

# StarLogoを用いた自己形成のシミュレーション

南 雲 秀 雄

新潟青陵大学福祉心理学科

## Simulating Self-Assembly Using StarLogo

Hideo Nagumo

NIIGATA SEIRYO UNIVERSITY  
DEPARTMENT OF SOCIAL WELFARE AND PSYCHOLOGY

### Abstract

We have simulated the growth of a snowflake and that of a flagellum using StarLogo, which is a programmable modeling environment for exploring the workings of decentralized systems. Although both of them are cases of self-assembly, the models we used for them are quite different. For a snowflake, we used the model in which new particles stick on the surface of the already assembled cluster. On the other hand, for a flagellum, we used the model in which new particles are generated inside the cluster and they push the surface of the cluster outwards. The method of simulation in both cases is to use cellular automata, and thus the simulation steps are processed in a parallel fashion.

### Key words

self-assembly, cellular automaton, simulation, StarLogo

### 要 旨

本研究では、分散システムのモデル化環境であるStarLogoを用いて、雪の結晶の成長と、鞭毛成長のシミュレーションを行った。いずれも自己形成のシミュレーションであるが、雪結晶の場合は、新たな粒子が外部から付着してパターンが成長していくというモデルを使い、鞭毛成長の場合は、新たな粒子は内部で発生し、そのためにパターンの外縁が外部に拡大するという反対のモデルを使った。シミュレーションの手法はいずれの場合も、セルオートマトンを用い、並列的な処理によってステップを進めた。

### キーワード

自己形成、セルオートマトン、シミュレーション、スター・ロゴ

## 1. はじめに

近年、ナノスケールでの自己形成 (self-assembly、self-organization) が新たな研究分野として注目を集めている<sup>1)</sup>。ここでの自己形成とは、小さな材料が自動的に組み合わせあって、整然とした構造物や複雑な構造物を作り上げることを意味する。

自己形成は様々な分野で応用が考えられている。化学の世界では、小さな分子のブロックを組み合わせることで正四面体や正六面体、おわん状の幾何学分子を作る実験が行われている<sup>2)</sup>が、その応用としては、新薬の製造、環境に負荷の少ないプラント、微量な物質を高感度で検出できるセンサーなどを作ることができると期待されている。

またコンピュータの世界では、従来のIC製造技術に代わる、次世代のIC製造技術として期待されている。ICの微細化は年とともに指数関数的に進んでいるが、ICを製造する設備に掛かる費用も指数関数的に増大している。そのため企業がどんなに投資を増大させても新たなIC製造施設を作れなくなる時が来ると見られていて、その時にはおそらく、精密に設計された素子が結合しながら、装置を作り上げる自己形成によるIC製造が現実的な選択肢となると考えられている<sup>1)</sup>。

ナノスケールでの自己形成のモデルは自然界に求められることが多い。例えば雪の結晶が成長する過程や、生きている微生物が成長する過程などである。微生物の場合には、例えばボーイング777ジェット機よりも何桁も複雑な構造を持っていながら、持続的に自分自身を構築する自然の工程を有している<sup>1)</sup>。

自己形成が重要性を増すのに伴い、コンピュータによるそのシミュレーションも重要となっている。本研究では、StarLogoというソフトウェアを用いて自己形成のシミュレーションを行い、このようなシミュレーションに対する実用性を検証する。シミュレーションを行うのは、雪の結晶成長と鞭毛の成長である。雪結晶の場合は、新たな粒子が外部から付着してパターンが成長していくというモデルを使い、鞭毛成長の場合は、新たな粒子は内部で発生し、そのためにパターンの外縁が

外部に拡大するという反対のモデルを使った。シミュレーションの手法はいずれの場合も、セルオートマトンを用い、並列的な処理によってステップを進めた。

## 2. StarLogo

StarLogoは、分散システム (中央制御が無い状態で組織化されるシステム) のプログラム可能なモデル化環境である。このソフトウェアを使うと、鳥の群れや、交通渋滞、群生 (コロニー)、市場経済など、現実にかかる多くの現象をモデル化し、それらについての識見を得ることができる。

StarLogoでは、何千というタートルと呼ばれる粒を並列的に制御し、スクリーン上に模様を描いたり、アニメーションとして動かしたりできる。また、何千というパッチと呼ばれる環境情報を持つ粒が画面上に配置されていて、それらをプログラムすることによりタートルの環境を複雑に形作ることができる。タートルの振る舞いは、近傍のパッチの種類により変化する<sup>3,6)</sup>。

図1にWindows環境でのStarLogoのウィンドウを示す。StarLogoのウィンドウは二つに分かれている。図の右側のウィンドウは、StarLogoウィンドウで、その中にタートルやパッチが表示されるグラフィック・キャンバスと、様々なコントロールが配置されるインターフェイス・ウィンドウが置かれている。グラフィック・キャンバスには、縦101個、横101個の正方格子に並んだ柵にパッチの粒が配置されており、この上にタートルの粒を置いて、それらの動きや色を制御する。左側のウィンドウは、コントロール・センターのウィンドウで、タートルに直接命令を与えるためのコマンドセンターのボックスと、プロシージャー・ペインと呼ばれるプログラムを記述するためのボックスが配置されている。コントロール・センターは、タートルに対する命令を書くためのものと、観測者として全体的な動作を制御するためのものがあり、これらはボタンで切り替えるようになっている<sup>5)</sup>。

