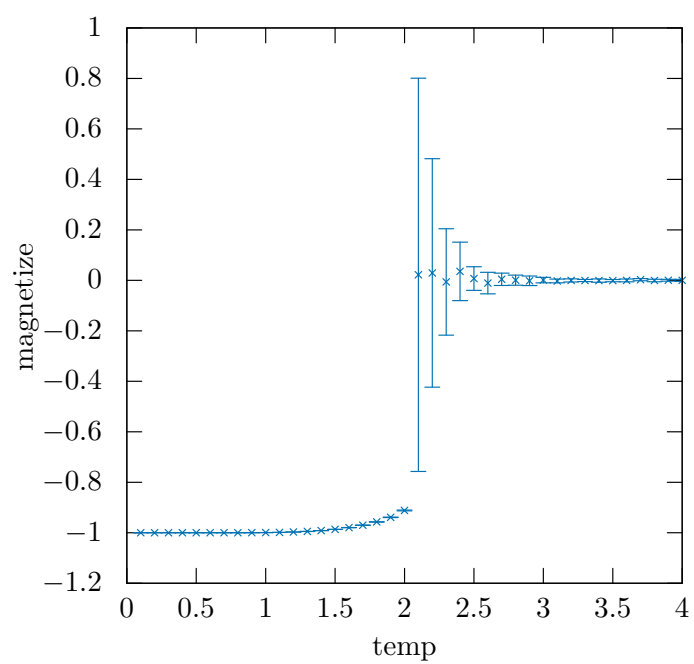


図 4 磁化の温度依存性

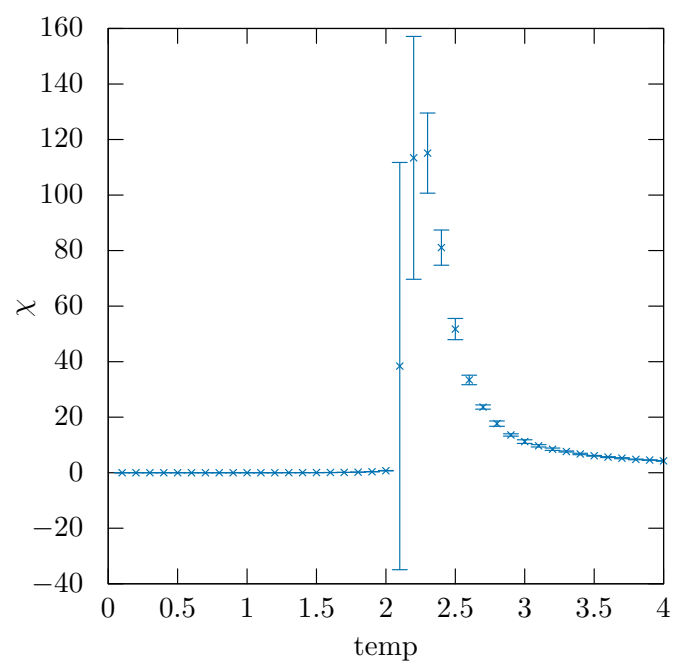


図 5 帯磁率の温度依存性

## 2.3 考察

### 2.3.1 内部エネルギーと比熱

図 6, 図 7 に内部エネルギーと比熱の厳密解とシミュレーションの結果を示す。破線は臨界温度 ( $\text{temp} = 2.269$ ) である。この結果から内部エネルギーと比熱についてはシミュレーションによって妥当な結果が得られていることがわかる。

図 7 から比熱が臨界温度付近で不連続に変化している。このことから相転移現象がシミュレーションでも現れていることがわかる。また比熱の標準偏差が臨界温度付近で大きくなっている。比熱が大きいということはエネルギー変化に対する温度変化の割合が小さいということであり、エネルギーの微小変化に対してより安定であると考えられる。特に Ising モデルにおいて、これはスピン反転が起こりやすいと理解できるため、これによって標準偏差が大きくなったと考えられる。

### 2.3.2 磁化と帯磁率

図 4, 図 5 から磁化や帯磁率についても臨界温度付近での不連続な変化が見られ、相転移現象が観測できる。これらに関しても不連続点付近で標準偏差が大きくなっており、これは比熱と同様の原理に依るものであると考えられる。

