

6

統計力学へ入る前に：配位空間と場合の数

統計力学の勉強をはじめてすぐにつまずくのが**配位空間 (configurational space)** の概念である。これは一般化した座標と運動量から構成される位相空間 (phase space) のうち、座標成分だけを指し、配置空間とも呼ばれる。実際に配位空間で場合の数を計算しようとするとき、配位空間は直感的な位置とは関係なくとられる。また、同じ対象でも、考え方によって違った配位空間をとる場合がある。逆に統計力学の問題を解くテクニックは、配位空間を問題に合わせて適切にとることにあるかも。統計力学を専門としない我々はそのコンセプトさえ理解できれば、テクニックは憶えなくても構わないはず。本章ではいくつかの例を示して、配位空間の感覚を掴んでもらう。

6.1 巡回セールスマン問題

最初に取り上げる**巡回セールスマン (traveling salesman) 問題**は、熱統計力学から少し外れている。これは、ある街から出発していくつかの街を次々とめぐって元の街に戻ってくる最短の経路を求める問題。訪れる街の数が少ないときにはすべての経路を数え上げればいいが、数が増えるとその計算時間は指数関数的に増えてしまうと予想される。このような問題はセールスマンだけでなく、コンピュータの CPU の配置や、都市ガスの配管設計などでも出会う。

街の配置が図 6.1 のようになっているとしよう。1つの街は2次元座標の特定の位置で表わされ、その並びが厳密な意味での配位空間である。しかし、Sales-

man 問題を解く場合には、1つ1つの街の細かな位置は問題にならない。あるいは問題が移動の所要時間や料金であるなら、位置座標そのものが意味を失う。問題になるのは、訪れる街の順番だけである。これが Salesman 問題の配位空間となる。

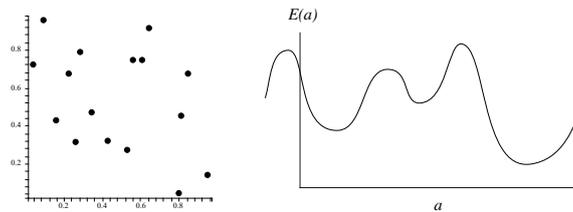


図 6.1 左図:街の位置を示す地図, 右図:配置 \mathbf{a} とその経路の総和 $E(\mathbf{a})$ を示す模式図.

ここでは距離を問題にすると、道順を配位空間とし、距離をエネルギーと見立てることによって、統計力学の手法である Monte Carlo 法によく似た**アニーリング法 (simulated annealing)** で最適化がおこなえる。これは格子欠陥を多く含んだ金属を高温へ上げて欠陥を掃き出し柔らかくする熱処理 (焼きなまし, annealing) からの類推で名付けられた。

対象となる経路の長さの合計は

$$E(\mathbf{a}) = \sum_{i=1..N} \|\mathbf{r}[\mathbf{a}_i] - \mathbf{r}[\mathbf{a}_{i+1}]\| \quad (6.1)$$

となる。ここで \mathbf{a} はそれぞれの街の順番を示している。 \mathbf{r} はそれぞれ街の座標で $\|\mathbf{r}\|$ で距離を求める。するとこの関数は模式的に図 6.1 のようになると考えられる。すべての配置について $E(\mathbf{a})$ を求めれば、最小値が求まる。あるいは初期の配置を

$$\mathbf{a} = [1, 2, 3, \dots, N, 1] \quad (6.2)$$

として、一定の手順で変更 $\delta\mathbf{a}$ を加え、 $E(\mathbf{a})$ が下がった場合にその配置を採用するという方法をとることもできる。しかし、この方法は局所的な極小値に落ちてしまう。最小値を探すには時として坂を駆け登る必要がある。simulated annealing のアルゴリズムは以下のとおり。

- 1) 配置 \mathbf{a} を仮定し $E(\mathbf{a})$ を求める.
- 2) \mathbf{a} からすこし違った配置 $\mathbf{a} + \delta\mathbf{a}$ を作る.
- 3) $\Delta E = E(\mathbf{a} + \delta\mathbf{a}) - E(\mathbf{a})$ を求める.
- 4) $\Delta E < 0$ なら新たな配置を採用する.
- 5) $\Delta E > 0$ なら新たな配置を $\exp(-\Delta E/T)$ の確率で受け入れる.
- 6) 手順 2 以下を適当な回数繰り返す.