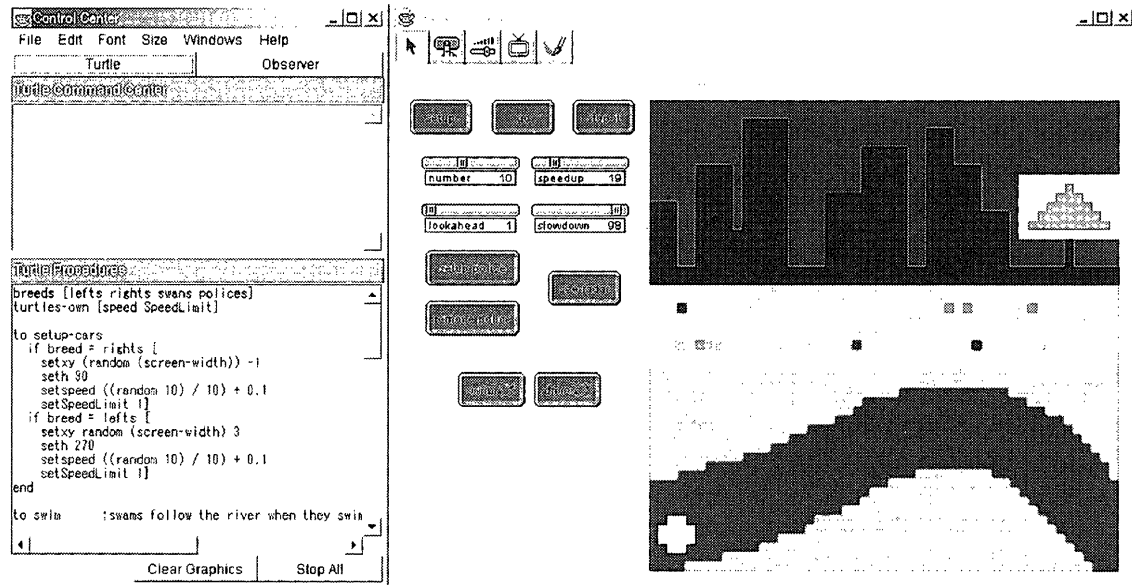


図1 Star Logoのウィンドウ

StarLogoはマサチューセッツ工科大学のメディア・ラボラトリーで、主に教育用に開発されたソフトウェアである。そのためStarLogoのプログラミングは高校生レベルでもできるように単純化されている。StarLogoは6)のURLからダウンロードすることができる。

### 3. 雪結晶の成長シミュレーション

#### 3.1 雪結晶のセルオートマトンによるモデル化

雪の結晶の多くが平面的な六角形の形状をしていて、これらは、角板、樹枝状六花、広幅六花、扇状六花に分類されている。その他、六角形でない雪の結晶も存在し、角柱、針の2種類に分けられている。本研究では、六角形の形状の雪結晶の類似パターンを、セルオートマトンを用いて生成する。

セルオートマトンとは、空間を一様に格子分割し、各格子点に有限の状態を持つセルを配置したものである。空間が1次元の場合は線上にセルが並び、2次元の場合は平面状にセルが並ぶというように、 $n$ 次元のセルオートマトンを定義できる。各セルは、有限な状態のうち一つを初期状態とし、時間の進行とともに状態を遷移する。ある時刻の自身の状態と、そのときの近傍のセルからの入力によって、次の時刻のセルの状態が一意的に決定

される。次の時刻でのセルの状態を決める規則を遷移関数と呼ぶが、全セルが同じ遷移関数をもつ。そして、状態遷移は全セルが同期して並列に行われる。<sup>7,8)</sup>

セルオートマトンを用いた雪結晶の類似パターン生成は、Packard<sup>9)</sup>や轡田ら<sup>10)</sup>によって行われている。いずれの場合も、2次元セルオートマトンを用いており、雪の結晶が六方対称性をもつことから、2次元セルオートマトンとして一般的な正方格子上的ものではなく、図2に示すような、六角格子上的の2次元セルオートマトンを用いている。しかし、StarLogoの柁は正方格子上に並んでいるため、StarLogoでこのシミュレーションを行う場合には、図3に示すように、濃く塗られた柁目だけを使うことになる。

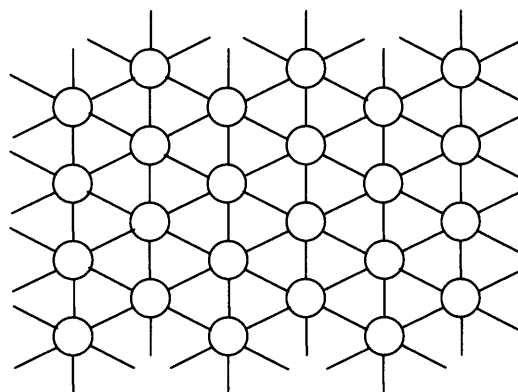


図2 六角格子上のセルオートマトン

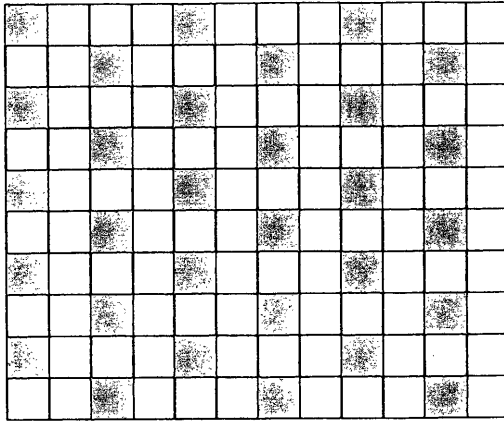


図3 StarLogoでの六角格子の作り方

### 3.2 Packardの外総和型モデル

Packardは雪の結晶を構成する粒子をセルオートマトンでモデル化するとき、(A)セルは水蒸気または氷の状態だけを持ち、(B)水蒸気から氷への遷移だけ起き、(C)初期様相では一個のセルだけが氷であるとした。セルの状態は、水蒸気と氷の2状態しかないので、各セルの状態は、水蒸気に対応する状態を「0」、氷に対応する状態を「1」としている。さらに、Packardは、あるセルの状態に影響を与えるのは、図4に示すような、自分自身

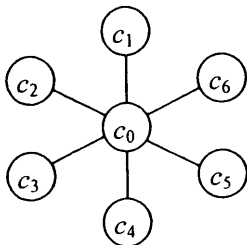


図4 近傍セル

( $c_0$ )の近傍の六つのセル、 $c_1$ 、 $c_2$ 、 $c_3$ 、 $c_4$ 、 $c_5$ 、 $c_6$ であり、この六つの近傍の中のある氷のセルの数によって、 $c_0$ が水蒸気の状態から氷に変化するかどうかが決まるというモデルを提案した。このように次の時刻のセル $c_0$ の状態が、セル $c_0$ の状態および周囲のセルの状態の総和によって決定されるセルオートマトンを外総和型 (outer totalistic) セルオートマトンと呼ぶ。外総和型セルオートマトンでは、セル $c_0$ の時刻 $t+1$ での状態 $s_0^{t+1}$ は、セル $c_0$ および $c_0$ の近傍の各セル $c_1$ 、 $c_2$ 、 $\dots$ 、 $c_6$ の時刻 $t$ での状態 $s_0^t$ 、 $s_1^t$ 、 $\dots$ 、 $s_6^t$ から遷移関数 $\delta$ によって式(1)のように決められる。上記(B)により、一旦氷の状態(「1」)になったセルは水蒸気の状態(「0」)に戻らないので、 $s_0^t=1$ のとき $s_0^{t+1}=1$ である。

$$s_0^{t+1} = \delta(s_0^t, s_1^t, \dots, s_6^t) = \begin{cases} f\left(\sum_{i \in \{1,2,\dots,6\}} s_i^t\right) & (s_0^t = 0) \\ 1 & (s_0^t = 1) \end{cases} \quad (1)$$

