Java 言語を用いた自己組織化現象のシミュレーション

木下 浩太

2004 年 2 月 24 日

電子情報工学科

目次

図目次

第 1 章 序論

1.1 研究の背景

近年多くの物理化学現象のモデルがコンピュータ上でシミュレートされているが、その多くがシーケンシャルと呼ばれる逐次型のプログラムで実行されている。しかし実世界の現象の多くは時間とともに同時並行で状態が変化しており、現実を忠実に再現したものではなかった。また現象の変更、追加、削除があった場合、モジュール、プロセス等の変更に多くの時間がかかってしまっていた。

これに対しコンカレントと呼ばれる並行型のプログラムは、一般的にシーケンシャルより現実を忠実に再現できることが知られている。このコンカレントに動作するシミュレーションでは、現象の変更、追加、削除があった場合でも容易に変更することができ、より複雑な動作を自然な記述で実現することができる。このようなコンカレントに動作するシミュレーションを用いて、より現実に近い現象を再現することが必要である。

1.2 Java 言語について

今回、並行なシミュレーションを行うに当たって用いるコンピュータ言語は Java である。

Java 言語はアメリカの Sun Microsystems 社が開発したプログラミング言語で、完全なオブジェクト指向性を備えている。また、強力なセキュリティ機構や豊富なネットワーク関連の機能が標準で搭載されており、ネットワーク環境で利用されることを強く意識した仕様になっている。

Java で開発されたソフトウェアは特定の OS やマイクロプロセッサに依存することなく、基本的にはどのようなプラットフォームでも動作する。この汎用性の高さは Java 最大の特徴であり、「Write Once, Run Anywhere」というキャッチコピーで、その利便性が強く主張されている。

Javaで記述されたソースコードは、コンパイル時にJavaバイトコードと呼ばれる中間コードにいったん変換される。ソフトウェアはJavaバイトコードの状態で配布され、実行時にはJava仮想マシン(Java Virtual Machine)と呼ばれるソフトウェアによって、実行するプラットフォームに対応した形式(ネイティブコード)に変換され、実行される。プラットフォーム間の違いはJava仮想マシンが吸収してしまうため、仮想マシン上で動作するJavaプログラムは、プラットフォームの違いを意識しなくてもよくなる。

一方、Javaの欠点として、プログラムをJavaバイトコードからネイティブコードに変換する際にある程度時間がかかるため、通常のプログラミング言語で開発されたソフトウェアよりも動作が遅くなってしまう。また、どのプラットフォームでも動作させるために、プラットフォーム固有の強力な機能を利用することはできない。

このような欠点を補うため、特定のプラットフォームでしか動作しないがその分高速で、プラットフォーム固有の強力な機能を利用できるJava開発環境を提供しているメーカーもある。

1.3 並列と並行の違い

まず並列処理と並行処理の違いについて簡単に述べておく。並列処理(parallel processing)とは「物理的に」多数のプロセッサによって実行される処理のこと指し、主に1つの大きな処理を分割して複数のプロセッサで計算させることが多い。並行処理(concurrent processing)とは「論理的に」複数の処理が実行される。よって、プロセッサの数が一個でも複数でも構わない。並行処理は1.4節でも述べるが、実時間に対して行われている処理は1つである。それに対して並列処理はプロセッサの数によって行われてる処理の数が変わってくる。図1.1に並列処理と並行処理の模式図を示す。

白色の枠の部分がそれぞれ処理を表している。図1.1の場合、並列処理の方はプロセッサが2個あると考えられる。並列処理は実時間に対して処理が重なっているが、並列処理の場合は実時間に対して行われている処理は1つである。

よく似た概念に分散処理(distributed processing)がある。これはネットワークで結ばれた複数の計算機を用いて処理を行うことをいう。

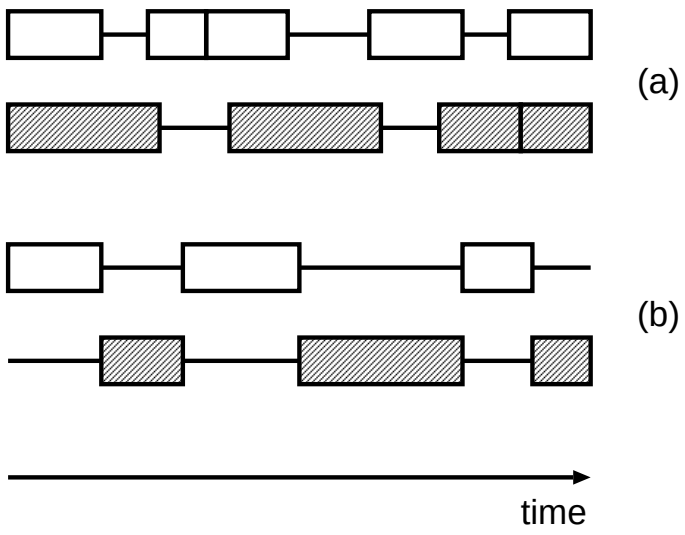


図 1.1 (a) 並列（parallel）と (b) 並行（concurrent）の違い

1.4　マルチスレッド

並行処理はマルチスレッドのプログラムによって実行される。スレッドとは本来「糸」を意味するもので、ここではプログラムを実行している主体のことを指す。従来のシーケンシャルなプログラムは時間に対して実行しているスレッドが 1 つしかないため、これをシングルスレッドのプログラムと呼ぶ。マルチスレッドとは時間に対して実行している 2 つ以上のスレッドを持ち、その処理を並行に行っていくことをいう。図 1.2 にシングルスレッドとマルチスレッドを比較した模式図を示す。白色部分と影色部分がプログラムの実行部分としている。シングルスレッドの場合、常にプログラムを実行している主体は 1 つしかないため、片方の処理が終了した後もう一方の処理に移る。マルチスレッドの場合ではスレッドの数に応じた数だけプログラムを実行している主体がある。つまり、図 1.2 の場合では 2 つの処理が並行に実行される。図 1.2 において、さらに詳しく見てみると、Thread1 がプログラムを実行している場合は, もう一方の Thread2 は休止状態にある。つまり、ある瞬間においてのプログラムの実行はシーケンシャルであるが、実行されているスレッドは 2 つあり、全体的に見ていくとコンカレントである。この切り替わるタイミングは計算機環境、及び Java 言語の実行系に依存するが、排他制御が行われるためスレッドの実行が重なり合うことはない。

Java 言語はマルチスレッドを標準でサポートしており、したがってシ

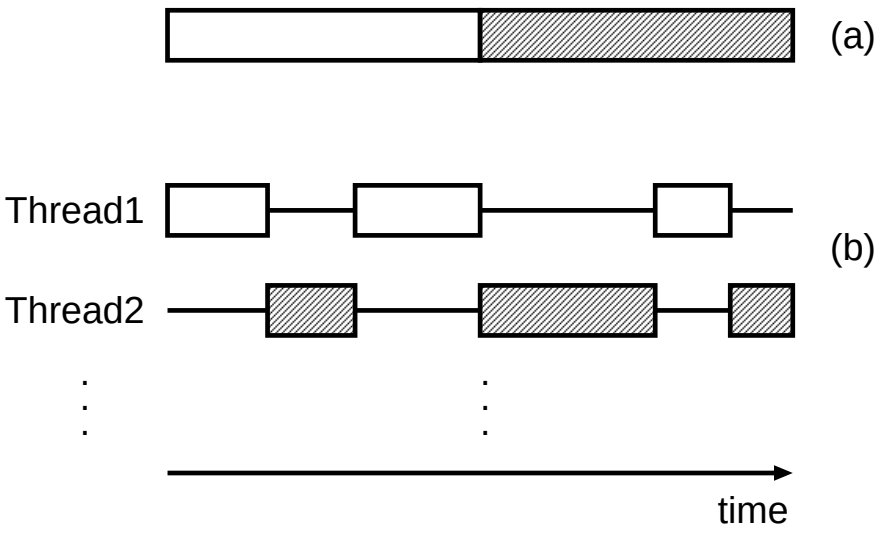


図 1.2 (a) シングルスレッドと (b) マルチスレッドの違い

ミュレーションごとに異なった並行動作の定義をすることなくまた、そのシミュレーションに応じたスレッドの数、処理を行うことができる。

このマルチスレッドを用いることによって並行動作のシミュレーションが実現可能であると考えられる。

1.5 本研究の目的

本研究では、コンカレントに動作するシミュレーションの利点である、より現実に忠実なシミュレーションを再現できる、現象に応じた自然なプログラミングが可能である、現象の変更が容易に実現できるということを、実際に Java 言語のマルチスレッドを用いて作成し検証を行う。1 つ目の利点を検証するため Turing パターンのシミュレーションを作成する。ここでは同じ条件でシーケンシャルに動作するシミュレーションを作成し比較を行う。また 2 つ目の利点を検証するために、スレッドに位置情報を与え周りとの位置関係を考慮しながら動作する細胞の生成と消滅のシミュレーションを作成する。そしてこのシミュレーションを元に条件を削除、追加することで別の分野である、磁束クリープ・フローのシミュレーション、永久磁石を用いた鉄球浮上のシミュレーションを作成し、3 つ目の利点について検証する。

第 2 章 シミュレーション

2.1 Turing パターンのシミュレーション

2.1.1 概要

ビーカーの水の中にインクを一滴垂らす。するとインクは徐々に広がって最終的には均一になる。ところが自然界を見まわすと、とくに動物たちの皮膚や殻に注目すると、多様なパターンで満ち溢れていることに気づく (図 2.1)。動物や昆虫、熱帯魚の図鑑をひらいてみると、シマウマの縞模様、チーターのまだら、チョウやテントウムシの斑点など独特のパターンで埋め尽くされている。これらの模様は、それぞれの動物によって多少の違いがあるものの、胚段階のときに皮膚細胞に色素やその発現を制御する化学物質などが拡散してできることが分かっている。

