

化学振動のパターンを見つけよう

1 背景

いくつかの化学物質を混合した溶液を攪拌して置いておくと、何段階かの化学変化が周期的に起きて、溶液の色が周期的に変化することがある。このような現象を化学振動と呼び、最もよく知られた例は臭素酸・マロン酸・セリウムイオンなどの混合溶液に見られるペロウソフ・ジャボチンスキー反応 (BZ 反応と略す) である。浅いシャーレ中で溶液を攪拌せずに BZ 反応を行なわせると、色が同心円上に変化してゆくダイナミックなパターンが観察できる。インターネットの動画サイトでもたくさんの実験例を見ることができる。ノーベル化学賞を受賞したイリヤ・プリゴジーヌはこのようなパターンの発生を散逸構造と呼んだ。今回の SuperCon の課題はこの化学振動によって発生するパターンにヒントを得たものである。



"Belousov Zhabotinsky reaction" by Stephen Morris (<https://www.flickr.com/photos/nonlin/3572095252>)
CC BY 2.0)

2 循環的セルオートマトン

浅いシャーレで見られる化学振動パターンによく似たパターンを生成するコンピュータ・モデルとして、Cyclic Cellular Automaton (循環的セルオートマトン。以下 CCA と略す) というものが知られている。これは化学反応をそのままモデル化したわけではなく、はるかに単純なものだが、発生するパターンは化学振動で作られるものとよく似ている。セルオートマトンとは空間をセル (マス目) に分割して、各セルが特定の規則にしたがっていくつかの状態の間で移り変わっていくコンピュータ・モデルの総称である。最も有名なのは J.H.Conway が提案した Game of Life (ライフゲーム) というもので、これは実際にプログラムを組んだことがある人も多いのではないだろうか。CCA はそのセルオートマトンの特別な場合である。CCA にもさまざまなバリエーションがあるが、今回の課題では以下の規則によって変化してゆく特定の CCA(5 状態 25 近傍で閾値 6 の

CCA と呼ぶ) を使う。

1. セルの状態: 平面 (碁盤を思い浮かべて盤面と呼ぼう) を正方形のセル (マス目) に分割し、各セルはいくつかの状態になれるとする。この状態が、BZ 反応で言えば各セルが化学反応のどの段階にいるかを表す。今回使うモデルでは状態は 5 種類であり、0,1,2,3,4 と番号をつける。化学反応が決まった順番で進行することに対応して、状態には順番が決められており、 $0 \rightarrow 1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 0$ と順番に状態が変化を繰り返す。最後の状態 4 の次に最初の状態 0 に戻るのが循環的と呼ばれる理由で、これは化学反応が周期的に繰り返すことに対応する
2. 状態変化の規則: セルが次の状態に変化するのには、周囲に十分な数の「次の状態のセル」がある時に限る。今回は周囲とは注目するセルを中心とした 5×5 の 25 セルを意味し、十分な数 (閾値という) は 6 であるとする。たとえば、注目するセルが状態 0 だとすると、周囲 25 セル中に状態 1 のセルが 6 個以上あるとき、注目するセルの状態は 1 に変化する。そうでない場合には注目するセルの状態は変化しない。
3. ステップの定義: セルの状態変化はすべてのセルで同時に起きる。これを CCA の 1 ステップと呼ぶ。つまり、全てのセルについて周囲 25 セルの状態を同時に確認し、次の状態のセルが閾値以上のセルは状態をいっせいに変化させ、閾値以下のセルについては変化させずにそのままにしておく。これで 1 ステップである。
4. 盤面の境界の処理: 今回のモデルでは盤面の境界の影響をなくするために、盤面の右端と左端、上端と下端がつながった周期的境界条件を採用する。つまり、盤面のいちばん右端のセルの右隣には盤面のいちばん左端のセルがあるとみなす。これは無限に広い平面に同じ盤面が繰り返し敷き詰められていると考えてもいいし、平面ではなくドーナツの表面のようなものを想定していると考えてもよい。

規則の実例を見てみよう。たとえば、以下の例では真ん中のセルの状態が 4 なのに対して、周囲 25 セルに 0 が 7 個あり、閾値以上なので、真ん中のセルの状態は 0 に変化する (1 が 9 個あって 0 より多いが、4 の次は 0 なので、1 の数は関係ない)。