もし、 $f(0)=1$ ならば、初期様相の次の時刻には格子上のほとんどのセルが氷の状態になってしまうので、 $f(0)=0$ である。また、パターンが成長するためには、 $f(1)=1$ でなければならない。したがって、 $f(2)$ 、 $f(3)$ 、 $f(4)$ 、 $f(5)$ 、 $f(6)$ が0となるか1となるかの違いによって、このような関数 $f$ は $2^6=32$ 通りあり得る。

### 3.3 StarLogoによるPackardモデルのシミュレーション

図5に、 $f(0)=0$ 、 $f(1)=1$ 、 $f(2)=0$ 、 $f(3)=1$ 、 $f(4)=0$ 、 $f(5)=1$ 、 $f(6)=0$ という条件(水蒸気の状態は、その周りに奇数個の氷の状態が存在する場合に氷の状態に変化する)におけるStarLogoによるシミュレーションの24ステップ目の結果を示す。この図で、白い粒子は、水蒸気の状態を表し、灰色の粒子は、氷の粒子を表す。したがって、灰色の粒子のパターンが、雪結晶の類似パターンである。このパターンは角板と呼ばれる雪の結晶に類似している。Packardのモデルでは、このような角板の類似パターンを生成することはでき

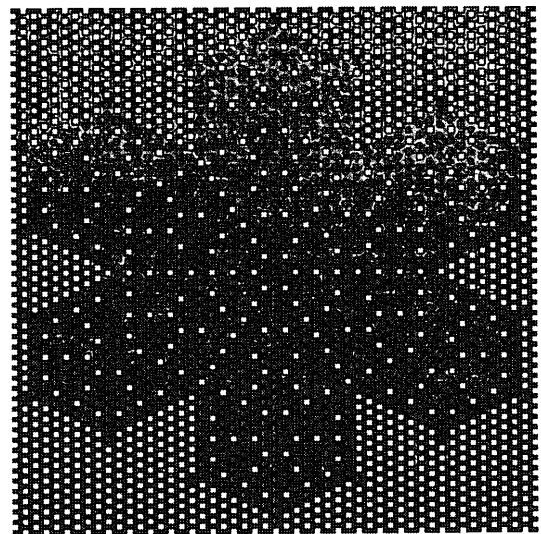


図5 PackardのモデルのStarLogoによるシミュレーション

るが、32通りのすべての遷移関数を試してみても、樹枝状六花、広幅六花、扇状六花などと類似したパターンを生成することはできない。

図6に、図5のパターンを生成するのに用いたStarLogoのコードを示す。StarLogoのコードで、観測者とし全体的な動作を制御するためのものは「Observer Procedure」と呼ばれ、個々のタートルをローカルに制御するための

ものは「Turtle Procedure」と呼ばれる。この図で、「Observer Procedure」の中の、「to setup」から「end」の間と、「Turtle Procedure」の中の、「to setup-particles」と「end」の間の部分は、図3に示した六角格子を作るためのコードであり、「Observer Procedure」の中の、「to freezing」から「end」の間と、「Turtle Procedure」の中の、「to freeze」と「end」の間および「to renew」と「end」の間の部分は、雪結晶のパ

```

Observer Procedures
to setup
  ct
  create-and-do 2550 [setup-particles]
  setoldice-at 0 0 1
  setc-at 0 0 113
end

to freezing
  ask-turtles [freeze]
  ask-turtles [renew]
end

Turtle Procedures
turtles-own [oldice newice]

to setup-particles
  setc 49
  setoldice 0
  setnewice 0
  setxy ((who div 50) * 2 - 50) ((who mod 50) * 2 - 50)
  if (xcor mod 4) = 2 [setxy xcor (ycor + 1)]
  if ycor = 48 [hatch [setxy xcor 50]]
end

to freeze
  if oldice = 0 [
    let [:number 0]
    set :number oldice + (oldice-at 0 2) + (oldice-at -2 1) +
      (oldice-at -2 -1) + (oldice-at 0 -2) +
      (oldice-at 2 -1) + (oldice-at 2 1)
    case :number [
      1 [setnewice 1]
      2 [setnewice 0]
      3 [setnewice 1]
      4 [setnewice 0]
      5 [setnewice 1]
      6 [setnewice 0]
    ]
  ]
end

to renew
  if (newice = 1) [
    setoldice newice
    setnewice 0
    setc 113
  ]
end

```

図6 Packardのパターンを生成するためのStarLogoコード

ターンを生成するためのコードである。

このように、StarLogoでは、セルオートマトンの動作を非常に簡潔に記述できるが、柵目が縦101、横101しかないため、雪結晶の成長シミュレーションは、24ステップ程度が限界である。また、雪結晶の成長シミュレーションでは、実際には正方格子状に並んだStarLogoの柵の中から、六角格子として使える柵だけを取り出して使っているため、使用可能な柵の4分の1しか利用してはず、無駄が多い。

轡田らは、水蒸気と氷の2状態ではなく、これらに加えて、氷になりつつある状態を導入して3状態とし、さらにセルオートマトンの外総和性も破棄して、それぞれの状態の個数だけでなく、その位置も考慮するというモデルを提案した。このモデルでは、セルオートマトンは、 $3^{2187}$ 通りという膨大な数の状態を持つことになったが、これにより轡田らは、樹枝状六花、幅広六花、扇状六花の類似パターン生成に成功した。

状態数の少ないモデルでは、ある程度のステップ数を経た後は、周期的な平面の埋め尽くしとなり、パターンは全体的に角板状に成長してしまうであろうことは容易に想像がつく。反対に、状態数が膨大であれば、周期性が現われるためにはより多くのステップ数を必要とし、上記の様々なパターンの生成が可能になる。しかし、膨大な数の組み合わせから、求めるパターンを生成する組み合わせを探し出すのは大変な作業である。

### 3.4 セルの位置を考慮したモデル

本研究では、Packardのモデルよりも若干複雑であるが、轡田らのモデルよりは単純なモデルを用いて、雪結晶の類似パターン生成を試みた。状態数は、Packardのモデルと同じく、水蒸気と氷の2状態とした。Packardは、遷移関数が近傍6セルの、氷であるものの総数によってのみ決まるというモデルを使ったが、本研究では、遷移関数は周辺のセル（近傍6セルよりも遠いセルを含む）の位置に依存するというモデルを使った。周辺のセルの位置に依存するというモデルを使えば、例えば、雪結晶の尖っている部分や、へこんでいる部