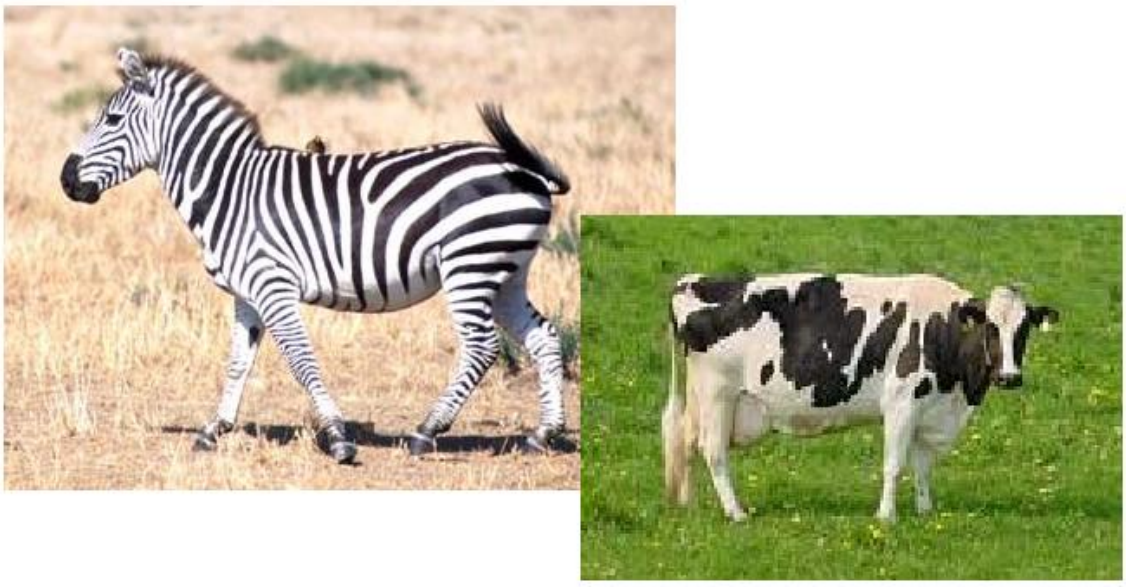


図 2.1 自然界の動物たち

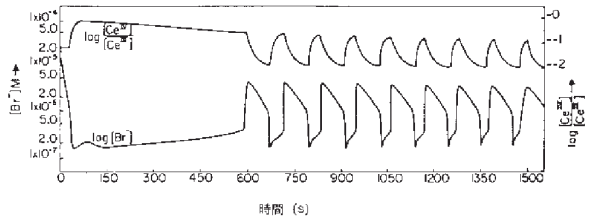


図 2.2 BZ 反応の振動現象の観測例

図 2.3 BZ 反応における空間パターンの観測例

このように直感的に反する現象に対してはじめて数学的に挑戦したのが、チューリング (Alan Turing) であった。チューリングは 1952 年に、反応拡散系でアニマル柄のように静止パターンがうまれることを提唱した。化学結合などによって最終的にとりうる構造とは対照的に、Turing パターンは反応・拡散の過程の中で一定のリズムやパターンがうまれてくる動的な秩序化である。ここで言うリズムとは時間的な細胞内の濃度の振動であり、パターンとは構成要素が自ら進んで、ある空間的な散逸構造を形成することである。自己組織化現象ではよく知られている BZ 反応の振動現象 (リズム)、空間パターンの観測例を図 2.2、図 2.3に示す。Turing パターンもこのような濃度の振動と空間パターンがうまれる。

生物は外部から物質を摂取し、内部で発生した熱などを排出する開放形であるが、やはり細胞のレベルで見ても同じことが当てはまる。化学反応の場所としての動物の皮膚細胞は、熱力学的に見れば散逸系ということになる。アニマル柄の現れるのは自己組織化のためである。

当時、化学反応のダイナミクス研究の歴史が浅かったため、チューリングの提案を実際に確かめることは出来なかったが、1990 年になってようやく実験によって確かめられた。また、近年の計算機の処理能力の向上によって、コンピュータのスクリーン上で容易に Turing パターンを発生させるこ

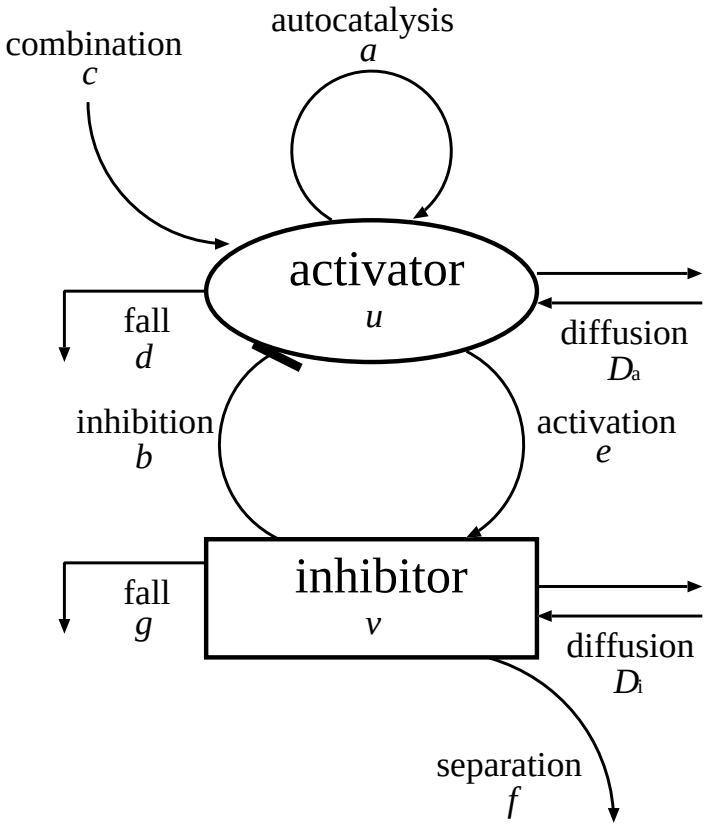


図 2.4 微分方程式が意味する化学反応の模式図

とが出来るようになったり、80 年ごろから注目を集めてきた複雑系などの後押しもあって、チューリングのアイディアは多くの人に認識されるようになった。

現在 Turing パターンは、日常的なスケールでの多様な生命現象を理解するために重要な役割を果たしている。

2.1.2 シミュレーション

チューリングは Turing パターンを以下のような数学モデルで表現した。今回この微分方程式を元にプログラムを作成した。図 2.4 にこの微分方程式が意味する化学反応の模式図を表す。

$$\frac{\partial u}{\partial t} = (a-d)u - bv + c + D_{\mathrm{a}}\nabla^2 u \tag{2.1}$$

$$\frac{\partial v}{\partial t} = eu - f - gv + D_{\mathrm{i}}\nabla^2 v \tag{2.2}$$

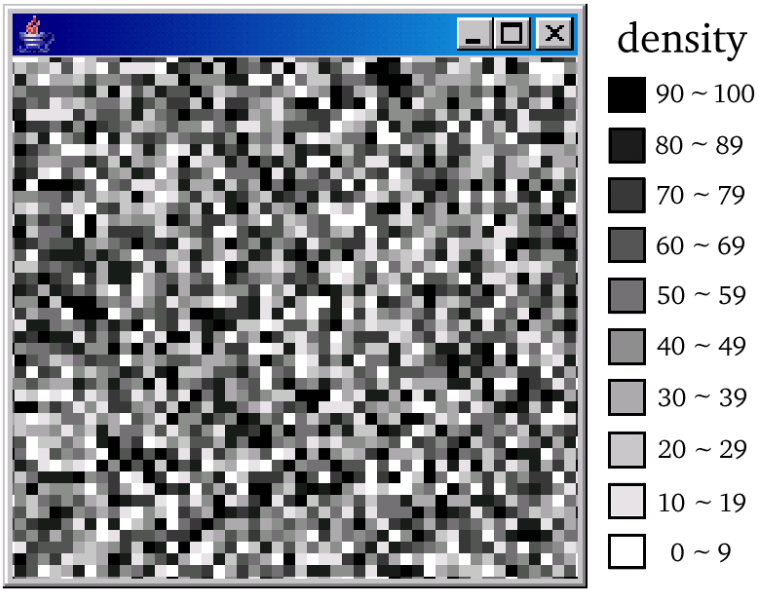


図 2.5 Turing パターンのシミュレーションの初期状態

Turing パターンは図 2.4 のように 1 つの細胞に活性因子 (activator) と抑制因子 (inhibitor) という 2 つの仮想分子が存在している。この微分方程式は単位時間あたりの活性因子、抑制因子の濃度の変化量を表しており、それぞれのパラメータによって濃度の増減が決定される。つまりこのパラメータによってアニマル模様が縞や斑になることが決定されるのである。ここで u、v はそれぞれ活性因子と抑制因子の濃度、D は拡散係数比、それ以外はそれぞれの反応の速度定数 (パラメータ) を表している。活性因子と抑制因子には相互作用があり、活性因子の濃度が増加すればするほど抑制因子の濃度が活性化され増加する。それに対して抑制因子の濃度が増加すればするほど活性因子の濃度が抑制され、抑制因子の濃度がある一定の値を超えると活性因子の濃度は減少する。また活性因子、抑制因子はそれぞれ自触媒作用、合成、減少といった反応に加え、隣接した細胞に濃度の拡散を行っている。

2.1.3 プログラム

シミュレーションには図 2.5 のような 50 × 50 の周期的境界条件を導入した 2 次元配列を活性因子、抑制因子それぞれ 1 つずつ用意し、そこに濃度として 0 ~ 100 までのランダムな値を代入したものを初期状態とする。

つまりある濃度をもった 50 × 50 の細胞が敷き詰められているというこ

とである。この表示は活性因子の濃度を表し黒いほど濃度が高いということを表している。ここで、活性因子、抑制因子それぞれ細胞1つを1つのスレッドで動かし図2.4に示しているパラメータによって濃度を変化させる。

2.2 細胞の生成と消滅のシミュレーション

2.2.1 概要とプログラム

続いて、細胞の生成と消滅のシミュレーションについて説明する。このプログラムでは1つの細胞を1つのスレッドで動かす。図2.6に示してあるように、中央のある丸いものを1つの細胞として表し、黒い細胞が1つある状態を初期状態とする。ランダムにスレッドが呼ばれ細胞を生成したり消滅したりすることを繰り返す。すべての細胞は同じ動作を行うが、ただ細胞の色によって生成する細胞の色が異なっており、図2.7に示すように黒い細胞は赤い細胞、赤は青、青は緑、緑は黒の細胞を生成する。またある確率で灰色となった細胞は何も生成せず一定時間が経つと消滅する。

すべての細胞はそれぞれ自分の位置情報を持っており、他の細胞と位置を計算し一定の距離を保つよう動作する。距離が近いと$a\exp(-r/r_0)$の力で斥力、遠いと$-b(p-p_{\mathrm{i}})$の力で引力が働く。ここでa、bは斥力、引力の強さを表すパラメータ、rは自分と相手の距離、r_0は保とうとする距離、pは自分自身の位置、p_iは相手の位置を表す。このシミュレーションで用いた距離に対する斥力、引力の強さを図2.8、図2.9に示す。