また図 9 に磁化、帯磁率の臨界温度付近での温度依存性を温度の冪で Fitting した結果を示す。これによって磁化の温度依存性は

$$M(x) = -1.008 \times (-x + 2.093)^{0.04265} \quad (2)$$

また帯磁率の温度依存性は

$$\chi(x) = 10.06 \times (x - 2.106)^{-1.734} \quad (3)$$

と求まった。したがって  $16 \times 16$  の 2 次元 Ising モデルでの臨界指数は

$$\begin{aligned} \beta &= 0.04266 \\ \gamma &= 1.734 \end{aligned} \quad (4)$$

である。



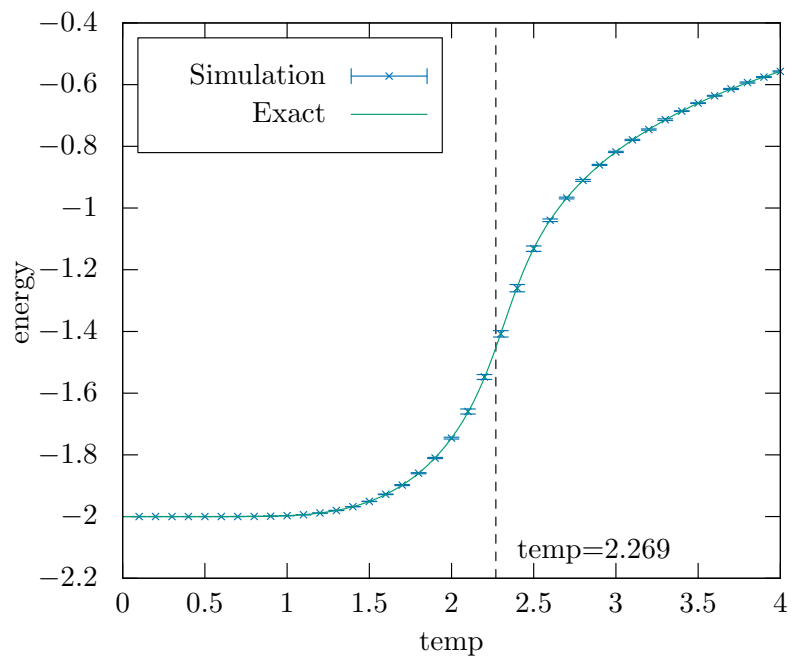


図 6 エネルギーの温度依存性

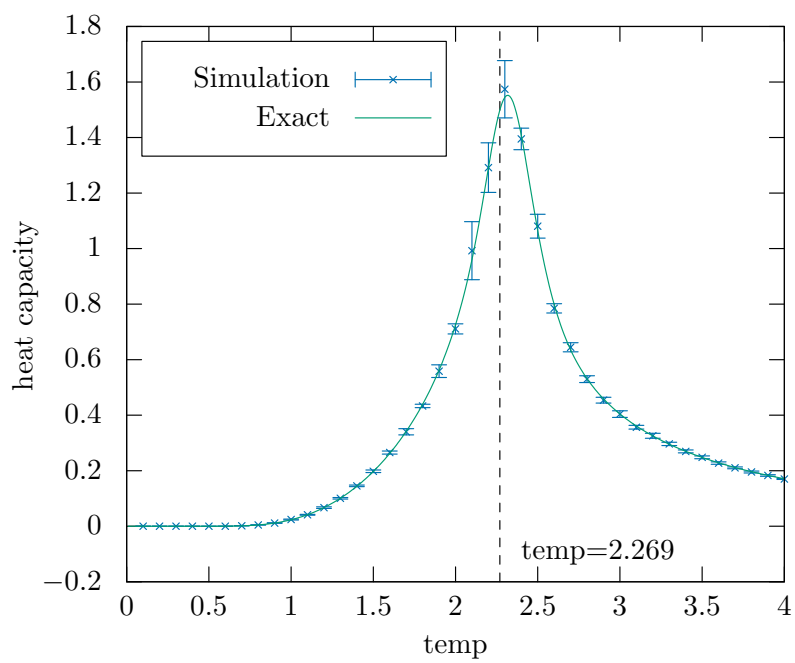


図 7 比熱の温度依存性

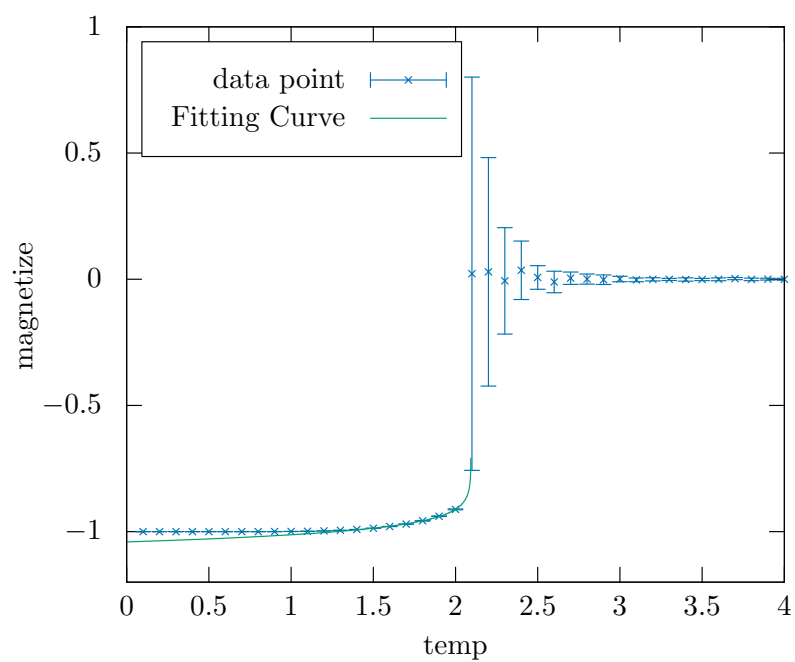


図 8 磁化と Fitting 曲線

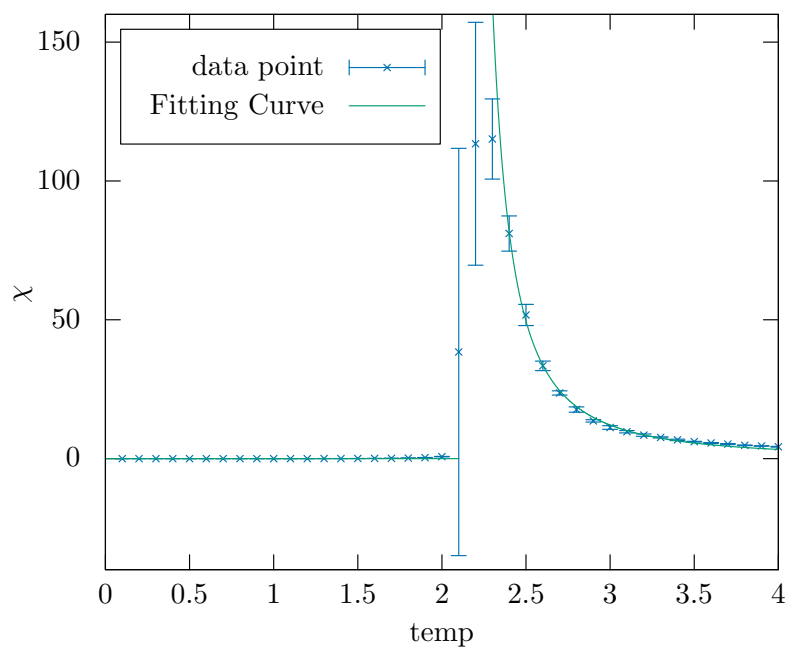


図 9 帯磁率と Fitting 曲線

## 補遺 A ソースコード

### ソースコード A.1 5-A のソースコード

---

```
1  !=====
2  ! block loop
3  !=====
4  sume=0;sumc=0;summ=0;sumx=0;
5  do nb=1,nblock
6      !=====
7      ! block-wa shokika
8      !=====
9      amag =0.0d0
10     amag2=0.0d0
11     aene =0.0d0
12     aene2=0.0d0
13
14     ~~~~~中略~~~~~
15
16     sume = sume + ave
17     sumc = sumc + avc
18     summ = summ + avm
19     sumx = sumx + avx
20 enddo
21 sume = sume / nblock
22 sumc = sumc / nblock
23 summ = summ / nblock
24 sumx = sumx / nblock
25 write(*,'(a10,ES24.16)') ' e= ',sume
26 write(*,'(a10,ES24.16)') ' c= ',sumc
27 write(*,'(a10,ES24.16)') ' m= ',summ