分では、水蒸気の状態から氷の状態へ変化しやすいというような条件を用いることができる。

図7に、雪結晶の尖っている部分で水蒸気から氷に変化するというモデルの例を示す。この図では、濃い灰色の円が氷の状態のセルで、薄い灰色の円が水蒸気の状態のセルである。このモデルでは、あるセルaの近傍に1個だけ氷のセルbがあり、aからbに向かう半直線から両側に60度の角度を持つ二つの半直線上でaから一つ目と二つ目のセルが両方とも水蒸気である時だけaの状態が水蒸気から氷に遷移するものとしている。

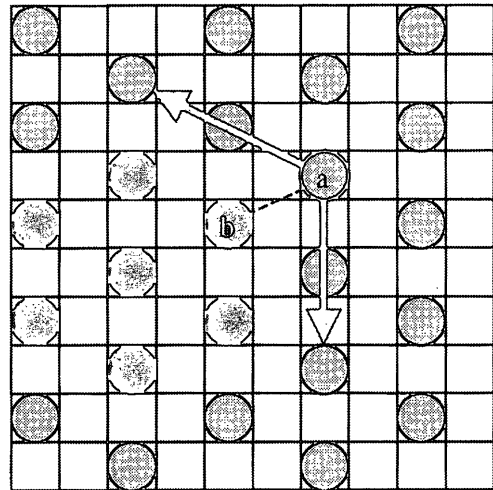


図7 セルの位置を考慮したモデル

このモデルを使ったときの、初期様相から16ステップ目までのパターン成長の様子を図8に示す。図8のパターンは図5のPackardモデルのパターンに比べると、6方向に伸びる軸の突起も顕著になり、樹枝状六花や幅広六花に近いものになっている。しかし、残念ながら、StarLogoでは24ステップ程度しかパターン生成を行えないので、それ以上ステップ数を増やしたときに、角板上に収束してしまうかどうかは分からない。