スレッドの最大数は30とし、最高30個の細胞が互いに相互作用しながら動作をする。

2.3 磁束クリープのシミュレーション

2.3.1 概要

磁束クリープとは、第2種超伝導体の混合状態でピン止めされていた磁束線が熱揺動により、ある確率でピンポテンシャルを飛び出す現象である。この現象の影響が顕著に現れるのは、超伝導永久電流の緩和である。これは磁束線がピンニングセンターに捕らえられている状態は、エネルギーの状態空間における局所的な極小に対応した準安定状態でしかなく、真の平衡状態でないことによる。そのため真の平衡状態に向けての緩和、すなわち遮蔽電流

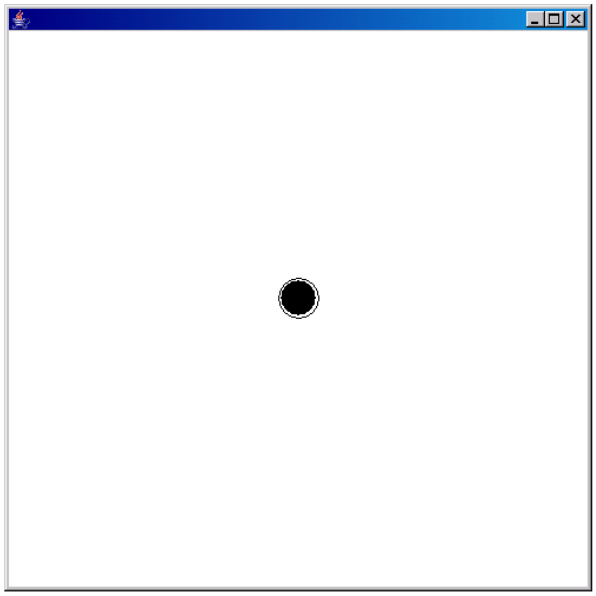


図 2.6 細胞の生成と消滅のシミュレーションの初期状態

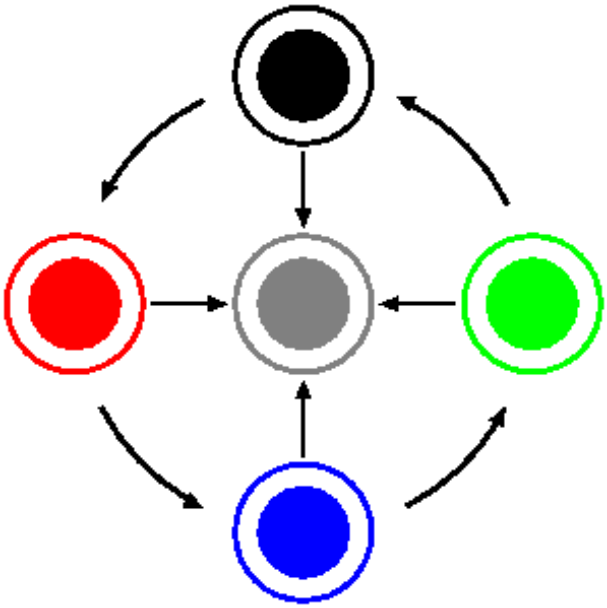


図 2.7 生成する細胞の種類

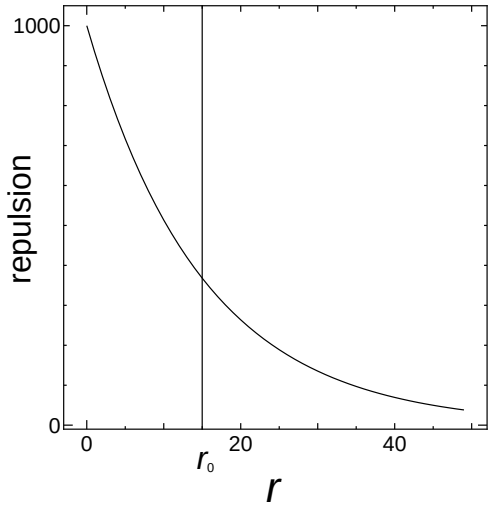


図 2.8 距離に対する斥力

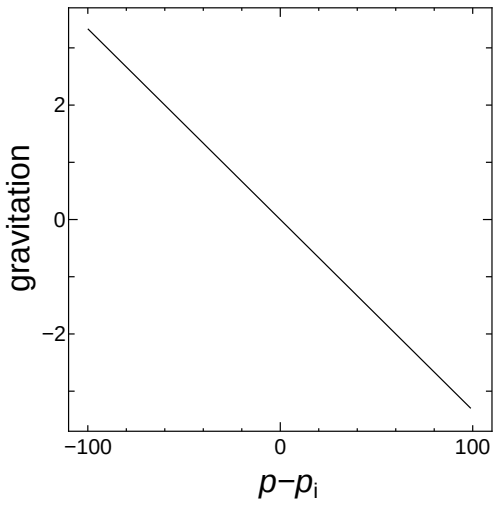


図 2.9 距離に対する引力

の減衰が起こる。この現象を磁化の緩和といい、緩和の時間変化が図 2.10 のように減衰が時間に対して対数的であることである。遮蔽電流の減衰は、超伝導体内の磁束分布の変化に対応している。こうした熱活性化による磁束線の運動は、磁束線が磁束フローのように磁束線格子全体の連続的な運動ではなく、一部の不連続なものであると考えられている。運動する磁束線の集団を磁束バンドルといい、こうした磁束線の運動を磁束クリープという。

超伝導体に電流が流れると、ピン止めされた磁束バンドルにローレンツ力が働く。その磁束バンドルを仮想的に変位させていった場合のエネルギーの変化を図 2.11 に示す。点 A は、磁束バンドルがピン止めされている状態であり、エネルギーが全体的に右下がりになっているのは、ローレンツ力による仕事を考慮しているためである。電流を流さない場合つまりローレンツ力が働かない場合、エネルギー図は水平になる。このときの活性化エネルギー U がピン・ポテンシャル U_0 と等しい。磁束クリープが生じると、熱エネルギーのために磁束バンドルが捕まっているピンニング・センターからはずれて点 B のエネルギー・バリヤーを越え、ローレンツ力の方向に動き出す。磁束バンドルがこのエネルギー・バリヤーを越えてローレンツ力の方向に動き出してしまう確率は Arrhenius の式 $\exp(-U/k_{\mathrm{B}}T)$ で与えられ、磁束クリープを起こして生じる電界の大きさは、ピン・ポテンシャル内での振動周波数を ν_0、磁束線格子間隔を a_{f} とすると、

$$E = Ba_{\mathrm{f}}\nu_0 \left[\exp\left(-\frac{U}{k_{\mathrm{B}}T}\right) - \exp\left(-\frac{U'}{k_{\mathrm{B}}T}\right)\right] \tag{2.3}$$

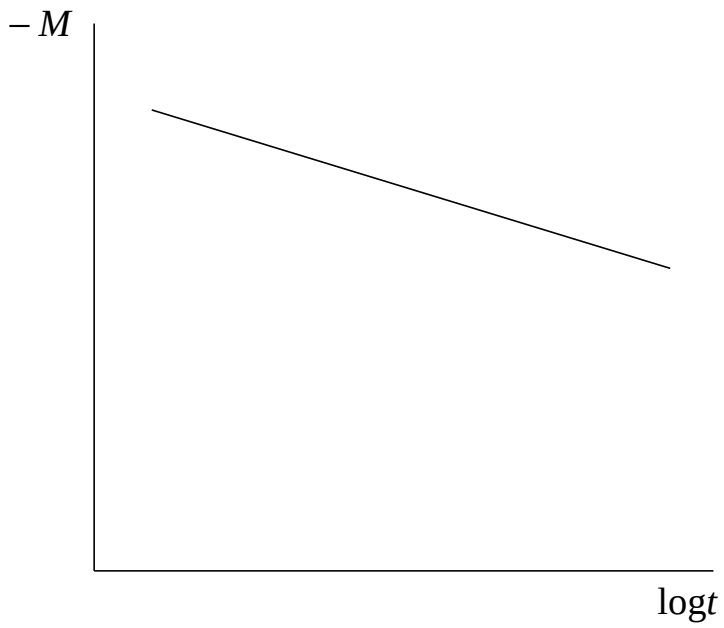


図 2.10 磁化の緩和

で表せる。

ここで、磁束バンドルの中心位置を x とし、図 2.11のポテンシャルに以下の正弦波的なものを仮定する。

$$F(x) = \frac{U_0}{2}\sin kx - fx \tag{2.4}$$

ここで $k = 2\pi/a_{\mathrm{f}}$ である。V を磁束バンドルの体積とすると、$f = JBV$ は磁束バンドルに働く Lorentz 力である。磁束バンドルの平衡位置は、(2.4) 式を x について微分して

$$x = \frac{1}{k}\cos^{-1}\left(\frac{2f}{U_0 k}\right) \equiv -x_0 \tag{2.5}$$

が得られる。また、$F(x)$ は $x = x_0$ で極大となっており、この関係から活性化エネルギーは $U = F(x_0) - F(-x_0)$ から求まる。したがって

$$\frac{U}{U_0} = \left[1 - \left(\frac{2f}{U_0 k}\right)^2\right]^{1/2} - \left(\frac{2f}{U_0 k}\right)\cos^{-1}\left(\frac{2f}{U_0 k}\right) \tag{2.6}$$

となる。もし熱揺動がなければ、$U = 0$ となる理想的な臨界状態が達成される。この場合は $x_0 = 0$ となるので、$2f/U_0 k = 1$ でなければならず、このときの電流密度 J が磁束クリープがないとした場合の仮想的な臨界電流密度 J_{c0} となる。したがって、