ここで T は温度から類推される制御パラメータで、十分大きい場合はすべての状態が採用される。一方、 T を下げるにつれて採用される試行が少なくなり、徐々に状態が凍結されていく。 $\mathbf{a} + \delta\mathbf{a}$ の生成方法と T の下げ方をうまく取れば最小値に近い状態が確率的に高く出現し、最小値かそれに近い状態へ落ち着く。 $N = 16$ の場合の (a) 初期配置, (b) 単純に $E(\mathbf{a})$ が下がった場合だけを採用した結果, および (c) simulated annealing の計算結果を図 6.2 に示した。

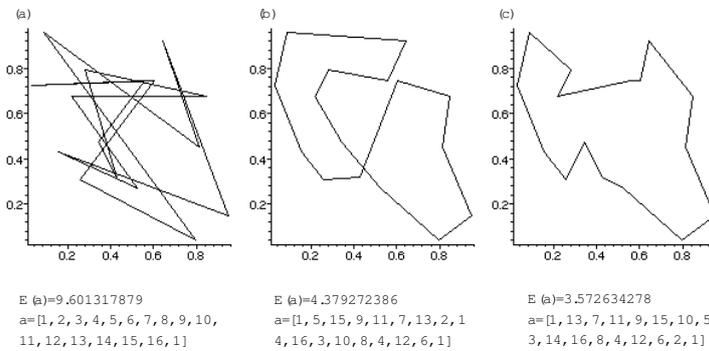


図 6.2 ランダムに生成した 16 個の街に適用した巡回セールスマン問題の (a) 初期配置, (b) 単純に $E(\mathbf{a})$ が下がった場合だけを採用した結果, および (c) simulated annealing の計算結果.

6.2 コイン投げ

コインを投げて、それを記録する。10 回投げた後に

$$\mathbf{a} = [\text{表}, \text{裏}, \text{表}, \text{裏}, \text{表}, \text{裏}, \text{裏}, \text{裏}, \text{表}, \text{裏}] \quad (6.3)$$

となったとする。このとき表のでた数は4である。これを何度も繰り返した後、表のでた回数ごとにプロットするとどのようになるであろうか。

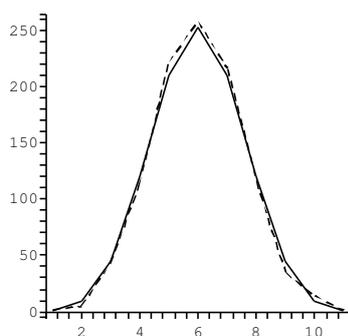


図 6.3 コイン投げの計算結果. 破線は 1024 回試行したシミュレーション結果. 実線は二項係数による理論値.

図 6.3 にシミュレーション結果を示した。これは、ある位相空間の一点が出現する場合の数を数えることに相当する。10 枚のコインのうち 4 枚が表になる場合の数は次のように計算できる。今、10 枚のコインに順番をつけて、4 枚が表だったとすると、先程の例ではその順番は、

$$\mathbf{a} = [1, 3, 5, 9] \quad (6.4)$$

となる。表となる順番の選び方は、最初は 1 から 10 までの 10 通り。二つ目は、すでに 10 から一つ減っているので残り 9 通り。これを 4 回繰り返せば、全ての場合の数は

$$W = 10 \cdot 9 \cdot 8 \cdot 7 \quad (6.5)$$

となる。ところがこれでは数え過ぎで、おなじ組み合わせを違った順番で取りだしたものを別々に勘定している。先程の例では

$$\mathbf{a} = [3, 1, 9, 5] \quad (6.6)$$

なども重複して数えていることになる。そこで4個の数の並べ方の数は

$$4 \cdot 3 \cdot 2 \cdot 1 \quad (6.7)$$

通りある。したがってこれで割っておく必要がある。一般に N 個から n 個とる取りだし方の数は、

$$W = \frac{N \cdot (N-1) \cdot (N-2) \cdots (N-n+1)}{n \cdot (n-1) \cdots 1} = \frac{N!}{(N-n)! n!} \quad (6.8)$$

である。すべての場合の数は 2^N であるので、 N 回投げて n 枚が表を向く確率は

$$P_N(n) = \frac{1}{2^N} {}_N C_n = \frac{1}{2^N} \frac{N!}{n!(N-n)!} \quad (6.9)$$

で与えられる。 ${}_N C_n$ は **2項係数 (Binomial coefficient)** と呼ばれる。

N の数を大きくすると、よりするどいピークを持った分布となる。つまり、コインを投げた場合に表がでる場合の数が丁度半分ぐらいのところを最も頻繁となり、そこからずれた場合の数は極端に小さくなる。試行回数 N が大きくなると

$$P_N(n) \simeq \sqrt{\frac{2}{\pi N}} \exp\left(-2\left(\frac{n-N/2}{\sqrt{N}}\right)^2\right) \quad (6.10)$$

という**正規分布 (normal distribution)** あるいは**ガウス (Gauss) 分布** に近づいていくことがよく知られている。この時の平均値からのずれのことを一般に**揺らぎ (fluctuation)** というが、試行回数 N の平方根に逆比例して揺らぎが小さくなっていく。このような自由度の大きな極限でのガウス型分布への漸近現象は**中心極限定理 (central limit theorem)** と呼ばれる。

コイン投げの例で揺らぎの低下とともにもう一つ重要な点は、 2^N 通りの可能なすべての微視的状态が完全に同じ確率 $1/2^N$ で現れるという、**等重率の原理 (principle of equal a priori probability)** である。系がエネルギーをもつ場合には、この等重率の原理は、「同一の全エネルギーを持つ系の全ての微視的状态は同じ確率で現れる」となる。