0	1	2	1	2	→					
3	4	0	0	1						
1	1	4	0	0			0			
0	1	0	2	2						
4	1	3	1	1						

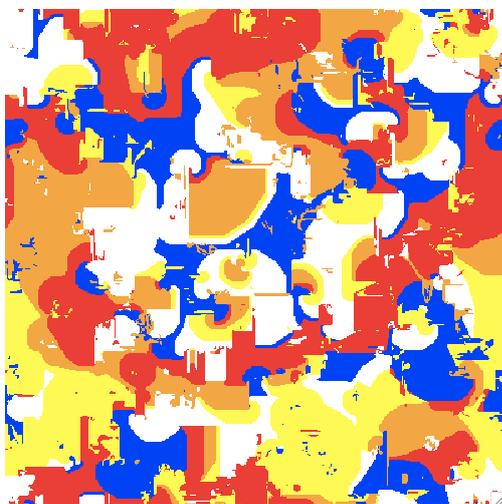
いっぽう、次の例では真ん中のセルが 1 なのに対して、周囲 25 セルに 2 が 4 個しかないので、真ん中のセルの状態は変化しない

0	1	2	1	2
3	4	0	0	1
1	1	1	0	0
0	1	0	2	2
4	1	3	1	1

→

		1		

最初に盤面の各セルの状態をランダムに決めて、上の規則に従って CCA を 1 ステップずつ進めていくと、最初はランダムだった平面にパターンができて変化してゆくのが観察できる。図は 400×400 セルの場合のある時刻での盤面である。状態ごとにセルの色を変えて表示している。少し化学反応らしくない部分もあるが、がまんしておこう。



3 問題

設定: (a) シャーレ内の化学振動の様子を収めたビデオ (b) ある瞬間にシャーレの一部を撮影した写真、のふたつが与えられたとき、(b) の写真は (a) のビデオのどの時刻のどの場所を撮影したものかを判定しなさい。ただし、写真はビデオより解像度が低く、少しぼんやりとしか写っていないものとする

CCA を用いた実際の問題: (a) CCA の最初の盤面 (b) CCA をあるステップ数だけ進めた盤面的一部分 (写真と呼ぼう)、のふたつが与えられる。(a) の盤面から CCA の規則に従って 1 ステップずつ進めていくとき、(b) に一致する盤面は何ステップ目のどの位置に出現するかを見出しなさい。ただし、(b) は本来の盤面に対して解像度を下げている

(b) について説明しよう。写真は CCA のあるステップでの盤面の一部であるが、実際には以下の手順によって解像度を下げたものである。

1. 元の盤面を端から順に 4×4 の 16 セルからなる大きなマス目 (画素と呼ぼう) に分割する (画素の総数は元のセル数の $1/16$ になる)
2. 画素内の 16 セルの各状態の数を数え、最も多い状態をその画素の状態とする (多数決則)
3. 画素内で最多となる状態が複数あるとき (最大で四つの状態が同時に最多になりうる) は、それらの中から乱数によって選ばれた状態を画素の状態とする。たとえば、16 セル中に 0 と 1 と 2 が 5 個ずつあるときには、この三つのうちからひとつを乱数で選ぶ。もし、乱数によって 0 が選ばれたとすれば、画素の状態は 0 である
4. このように作られた解像度の低い画像の一部を取り出したものが写真である