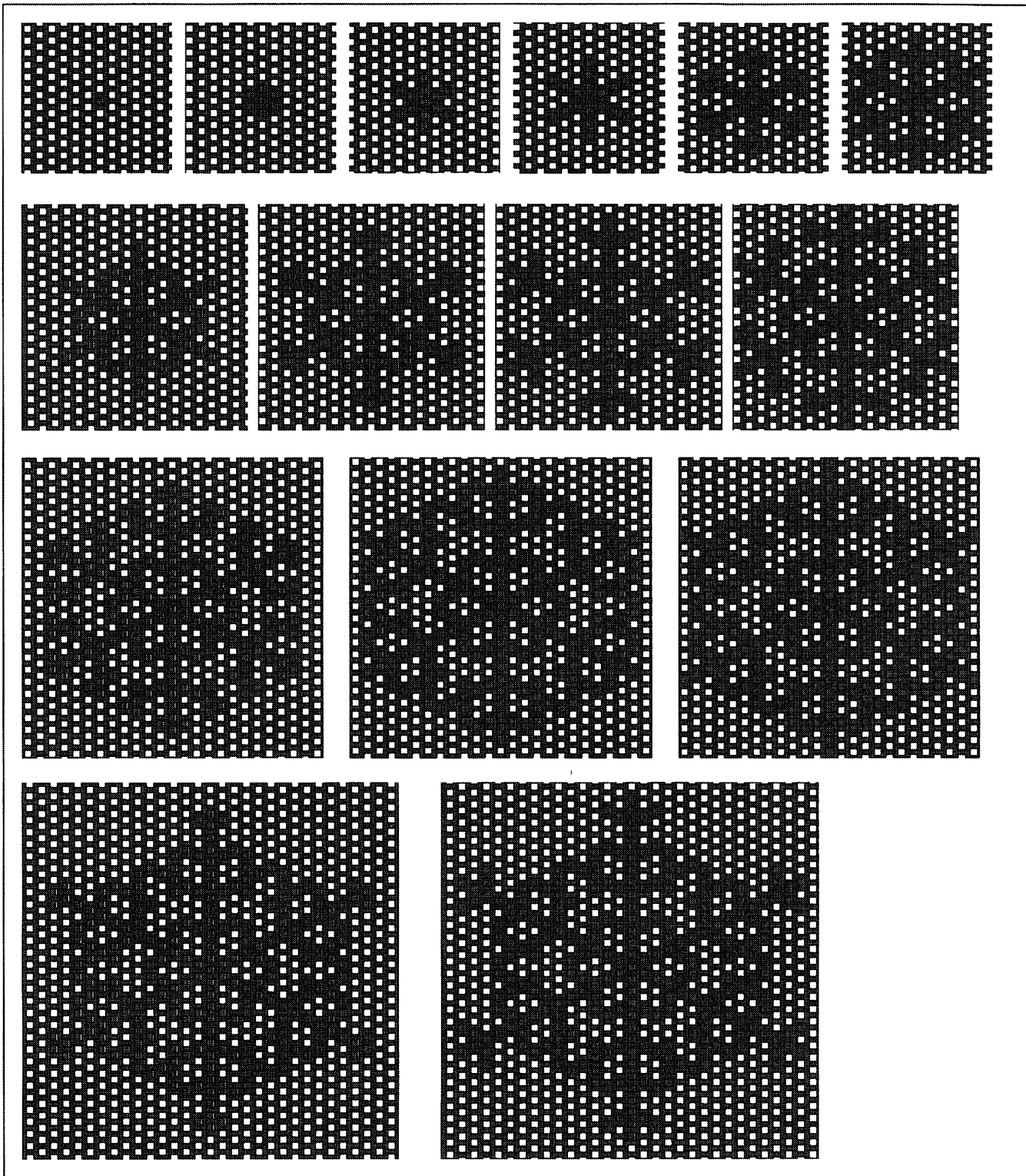


図8 セルの位置を考慮したモデルのシミュレーション

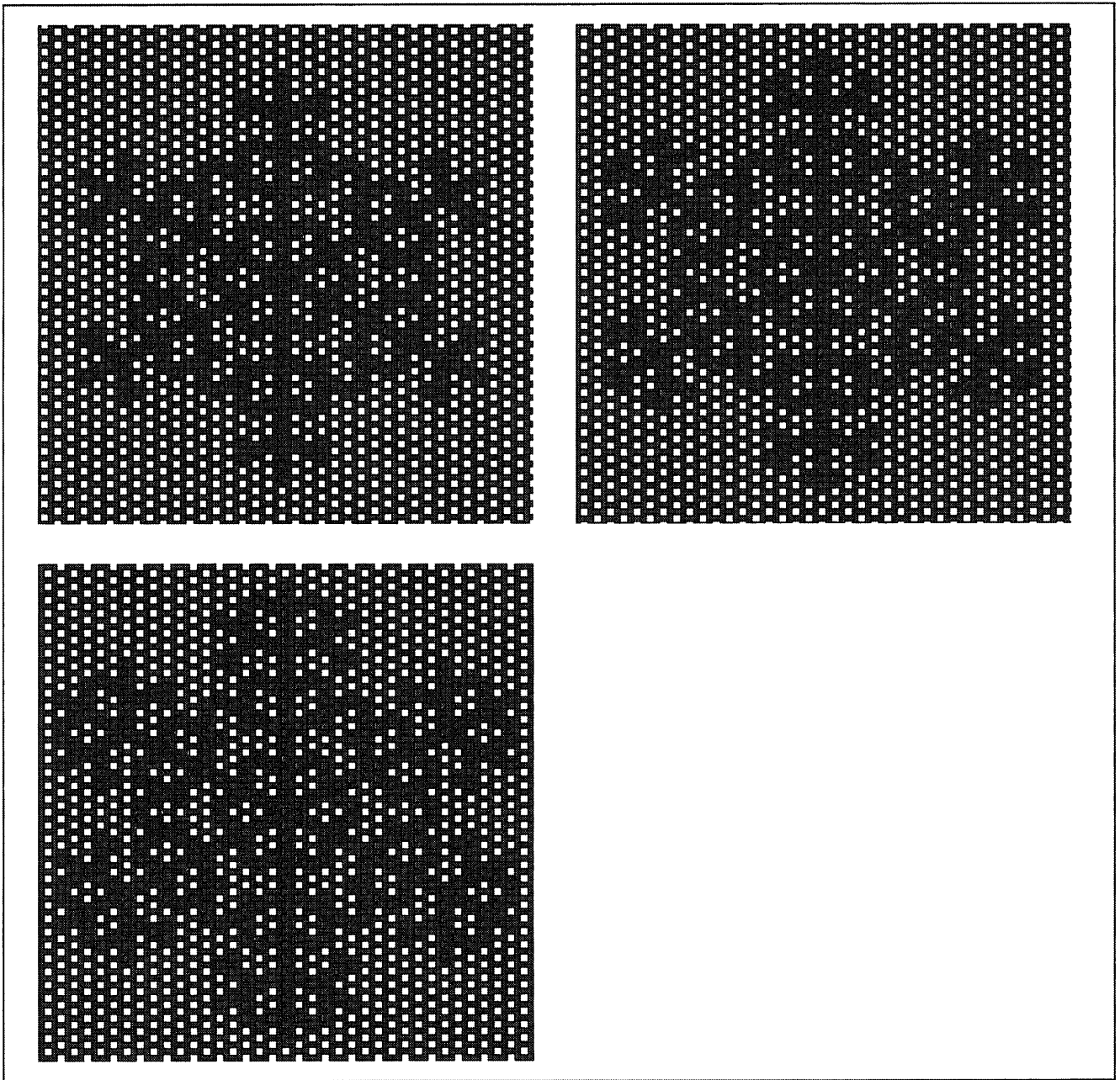


図8 セルの位置を考慮したモデルのシミュレーション (続き)



#### 4. 鞭毛の成長シミュレーション

##### 4.1 粒子クラスタの形状変形

雪結晶の成長シミュレーションの場合は、ある時点までに生成されたパターンの外縁に新たな粒子が付着しながらパターン自体の形状が変化するというモデルを使い、パターンの内部では粒子の位置は固定して移動がないものと仮定していた。これに対し、シャボン玉の泡がストローの先から出てくる現象や、鞭毛先端の成長点で細胞分裂が起こる現象では、すでに生成されたパターンの内部で粒子が発生したり、移動したりして、それによりパターンの形状が変化する。この章では、この様な内部に起因する形状変形のモデル化

を、鞭毛成長を例として取り扱う。

内部に起因する形状変形の分かりやすい例として、図9に示すような、平面上にすき間なく配置された多数の剛体球からなるクラスタ（粒子クラスタ）を、粒子のせり上がりや慣性運動が起こらないように準静的に障害物で押したときの形状変形がある。高井らは、この様な粒子クラスタの形状変形を、確率的状態遷移を含む4状態セルオートマトンでモデル化した<sup>11)</sup>。

粒子クラスタの形状変形では、障害物が、粒子クラスタの存在する空間に入り込むなどの理由により、粒子の移動が起こる。高井らは、セルオートマトンを用いてこの粒子の移動をモデル化するとき、1つのスペースに粒