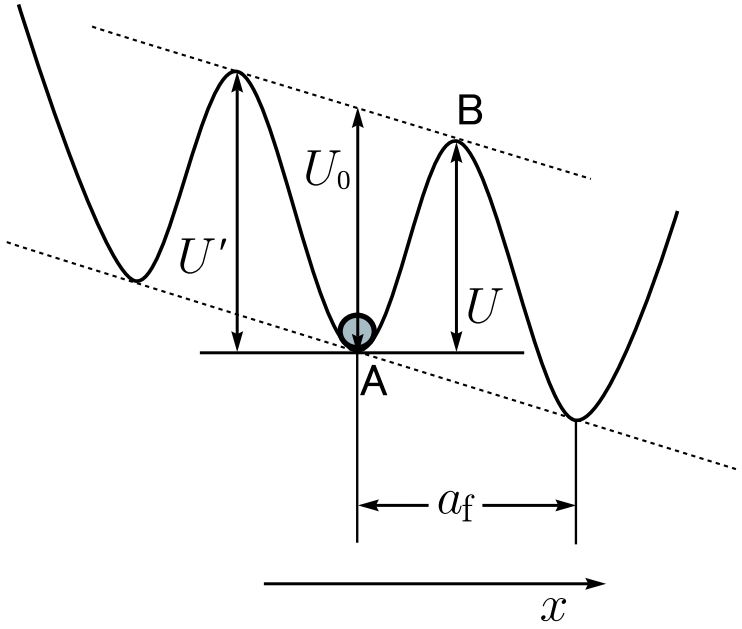


図 2.11 磁束バンドルの位置とエネルギーの関係

$$\left(\frac{2f}{U_0 k}\right) = \frac{J}{J_{\mathrm{c0}}} \equiv j \tag{2.7}$$

の関係が得られる。よって (2.6) 式は

$$U(j) = U_0[(1-j^2)^{1/2} - j\cos^{-1}j] \tag{2.8}$$

となる。また、

$$U' \simeq U + f a_{\mathrm{f}} = U + \pi U_0 \frac{J}{J_{\mathrm{c0}}} \tag{2.9}$$

の関係が得られる。これより (2.3) 式は

$$E_{\mathrm{cr}} = B a_{\mathrm{f}} \nu_0 \exp\left[-\frac{U(j)}{k_{\mathrm{B}}T}\right]\left[1 - \exp\left(-\frac{\pi U_0 j}{k_{\mathrm{B}}T}\right)\right] \tag{2.10}$$

と表すことができる。

2.3.2 プログラム

まず図 2.12 は実際の柱状欠陥 (ピンニングセンター) を導入した超伝導体表面の様子である。白く見える部分が柱状欠陥であり、黒い部分は超伝導部分である。図 2.13 にシミュレーションの様子を示す。図 2.12 と同じように超伝導体を上から見た図であり、黒い丸は磁束線、黄色い丸はピンニングセンター、それ以外は超伝導部分を表す。磁束線 150 個、ピンニングセ

ンター 200 個を周期的境界条件を導入した 250 × 400 の領域にランダムで配置したものを初期状態とし、磁束線 1 つを 1 つのスレッドで動かす。ローレンツ力は右方向にかかっており、周りに何もない状態のとき、磁束線は右方向に移動する。また磁束同士には相互作用があり、一定の距離を保とうとする力が働く。

T、k_{B}、ν_0、J、J_{c0} の値を適当に決め、(2.9) 式を用いて j を導出し、(2.8) 式を用いて U を導出した。この U を Arrhenius の式に代入し、磁束バンドルがエネルギーバリアーを越えてローレンツ力方向に動き出す確率を求めた。

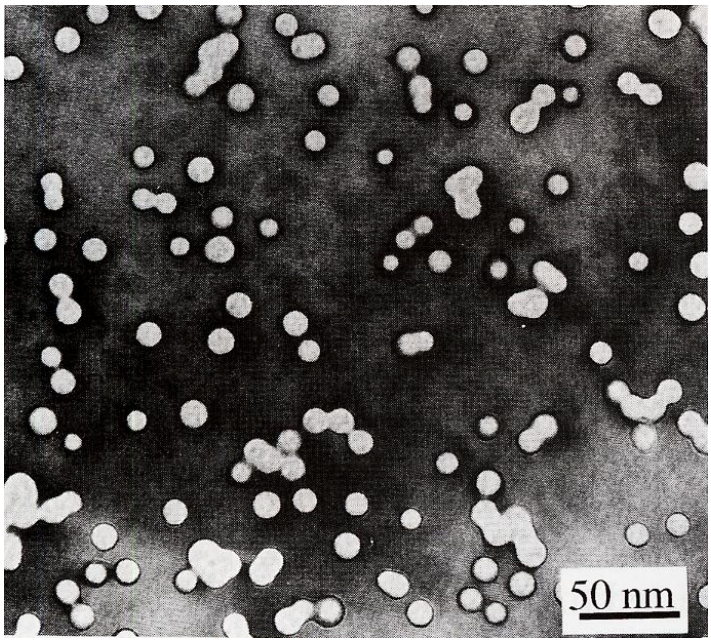


図 2.12 柱状欠陥を導入した超伝導体表面の様子

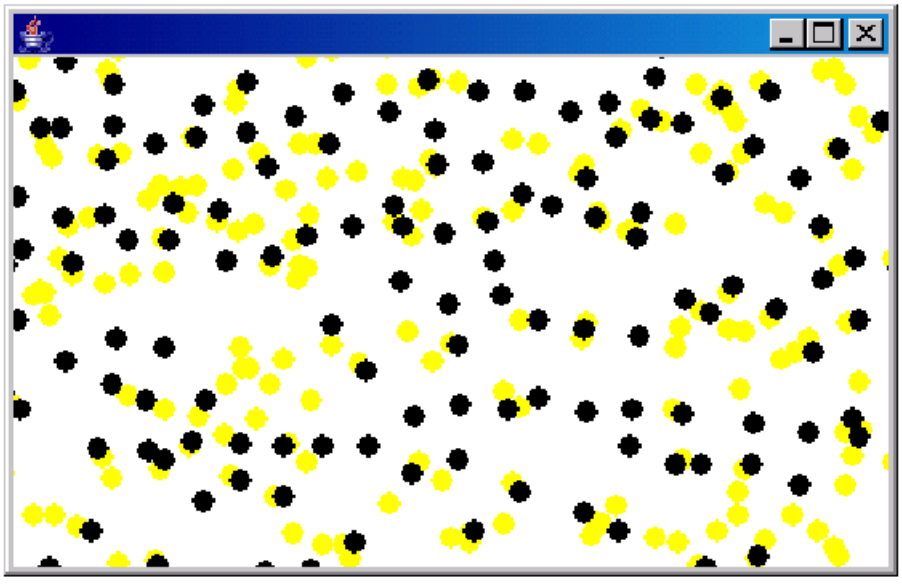


図 2.13 シミュレーションの様子

2.4　永久磁石を用いた鉄球浮上のシミュレーション

2.4.1　概要

私立岩手高校 (盛岡市) の佐々木修一教諭ら[4] の研究グループが、永久磁石を使って鉄球を浮上させる実験に成功、岩手県庁で 2003 年 12 月 24 日、成果発表の記者会見と実験を行った。超伝導物質による浮上は知られているが、1842 年に発表された「アーンショウの定理」で、永久磁石を動かさずに鉄球が浮くことはないとされていた。

実際の実験の様子を図 2.14 に示す。実験では約 20 個のパチンコ玉 (鉄製、直径 1cm、重さ 4g) を入れたプラスチックの箱の上部にリング状の強力な永久磁石を置いた。すると 5 つの玉が 1 列に並び磁石に吸い付いたが、約 3mm 離れた中空で別の 1 つが静止した。

永久磁石の磁極を紙面向きとすると、鉄球に誘導される磁極は、紙面と反対向きとなる。上列の鉄球はすべて同じ極が方向を揃えて隣り合っており反発する力がお互い働くが、永久磁石に引き付けられる力の方がはるかに大きいため、お互いが離れることはない。そして浮上している鉄球は永久磁石へ引き付けられる力、上列の磁化された鉄球と反発する力、浮上している鉄球の重力がつりあっているため浮上したものと考えられる。

2.4.2　プログラム

まず図 2.15 のように磁石、プラスチックの箱、鉄球を配置する。鉄球 1 つを 1 つのスレッドで動かし、それぞれの鉄球は磁石によって磁化されており、同じ極が隣り合っているため反発する力が働いている。また磁石との距離が近いほど磁石に引きつけられる力が強く、すべての鉄球に重力も働いている。磁石に引き付けられる力は $\exp(a/R^3)$、鉄球同士の反発は $b\exp(-r/r_0)$ とした。ここで a、b は磁石の引力、磁化された鉄球の反発力を表すパラメータであり、R は磁石と鉄球の距離、r は鉄球と鉄球の距離、r_0 は反発力が働く距離を表している。磁石と鉄球の引力はしばらくして磁石を右に動かし鉄球の様子を見る。

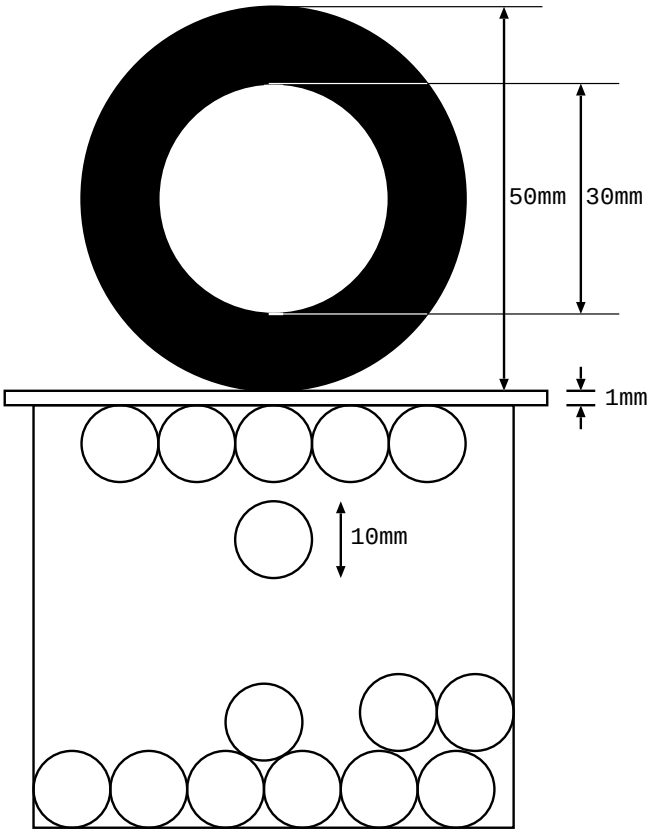


図 2.14 実験の様子

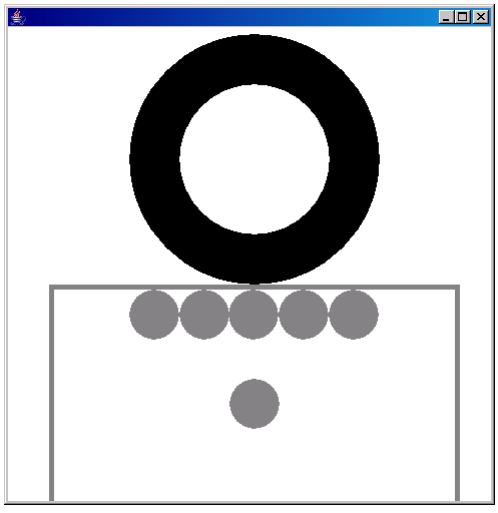


図 2.15 シミュレーションの様子

第 3 章 結果及び検討

3.1 Turing パターンのシミュレーション

3.1.1 実行結果と検証

シミュレーションに用いるパラメータの値をまとめたものを表 3.1 に示す。D_{a} 以外のパラメータを固定し、D_{a} のみを 0.020∼0.040 まで変化させ