6.3 2 元 固 溶 体

A,B の 2 原子種で構成された合金を考えてみよう。単純なモデルとして図 6.4 のように 2 次元正方格子上に 2 種類の原子をばらまいたとする。配位空間は AB 原子を表す二次元の行列と取るのが自然であろう。

$$\mathbf{a} = \begin{bmatrix} A & A & \cdots & B \\ B & A & \cdots & A \\ \vdots & \vdots & \ddots & \vdots \\ A & B & \cdots & B \end{bmatrix} \quad (6.11)$$

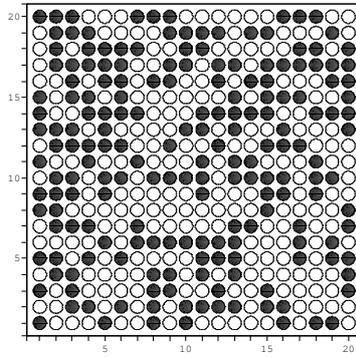


図 6.4 AB 合金のモデル。

しかし、配位空間はそれぞれの列（あるいは行）を一列につないだ並びとみなして、

$$\mathbf{a} = [ABAAAA \cdots] \quad (6.12)$$

などとも見なせ、先程のコイン投げと同じと考えることも可能である。ただし、コイン投げでは表裏は $1/2$ の確率であったが、A,B 原子の数 N_A, N_B が $1:1$ でない場合も考える必要がある。この場合は、 $N(= N_A + N_B)$ 個の球から赤

玉, 白玉それぞれ N_A, N_B 個を取り出す場合と同じ計算になる. まず, すべての球に通し番号 $1, 2, \dots, N$ をつけると, その並べ方は $N!$. つぎに番号を消すと, このうち区別できない並べ方が現れる. その重複の数は同色の球 (番号付き) の間の順列の数 $N_A!$ と $N_B!$ との積となる. したがって

$$W = \frac{N!}{N_A!N_B!} \quad (6.13)$$

となる. この場合のエネルギーは, 配置に依存して変わり, 等エネルギー変化ではないので注意が必要.

このように配位空間の取り方は一意的ではない. 問題の性質によって適当に変える必要が出てくる. 逆に一見違った問題も, よく似た計算法で解くことができる.

演習問題

課題 1 (アインシュタイン結晶) 解答略

結晶の熱振動を調和振動子とみなしたアインシュタイン結晶の配位空間を考えよう. 振動数 ν を持つ一つの調和振動子のエネルギー準位は

$$E_n = \left(n + \frac{1}{2}\right) h\nu \quad (6.14)$$

で与えられる. 全体が N 個の振動子からなるほとんど独立な系が, 全エネルギー

$$E = \frac{N}{2} h\nu + Mh\nu \quad (6.15)$$

である場合, その状態数 W を求めよ.

ヒント: i 番目の振動子の量子数を n_i とすれば, 系の全エネルギーが (6.15) 式となるのは,

$$n_1 + n_2 + \dots + n_N = M \quad (6.16)$$

を意味している. 図 6.5 を参照して, 前述の赤玉, 白玉の場合の数と同じ計算法でいけるはず.

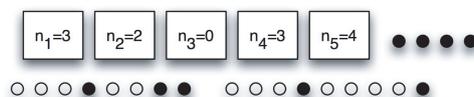


図 6.5 調和振動子の箱へ球をつめこむ場合の数の数え方.

課題 2 (さいころ投げ)

図 6.3 の計算機シミュレーションプログラムを作成し、実行せよ。 N を増やすことによってどのようにような変化があるかを観察せよ。また、2 項分布およびガウス分布と比較せよ。

課題 3 (AB 原子の描画)

図 6.4 のような AB 原子の分布を描画するプログラムを作成せよ。

解答例

```

2 >restart;
  with(plots)
  toss:=rand(0..1):
  N:=10:TRIAL:=1024:
  total:=[seq(0,i=0..N)]:
  for i from 1 to TRIAL do
    up:=0:
    for j from 1 to N do
      if toss()=1 then
        up:=up+1;
      end if;
    end do;
    total[up+1]:=total[up+1]+1;
  end do:

>l1:=listplot(total,linestyle=DASH):
  l2:=listplot([seq(binomial(N,i),i=0..N)]):
  display(l1,l2);

```

```
3 toss:=random(0..1);
  lWhite:=[]:lBlack:=[]:
  for i from 1 to 20 do
  for j from 1 to 20 do
  if toss()=1 then
    lBlack:=[op(lBlack),[i,j]];
  else
    lWhite:=[op(lWhite),[i,j]];
  end if;
  end do;
  end do;
with(plottools):with(plots):
pBlack:=[seq(pieslice(lBlack[i],0.4,0..2*Pi,color=BLACK),
  i=1..nops(lBlack))]:
pWhite:=[seq(pieslice(lWhite[i],0.4,0..2*Pi,color=WHITE),
  i=1..nops(lWhite))]:
あるいは
pBlack:=pointplot(lBlack,symbol=circle,symbolsize=20,color=black):
pWhite:=pointplot(lWhite,symbol=circle,symbolsize=20,color=red):
display(pBlack,pWhite);
```