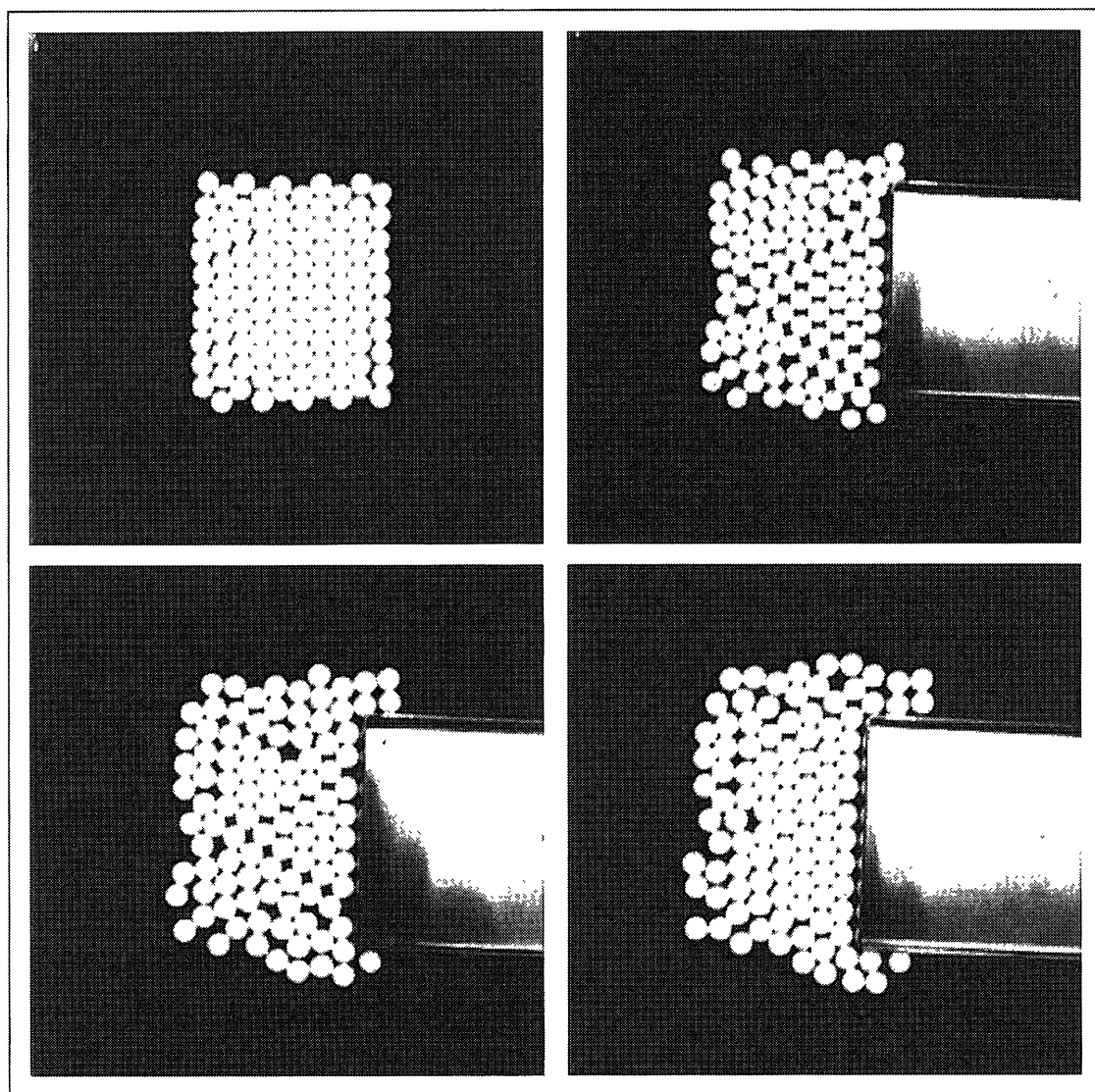


図9 剛体球クラスタを障害物で押したときの形状変形

子が2個入る過度的な状態を仮定した。この不安定な状態のセルをランダムに移動させ、粒子クラスタの外縁から1粒子分出たところで、1個の粒子として落ち着かせる。これにより、障害物によって押され、排除された粒子に変わって、確率的に選ばれたクラスタの外縁の1ヶ所に粒子が現れることになる。クラスタのある平面には、障害物が存在する領域や、何もない領域もあるので、各セルは次の四つの状態の中のいずれか一つの状態をとる。

状態0：粒子も障害物も存在しない

状態1：粒子が1個存在する

状態2：粒子が2個存在する

状態3：障害物が存在する

一般に、不安定セルは複数存在し、それらは重なり合わないよう移動しなければならない。このため高井らは、マーゴラス近傍と呼ばれる最隣接のセルブロックの中で状態遷移を発生させ、ステップごとにこのブロックの組み合わせを換えるという手法をとっている。剛体球が密に詰められると、それらは六角格子状になるが、高井らは、最隣接の三つのセルを一つのセルブロックとし、それをトリプレットと呼んだ。このトリプレットを用いて六角格子を重複なく分割する方法は、図10に示す $P_0, \dots, P_5$ の6パターン存在する。この6種類の近傍分割パターンを、状態遷移の1ステップごとに、 $P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow P_3 \rightarrow P_4 \rightarrow P_5 \rightarrow P_0 \dots$  というように巡回して適用する。トリプレットの中での1ステップの状態遷移は、次の規則に基づいて行われる。ここに無い組み合わせでは、状態遷移は起こらない。

- 状態2のセルが1個、状態0のセルが2個の場合、 $1/2$ の確率で状態0のセルのどちらかが状態1に変わり、状態2のセルは状態1に変わる(安定化する)。
- 状態2のセルが1個、状態1のセルが1個、状態0のセルが1個の場合、状態0のセルが状態1に変わり、状態2のセルが状態1に変わる(安定化する)。
- 状態2のセルが1個、状態1のセルが2個の場合、 $1/2$ の確立で状態1のセルのどちらかが状態2に変わり、状態2だ

ったセルは状態1に変わる

- 状態2のセルが2個で、状態0のセルが1個の場合、状態0のセルが状態1に変わり、 $1/2$ の確率で状態2のセルのどちらかが状態1に変わる。
- 状態2のセルが2個で、状態1のセルが1個の場合、状態1のセルは状態2に変わり、 $1/2$ の確率で状態2のセルのどちらかが状態1に変わる
- 状態3のセルが1個、状態2のセルが1個、状態0のセルが1個の場合、状態0のセルは状態1に変わり、状態2のセルは状態1に変わる(安定化する)
- 状態3のセルが1個、状態2のセルが1個、状態1のセルが1個の場合、状態1のセルが状態2に変わり、状態2だったセルが状態1に変わる

#### 4.2 StarLogoによる粒子クラスタの形状変形シミュレーション

高井らが行った、剛体球クラスタを障害物で押したときの形状変形のシミュレーションを、StarLogoを用いて行った。剛体球クラスタのモデル化でも六角格子上のセルオートマトンを用いるので、StarLogoの柵目の中で、図3の濃く塗られた柵目だけを使っている。このシミュレーションの様子を図11に示す。シミュレーションの1サイクルは、右側の障害物(縦に並んだ濃いセル)を1粒子分左に移動させ、障害物の左側に状態2のセルを発生させて始まる。この後、トリプレットの中の状態遷移とトリプレットの組み合わせの切り換えを順次行って、クラスタ内部での状態2の粒子のランダムウォークを行う。そして、全ての状態2の粒子がクラスタの外縁で状態1の粒子となって安定したところで1サイクルが終わる。図11では、2サイクルごとのスナップが表示されている。