表 3.1 Turing パターンのシミュレーションに用いるパラメータ

activator		inhibitor	
a	0.1135	b	0.10
c	0.02	f	0.06
d	0.0315	g	0.06
e	0.1	D_{i}	0.5
D_{a}	$0.020 \sim 0.040$		

たときの実行結果を図 3.1 に示す。

この結果から傾向はほとんど変わらないのに対し、D_{a} が大きいほど濃度が高い部分の幅が広くなっていることがわかる。少しのパラメータを変化させただけで異なる自己形成が起っており、これからパラメータを任意に変化させることで、さまざまな Turing パターンが再現できることが予想できる。

3.1.2 コンカレントとシーケンシャルの実行結果の比較

次に全く同じ初期条件で細胞の配列の順序にシーケンシャルに濃度の計算をさせたシミュレーションを Java 言語で作成し実行結果の比較を行った。図 3.2 に $D_{\mathrm{a}} = 0.020$ で、コンカレント、シーケンシャルなプログラム

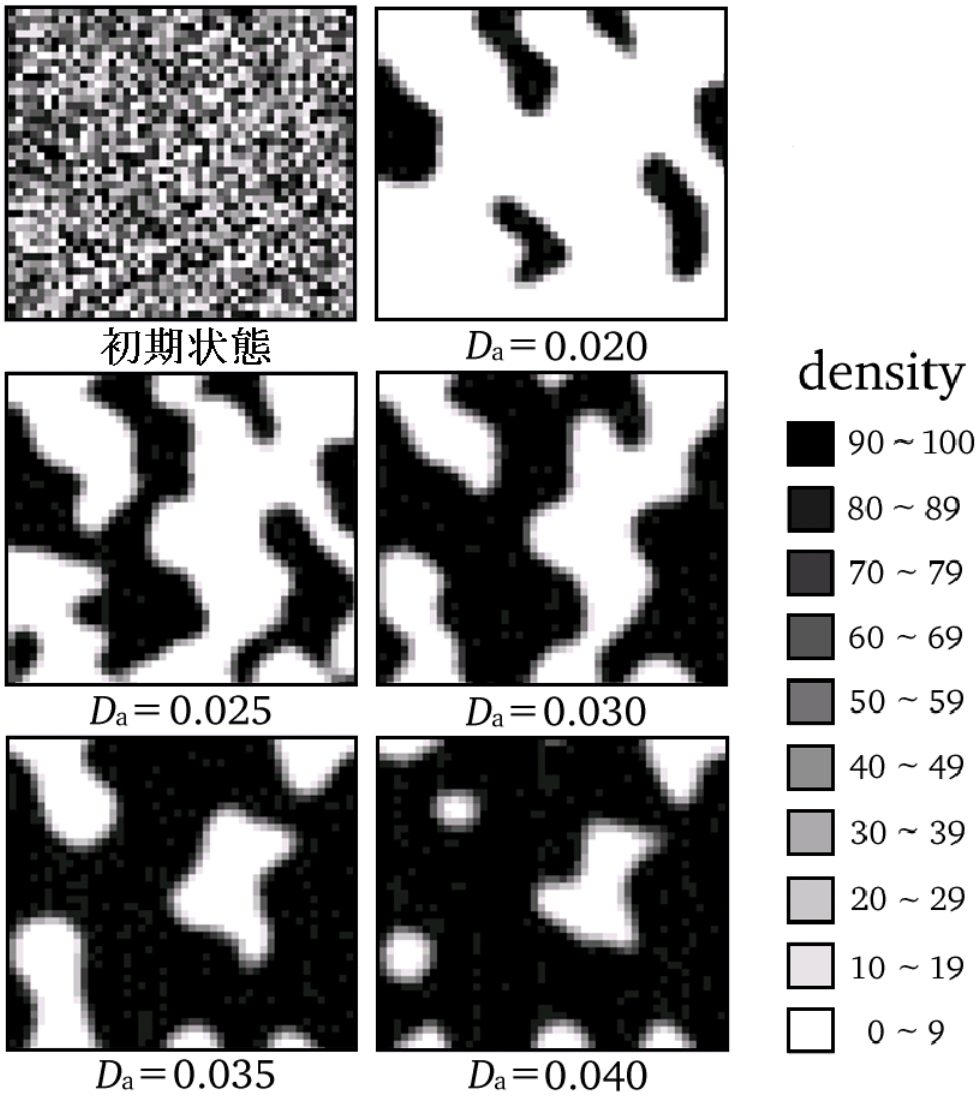


図 3.1 D_a=0.020 ～ 0.040 まで変化させたときの実行結果

で実行させたときの実行結果を示す。この結果を比較してみると、ほぼ同様の形状が得られていることがわかる。しかしよく見ると違いが現れており、図 3.3 に図 3.2 の活性因子の濃度が 80 ～ 89 の細胞を赤く表示させたものを示す。これから明らかにコンカレントなプログラムの方が変化の様子が細かく現れており、シーケンシャルなプログラムでは白と黒の境目付近しか変化の様子が現れていないのに対し、コンカレントなプログラムでは黒の領域内でも変化の様子が現れている。

次に活性因子の個数の変化の比較を行った。図 3.4 にコンカレントなプログラム、シーケンシャルなプログラムのグラフを示す。

グラフは横軸にプログラム上の時間、縦軸に活性因子の個数をとり、濃度によって 10 種類の色に分類しそれぞれの変化の様子を表している。つまり黒のラインは活性因子の濃度が 90 ～ 100 である細胞の時間に対する個数の変化を表している。図 3.3 で違いが現れた活性因子の濃度が 80 ～ 89 の細胞の変化の様子はグラフでは赤のラインで表されている。グラフからも変化に違いが出ており、コンカレントなプログラムではある値で飽和しているのに対し、シーケンシャルなプログラムでは減少し 0 に近づいている。また、黒のラインである活性因子の濃度が 90 ～ 100 である変化の様子がコンカレントなプログラムでは一定時間で飽和状態に達しているのに対し、シーケンシャルなプログラムでは増加を続けている。このように高濃度の細胞の変化に違いが現れている。

次にこの違いを明らかにするために高濃度の細胞に注目して変化の様子の比較を行った。図 3.5 にコンカレントなプログラム、シーケンシャルなプログラムのグラフを示す。

横軸にプログラム上の時間、縦軸に活性因子の濃度をとり、高濃度に飽和する 5 つの細胞を色分けして変化の様子を表している。このグラフからコンカレントなプログラムでは飽和した細胞に周期的な振動が見られ、図 3.4 で赤で示していた濃度が 80 ～ 90 の範囲に変化の様子がよく現れていることが図 3.5 からも理解することができる。つまり Turing パターンの特徴である反応・拡散の過程の中で一定の時間的な振動 (リズム) が生まれている。シーケンシャルなプログラムではこのリズムはほとんど見られなかった。またシーケンシャルなプログラムで濃度が増加傾向にあった理由として、高濃度となった細胞はほとんど振動する変化が見られず減少していな