たとえば、元の盤面で

0	1	2	1
3	4	0	0
1	1	1	0
0	1	1	2

の 16 セルなら、1 が 7 個で最多なので、画素は 1 になる。

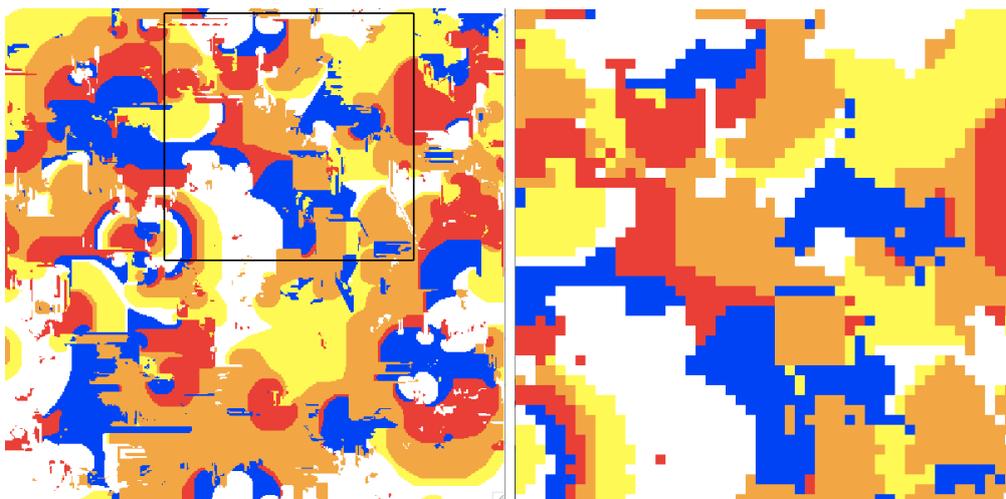
いっぽう

0	1	3	3
3	4	0	0
1	1	1	0
0	1	3	3

の 16 セルなら、0,1,3 がそれぞれ 5 個で最多なので、画素は 0 または 1 ま

たは 3 のいずれかが乱数によって選ばれる。

次の左図はあるステップでの盤面であり、黒枠の部分の解像度を落とした写真が右図である。比較してみてください。



4 詳細

1. 盤面は $N \times N$ セルの正方形である。最終課題では $N = 2000$ とする。
2. 盤の横方向を x 方向、縦方向を y 方向とし、盤面上のセルの位置は (x, y) で表す。最初のセルの位置が $(0, 0)$ である。
3. 写真のサイズは全盤面の $1/4$ サイズ (一辺は半分) の正方形である。したがって、写真の画素数は $(N/8) \times (N/8)$ となる。
4. 最大のステップ数は 5 万ステップである。5 万ステップ以内に写真に一致する盤面が複数回現れるときは、最初に現れたステップ数が正解となる。なお、与えられた初期盤面から 1 ステップ進めた結果を「第 1 ステップ」とする。
5. 写真は CCA の 100 ステップ毎に撮られているものとする。したがって、答のステップ数は必ず 100 の倍数である。
6. 解像度を下げる操作では盤面の端から分割していくので、写真の位置は x, y とも必ず 4 の倍数である。

5 勝利条件

1. $N = 2000$ の問題を 3 題解く。なお、計算時間の上限は 3 題合計で 10 分とする。第一問から順に計算していったら、計算時間の上限に達した時点で、計算途中であっても終了とする
2. まず、正解数によって順位をつけ、正解数が同じであれば正解を出した計算の総実行時間が短いほど上位とする (並列化を使うので CPU 時間ではなく、経過時間である)
3. なお、秒単位まで実行時間が一致する場合には、さらに問題数を増やすことがある

6 ヘッダーファイル

配布するコンテスト専用ヘッダーファイル `sc15.h` には以下のように定数・配列・関数が定義されている。

```
#define MAXCELL 2000
void input(int n);
void output();
int cell[MAXCELL][MAXCELL];
int pixel[MAXCELL/8][MAXCELL/8];
void input(int n)
{ int i,j;
for(i=0;i<n;++i){for(j=0;j<n;++j){scanf("%d",&cell[i][j]);}}
```

```
for(i=0;i<n/8;++i){for(j=0;j<n/8;++j){scanf("%d",&pixel[i][j]);}}
void output(int step, int x, int y)
{ printf("%d %d %d\n",step,x,y);}
```

このヘッダーファイルをプログラム冒頭で

```
#include<stdio.h>
#include<stdlib.h>
#include<math.h>
#include "sc15.h"
```

などのようにインクルードして使う。

1. 関数 input は必ずプログラム中で最初の実行文とすること。これ以前に実行文を書いてはいけない。input を実行することにより、問題の初期配置と写真のデータがそれぞれ cell と pixel という配列に標準入力から読み込まれる。データの読み込みには必ずこの input 関数を使うこと。input の引数 n は盤面の一辺あたりのセルの数であり、審査本番では 2000 に固定される。
2. 配列 cell,pixel とともに 2 次元配列で、サイズは定数 MAXCELL を使って MAXCELL*MAXCELL と (MAXCELL/8)*(MAXCELL/8) に固定されている。なお、2 次元配列はいずれも [y 座標][x 座標] の順である。
3. MAXCELL は 2000(盤面の一辺あたり最大のセル数) である。
4. 関数 output を呼ぶことにより、結果が出力される。引数は答の (ステップ数、x 座標、y 座標) の順であり、すべて整数である。output 関数以外の方法で出力してはならない。

なお、プログラム提出後の審査には出題側で用意したヘッダーファイル(本質的には上と同じだが、審査用のコードが含まれる)を使うので、ヘッダーファイルを独自に変更しても反映されない。プログラム作成時にヘッダーファイルを書き換えて使うのは自由である。

7 ポイント

この課題で作成するプログラムは大きくふたつの部分に分けられる。ひとつは CCA を実行する部分、もうひとつはパターンを判定する部分である。前者に関しては、SX-ACE のベクトル化と並列化を十分に活用した高速のプログラムを書くことが必須となる。この部分はベクトル化した場合としない場合とで計算速度が 10 倍以上変わると思われるので、高速化できなければ計算時間が足りなくなるだろう。高速プログラムを目指して頑張してほしい。いっぽう、後者ももちろんベクトル化を活用することが重要だが、それ以上にパターン・マッチングをいかにうまく「さぼる」かが鍵となるだろう。今回はまったく同じパターンを見つける単純なパターンマッチングではなく、解像度の低い写真からそれに合う元のパターンを見つける必要があるので、工夫が必要となる。具体

的には、複数の可能性の中から乱数で決められた画素をどのように処理するかを考えて欲しい(どの画素が乱数で決められたものなのかは、見ただけではわからない)。なお、高速化のこつはプログラム中で時間がかかる部分をうまくベクトル化することである。時間がかからない部分をどれほど工夫してもプログラム全体は高速化されない