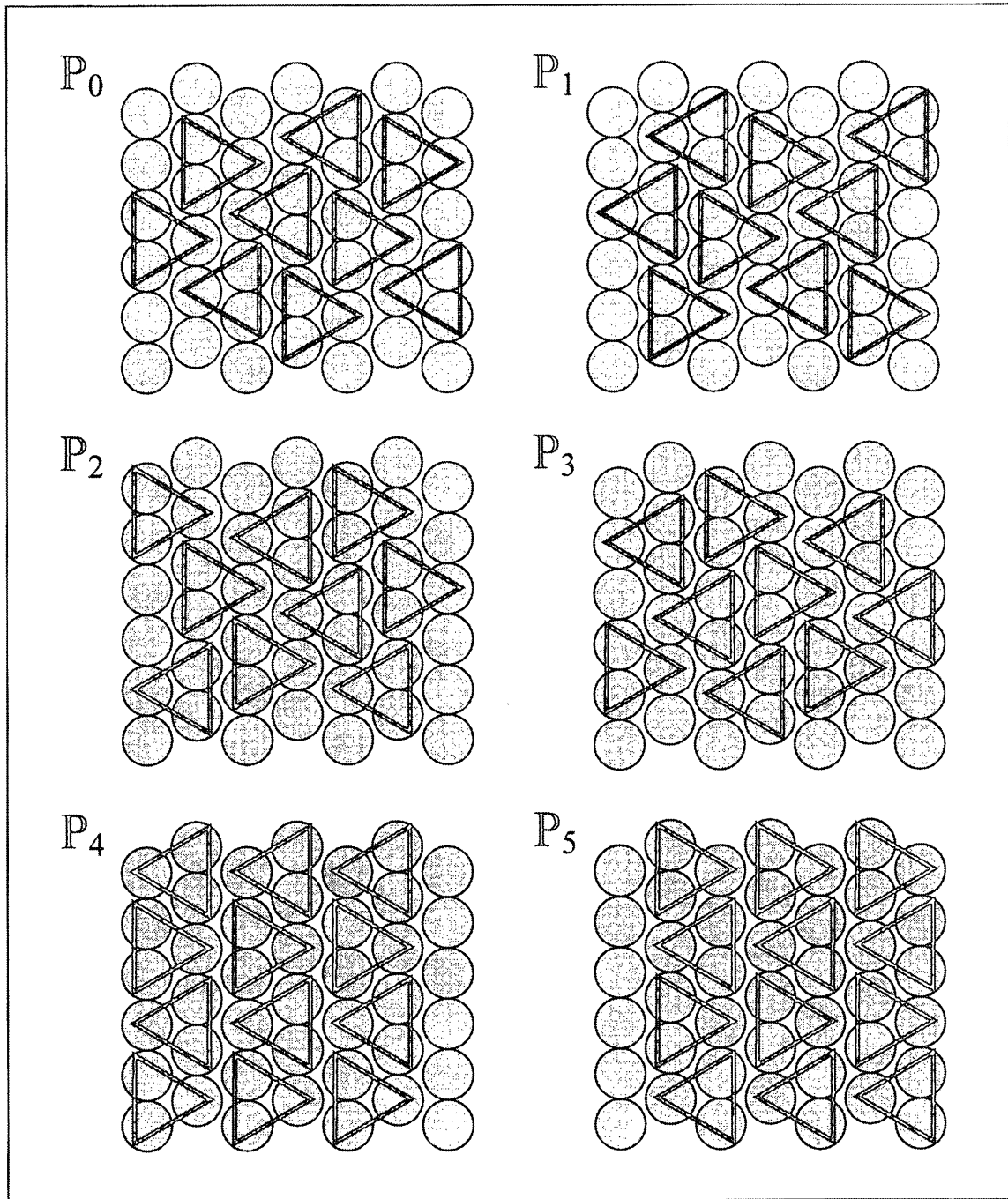


図10 トリプレットを用いて六角格子を重複なく分割する方法

#### 4.3 StarLogoによる鞭毛成長のシミュレーション

本研究では、鞭毛の組織をごく単純化してその成長のシミュレーションを行う。鞭毛は3次元の構造を持っているが、それを軸線に沿って切った場合の切り口の平面だけをモデル化する。さらに、細胞の粒は全て同じ大きさで、六角格子状に並んでいるものとする。

このように単純化すると、鞭毛成長のシミュレーションは、剛体球クラスタを障害物で押したときの形状変形のシミュレーションと似通ったものになる。鞭毛成長の場合にも、クラスタ内部で状態2の不安定セルが発生し、それがクラスタ内でランダムウォークし外縁まで達したときに安定した状態1のセルになるという手法が使える。ただ、剛体球クラスタの場合には、状態2のセルは障害物で