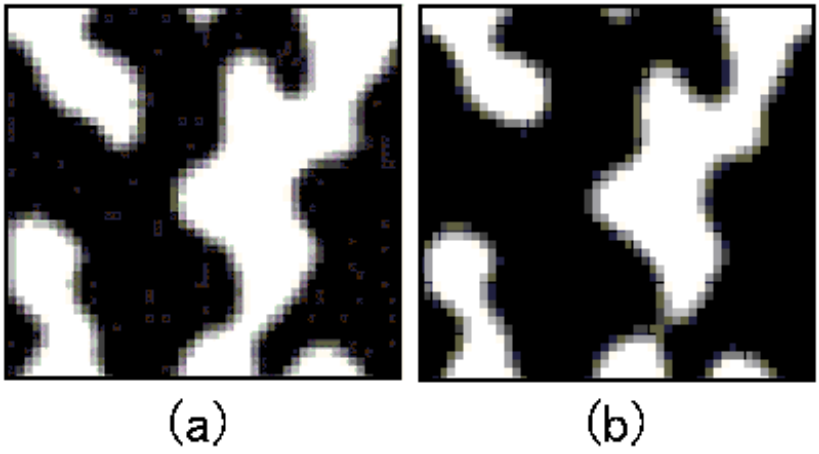


図 3.2 D_a=0.020 のときの実行結果 (a) コンカレント (b) シーケンシャル

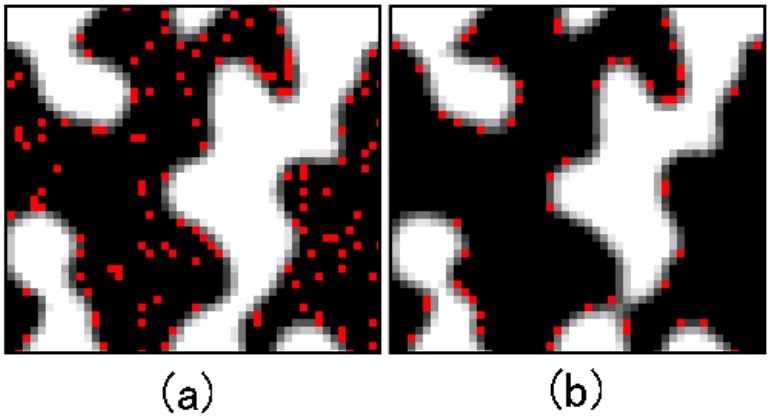
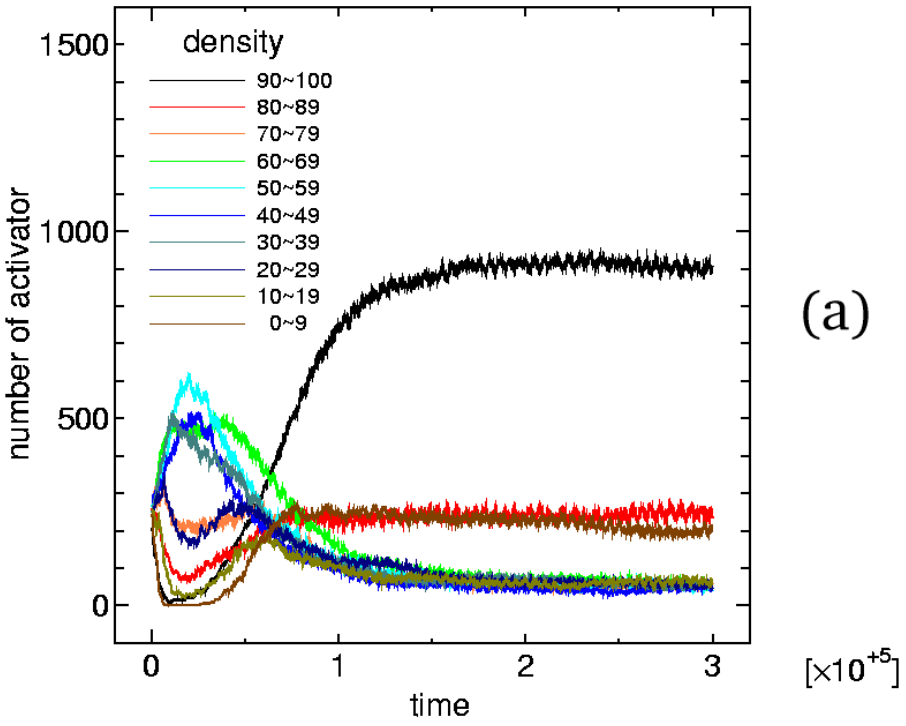


図 3.3 D_a=0.020 のときの実行結果 (濃度が 80 ～ 89 の細胞を赤で表示したもの (a) コンカレント (b) シーケンシャル)



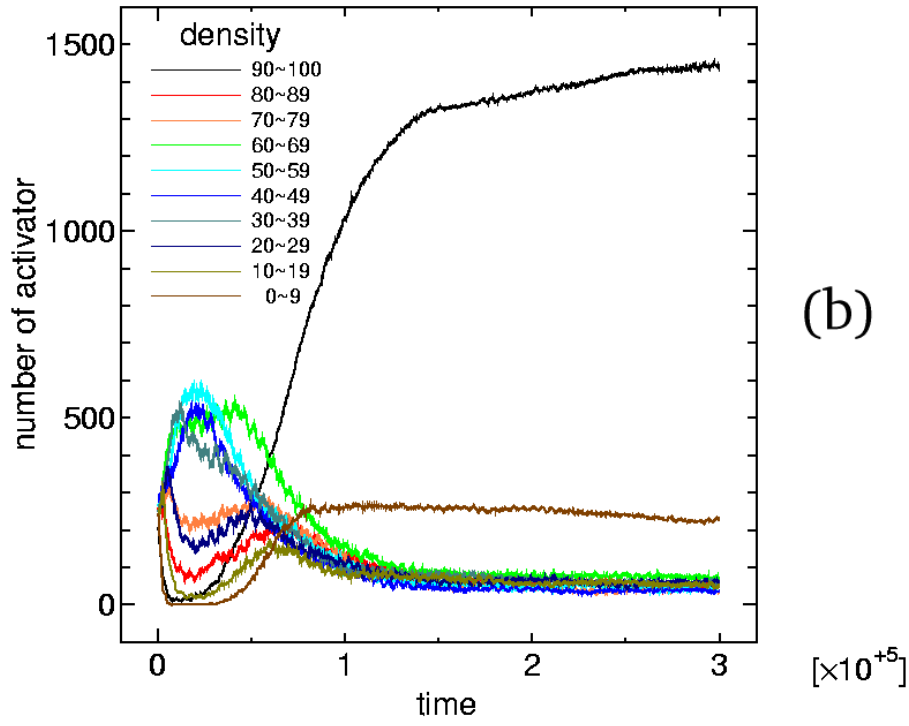
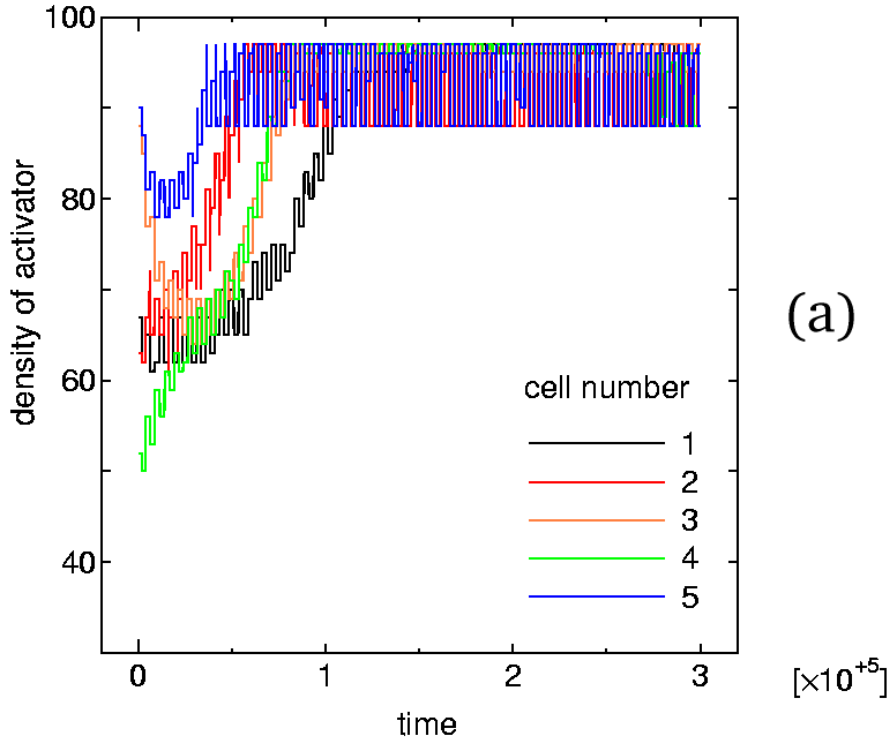


図 3.4 活性因子の個数の変化 (a) コンカレント (b) シーケンシャル



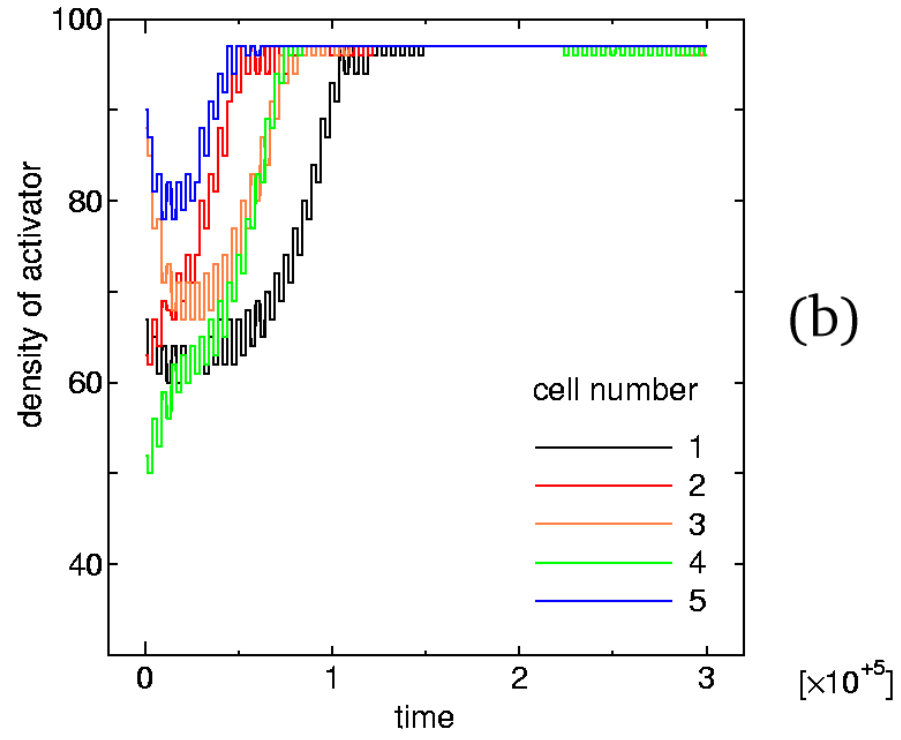


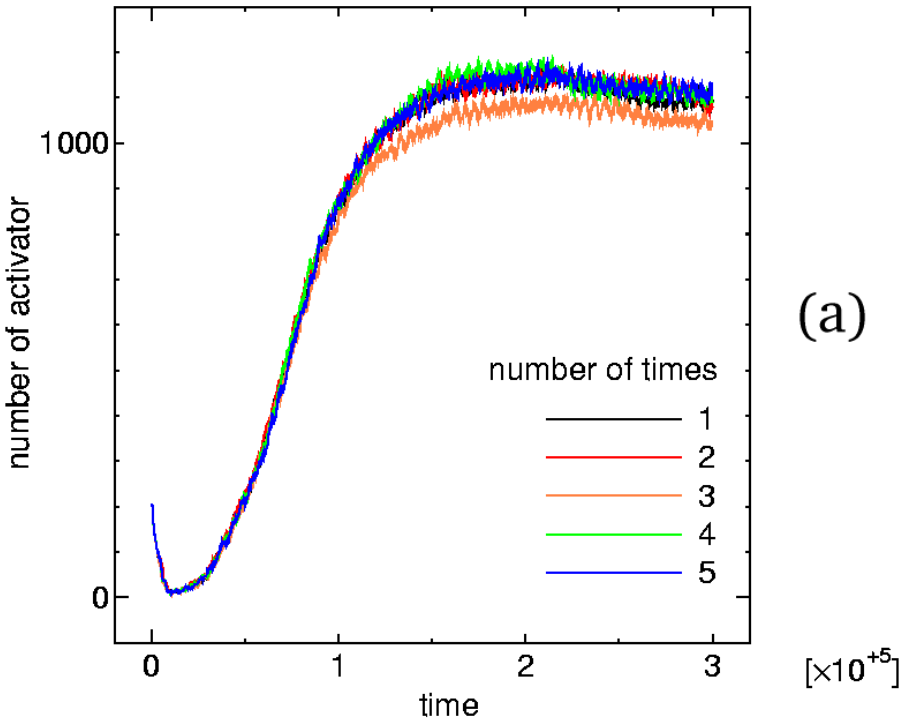
図 3.5 高濃度細胞の個数の変化 (a) コンカレント (b) シーケンシャル