28 write(*,'(a10,ES24.16)') ' x= ',sumx
```

---

## ソースコード A.2 5-A のソースコード

---

```

1  !=====
2  ! sen'i-kakuritsu shokika
3  !=====
4  do while (temp>=0d0)
5      do n=-nnno,nnno
6          de=2.0d0*dble(n)/temp
7          if(de.gt.0.0d0) then
8              trprob(n)=dexp(-de)
9          else
10             trprob(n)=1.0d0
11         endif
12     enddo
13
14  ~~~~~中略~~~~~
15
16     sume = sume + ave
17     sumc = sumc + avc
18     summ = summ + avm
19     sumx = sumx + avx
20     sume2 = sume2 + ave*ave
21     sumc2 = sumc2 + avc*avc
22     summ2 = summ2 + avm*avm
23     sumx2 = sumx2 + avx*avx
24     enddo
25     sume = sume / nblock
26     sumc = sumc / nblock
27     summ = summ / nblock
28     sumx = sumx / nblock
29     sume2 = sume2 / nblock - sume*sume
30     sumc2 = sumc2 / nblock - sumc*sumc
31     summ2 = summ2 / nblock - summ*summ
32     sumx2 = sumx2 / nblock - sumx*sumx
33     sume2 = sqrt(sume2)
34     sumc2 = sqrt(sumc2)
35     summ2 = sqrt(summ2)
36     sumx2 = sqrt(sumx2)
37     write(*,'(ES24.16, ES24.16, ES24.16, ES24.16, ES24.16, ES24.16, ES24
        .16, ES24.16, ES24.16)') temp,sume,sume2,sumc,sumc2,&
38     summ,summ2,sumx,sumx2
39     temp = temp - 0.1d0
40 end do

```

---