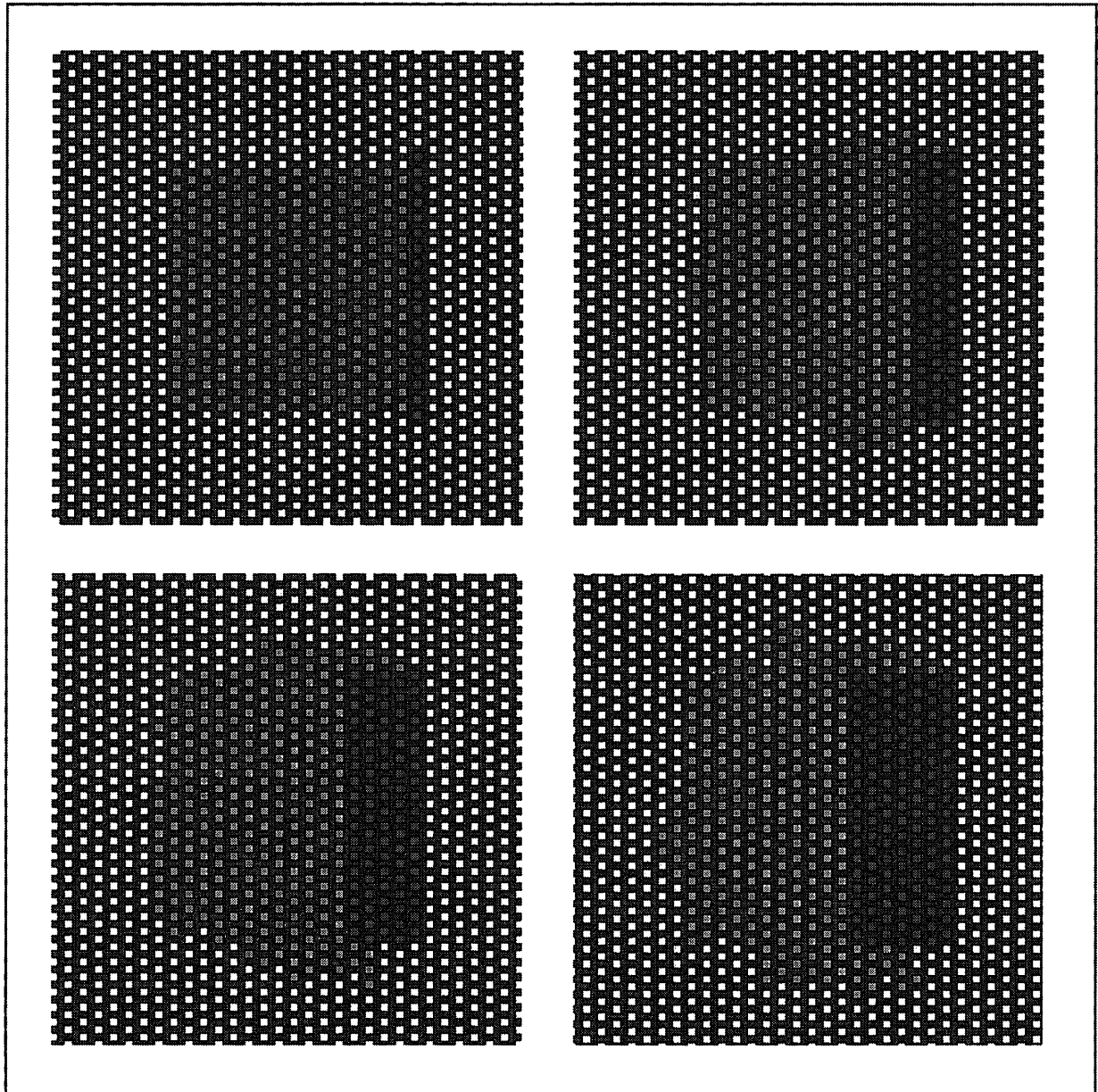


図11 剛体球クラスタを障害物で押したときの形状変形のStarLogoによるシミュレーション

押すことによって発生し、粒子の数は不変であるが、鞭毛成長の場合には、状態2のセルの発生は、成長点で細胞分裂が起こり、新しい細胞が発生する現象のモデル化であり、粒子の数は増えて行く。

状態2のセルが発生する位置を1点で固定して、サイクルを進めると、粒子クラスタは、シャボン玉の泡が膨らむように全方向に成長する。鞭毛はある方向に成長し、細長い形に

ならなくてはならないので、本研究のシミュレーションでは、細胞分裂が起こる成長点もある方向に移動するものとする。

成長点は直線的に移動し、通過した跡は、障害物として振舞う硬い軸ができると仮定した場合の、StarLogoによるシミュレーションのスナップを図12に示す。この図では、濃い灰色のセルが障害物のセル、薄い灰色のセルが普通の細胞のセル、白いセルが空白のセル、

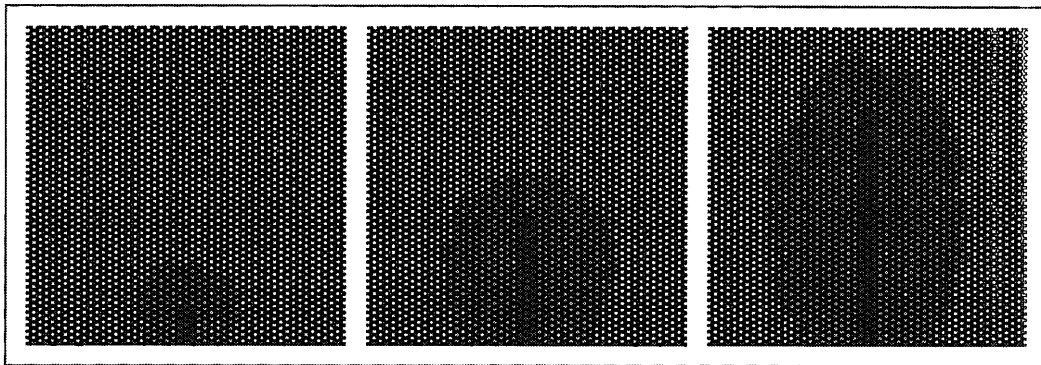


図12 StarLogoによる鞭毛成長のシミュレーション

黒い部分はStarLogoの柵目で使われていない部分である。このシミュレーションでは、グラフィック・キャンパスの下辺に沿って障害物を配置し、細胞がそれよりも下に成長しないようにしている。新しい細胞が発生する成長点は、横位置での中央にほぼ横に並んだ3点で、その下は障害物（軸）になっている。成長点は上方に移動していくが、図12のシミュレーションでは、細胞が10回発生するたびに一度1セル分移動するように設定してある。成長点が移動する動作は、剛体球クラスターを障害物で押したときの形状変形シミュレーションの1サイクルとまったく同じである。

このシミュレーションにおいても、StarLogoの柵目の数が十分ではなく、鞭毛が成長する方向が光などの影響で変化するという動作を表現するのは難しい。

## 5. まとめ

本研究では、分散システムのモデル化環境であるStarLogoを用いて、雪の結晶の成長と、鞭毛成長のシミュレーションを行った。いずれも自己形成のシミュレーションであるが、雪結晶の場合は、新たな粒子が外部に付着してパターンが成長していくというモデルを使い、鞭毛成長の場合は、新たな粒子は内部で発生し、そのためにパターンの外縁が外部に拡大するという反対のモデルを使った。シミュレーションの手法はいずれの場合もセルオートマトンを用い、並列的な処理によってステップを進めた。

StarLogoは分散システムを扱うための環境であるので、本研究のようにセルオートマトンを用いたシミュレーションを行う際には、プログラムが容易であり、簡単にグラフィカルな表示ができるというメリットがある。しかし、柵目の数が101×101個しかなく、本格的なシミュレーションを行うには、不十分である。

## 参考文献

- 1) Zhimov V, Herr D: New Frontiers: Self-Assembly and Nanoelectronics. Computer: 34-43, January 2001
- 2) Fujita M, Fujita N, Ogura K, et al.: Spontaneous assembly of ten components into two interlocked, identical coordination cages. Nature 400: 52-55, 1999
- 3) 本郷健: 分散システムシミュレーションの教材化に関する研究. 日本産業技術教育学会第42回全国大会講演要旨集: 84, 1999
- 4) Resnick M: Turtles, Termites, and Traffic Jams: Explorations in Massively Parallel Microworlds. MIT Press, Cambridge, 1997
- 5) Colella V, Klopfer E, Resnick M: Adventures in Modeling: Exploring Complex, Dynamic Systems with StarLogo. Teachers College Press, New York, 2001
- 6) StarLogo on the Web: <http://education.mit.edu/starlogo/>
- 7) 湯浅太一、安村通晃、中田登志之: はじめての並列プログラミング. 共立出版、東京、1998
- 8) Flake G: The Computational Beauty of Nature: Computer Explorations of Fractals, Chaos, Complex Systems, and Adaptation. MIT Press, Cambridge, 1998
- 9) Packard N: Lattice models for solidification and aggregation. Proc. 1st Int. Symposium for Science on Form: 95-101, 1985
- 10) 響田康、広瀬貞樹、蜷川繁、ほか: セルオートマトンを用いた雪の結晶の類似パターン生成.

- 電子情報通信学会論文誌 J83-D-1 (9) : 909-918, 2000
- 11) 高井昌彰、斎藤敏樹、富川裕樹、ほか：確率的セルラオートマトンによる粒子クラスターの形状変形モデル. 電子情報通信学会論文誌 J83-A (12) : 1565-1568, 2000
- 12) Margolus N : Physics-like models of computation. Physica 10D: 81-95, 1984