いためその分コンカレントなプログラムより多く拡散し全体として活性因子の濃度が増加したためだと考えられる。

次に全く同じ条件で複数回実行させたときの比較を行った。図 3.6 にコンカレントなプログラム、シーケンシャルなプログラムのグラフを示す。

横軸にプログラム上の時間、縦軸に活性因子の濃度をとり、合計 5 回実行させ色分けして変化の様子を表している。このグラフからシーケンシャルなプログラムでは同じ条件で複数回実行させると必ず同じ結果になるのに対し、コンカレントなプログラムではほぼ同じ傾向を示しているものの、全く同じ結果になることがないということがわかる。これはシーケンシャルなプログラムでは、細胞の配列順に必ず実行されるが、コンカレントなプログラムではスレッドの実行される順序が計算機環境、Java 言語の実行系に依存しているため全く同じ順序では実行されないからである。

このシミュレーションからマルチスレッドを用いて Turing パターンをシミュレート可能であるということ、パラメータを任意に変化させることでさまざまな Turing パターンが再現できるということがわかった。またシーケンシャルなプログラムに比べてコンカレントなプログラムは、高濃度の細胞について変化過程の情報量が多い、同じ条件で複数回実行させた場合全く同じ結果になることがないという違いが現れた。この違いからコンカレントなプログラムの方がより現実に近いシミュレーションであると言える。



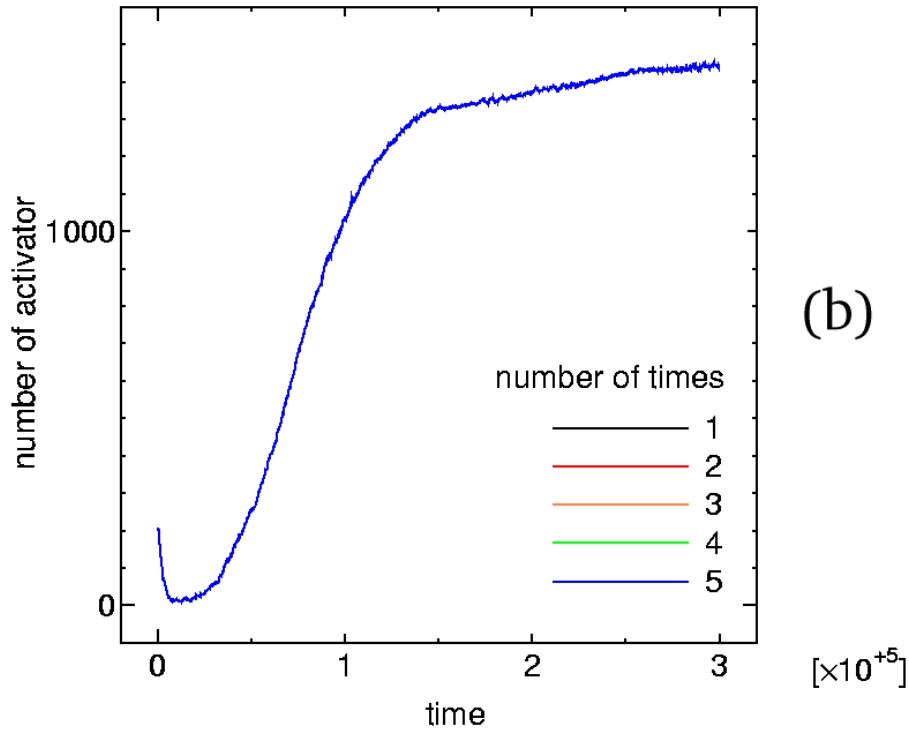


図 3.6 複数回実行させたときの変化の様子 (a) コンカレント (b) シーケンシャル

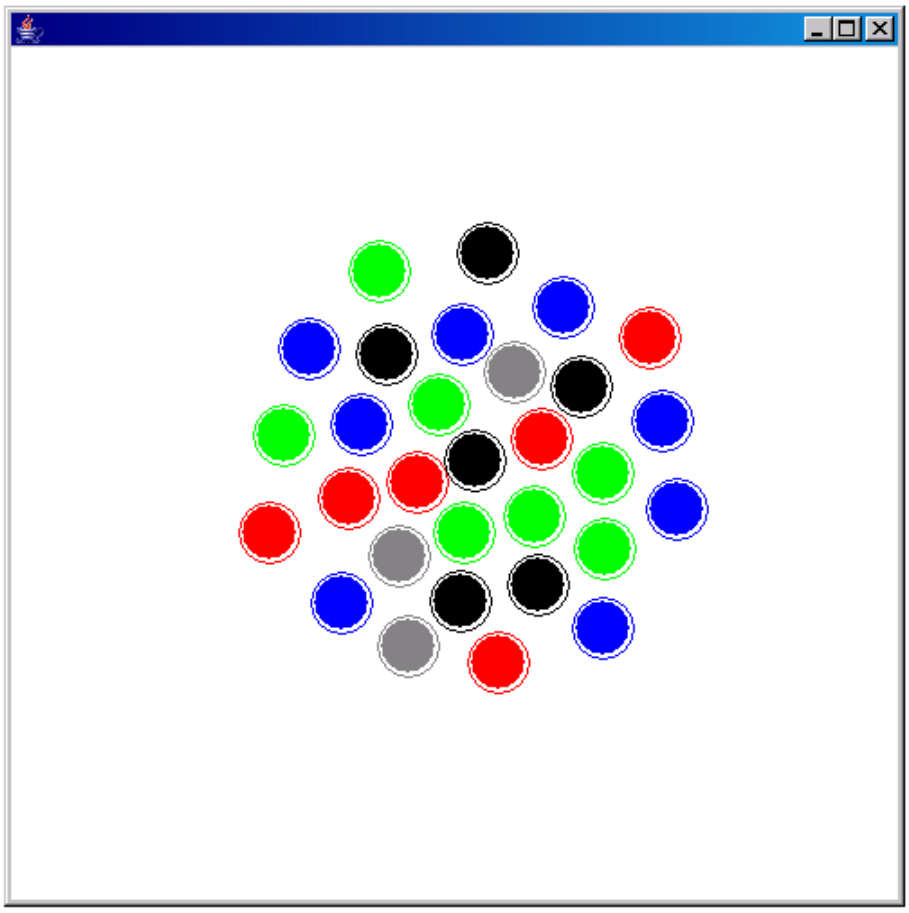


図 3.7 シミュレーションの実行結果

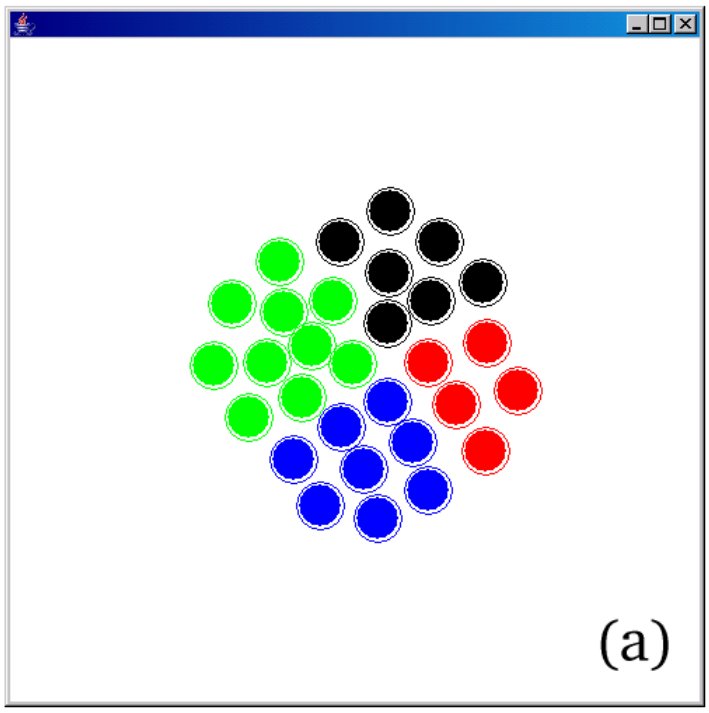
3.2 細胞の生成と消滅のシミュレーション

3.2.1 実行結果と検証

実行結果を図 3.7 に示す。これから 30 個以下の細胞がそれぞれ相互作用しながら動作していることがわかる。次に一部の条件を変更した実行結果を示す。

図 3.8(a) は同色細胞に通常より強い引力を与えたもの。図 3.8(b) は同色細胞に通常より強い引力を与え、さらに横方向に強い引力を与えたものである。

同色細胞に強い引力を与えた場合、すべての細胞と相互作用しながらも同色の細胞が集まり斑模様のようなものができている。またさらに横方向に強い引力を与えると、それぞれの細胞の色で横に縞模様ができている。このように簡単な細胞のシミュレーションからもパラメータを少し変える



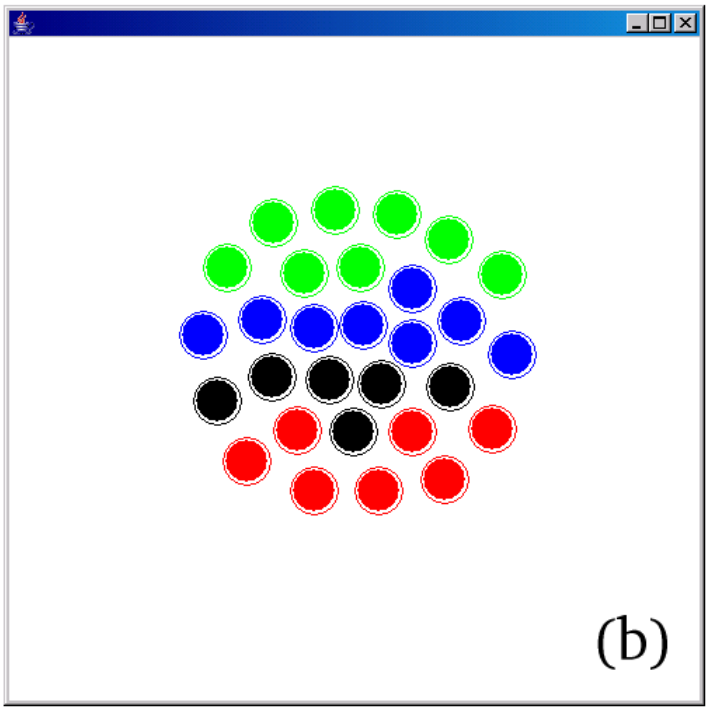


図 3.8 (a) 同色細胞に強い引力を与えた実行結果 (b) 同色細胞、横方向に強い引力を与えた実行結果

ことによって異なるパターンを作ることができる。またそれぞれの細胞が別のスレッドで動作しているため条件を容易に与えることができ、動きを制御したり変更したりすることができる。

3.3 磁束クリープのシミュレーション

3.3.1 実行結果と検証

シミュレーションを実行させると磁束バンドルが、捕まっているピンニング・センターからはずれてローレンツ力の方向に動き出す様子を見ることができた。次に電界 E をすべての磁束線の平均速度として求め、J の値を変化させ $E-J$ 特性を求めた。そのグラフを図 3.9 に示す。

このグラフから J が 20 よりも小さいときはあまり大きな増加は見られないが、20 ～ 80 のとき著しく増加している。そして 90 よりも大きくなるとほとんど増加する様子は現れない。また T が大きいほど E が大きくなることがわかる。

次に図 3.10 に松下研究室岡村和憲ら[3)] の FZ 法で作成した Bi2212 単結晶を磁化緩和測定法を用いて得られた $E-J$ 特性を示す。実験に用いた試料のサイズ、ドープ条件は表 3.2 、表 3.3 に示すとおりである。

表 3.2 試料のサイズ

ab 面内サイズ	c 軸方向の長さ
1.82mm × 2.04	13μm

表 3.3 試料の諸元

ドープ条件	照射前の T_c	照射後の T_c	異方性パラメータ (γ_a)
～ 700℃ 0.1MPa(air)	89 K	87 K	187

図 3.9は実際の実験から得られたものとは定性的、定量的にも異なっている。それはこのシミュレーションに用いた値、全体に対してのピンニングセンターの割合、大きさなど、現実とは異なるものを多く用いたためである。実際に実験から得られたグラフと比較し、検証を行うには、まずより

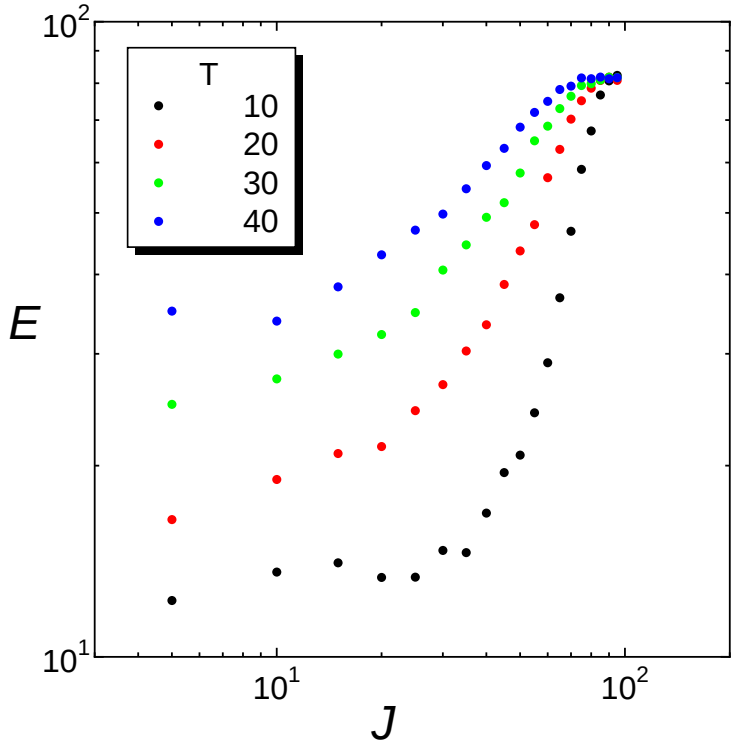


図 3.9 シミュレーションから得られた $E - J$ 特性

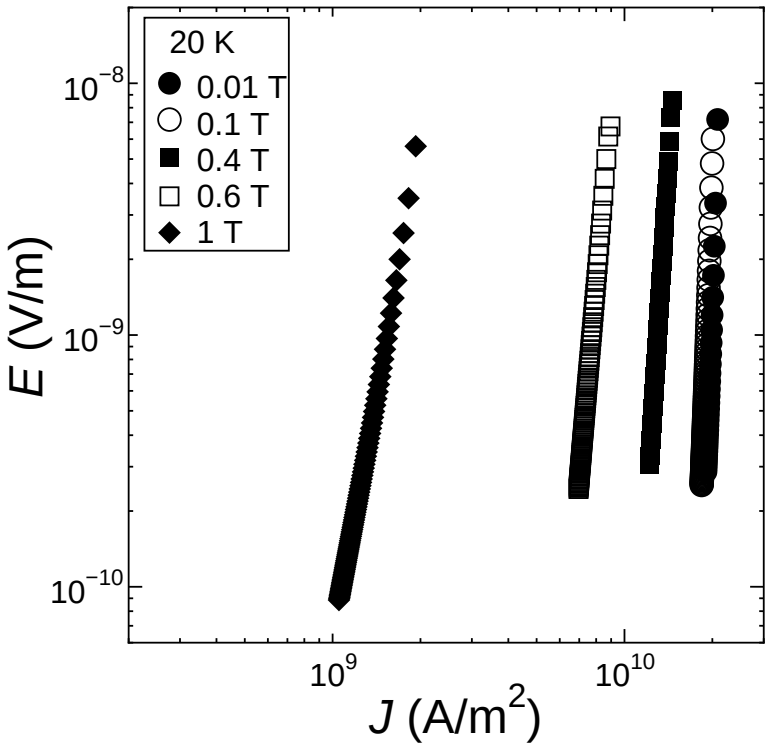


図 3.10 実験から得られた $E - J$ 特性

現実に近い値を用いて計算することが必要であるが、このシミュレーションに適用するのは困難であった。この問題を克服することは今後の課題となる。またこのシミュレーションは細胞の生成と消滅のシミュレーションから細胞を生成する条件を除き、ローレンツ力と柱状欠陥の条件を与えたもので、容易に現象を変更できることが示された。

3.4 永久磁石を用いた鉄球浮上のシミュレーション

3.4.1 実行結果と検証

シミュレーションを実行させてみた。磁石を右に動かすとそれに合わせて鉄球も動いた。中空で静止していた鉄球も磁石と同じ方向へ動き、確かに磁石とそれぞれの鉄球の力、重力がつりあっていることがわかった。

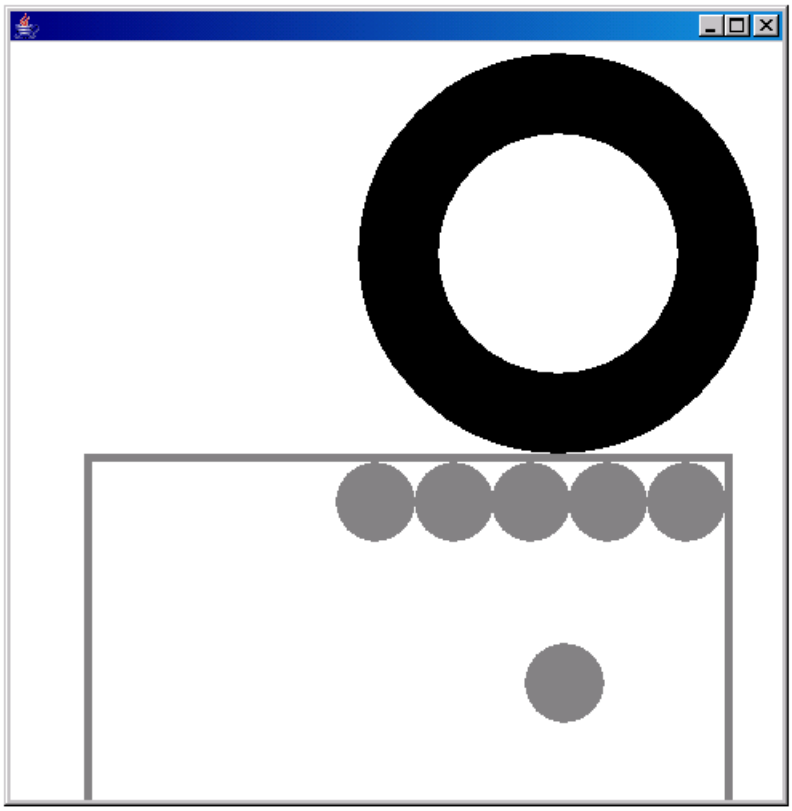
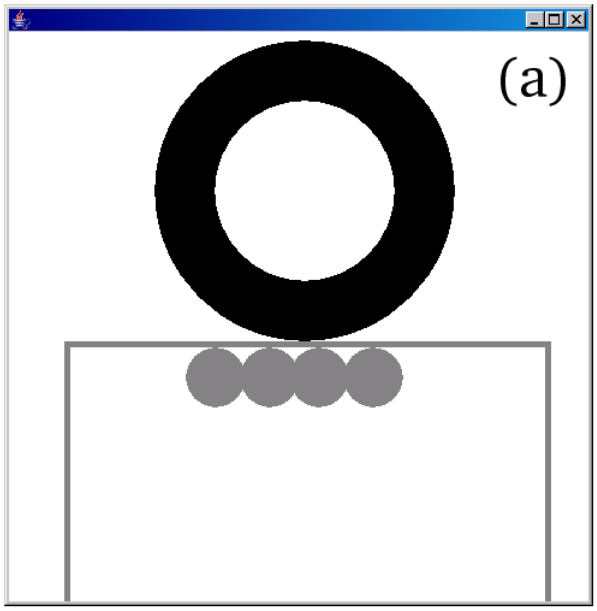


図 3.11 永久磁石を右に動かしたときの様子

次に上段の鉄球を3つにして磁石の磁力を強くした場合と磁力を弱くした場合の実行結果を図3.12 に示す。磁力が強いと浮いていた鉄球も上段の鉄球との反発力より磁石の引きつける力が大きく、上段に4つの鉄球が並

ぶ。磁力が弱いと重力の力で浮いていた鉄球が落ちてしまう。またこのシミュレーションでは磁束クリープ・フローのシミュレーションから、柱状欠陥の条件を永久磁石に与え、鉄球に動作範囲を制限し、重力の力を加えたものである。これは容易に条件を変えることができ、様々な条件でのシミュレーションが可能であるということを示している。



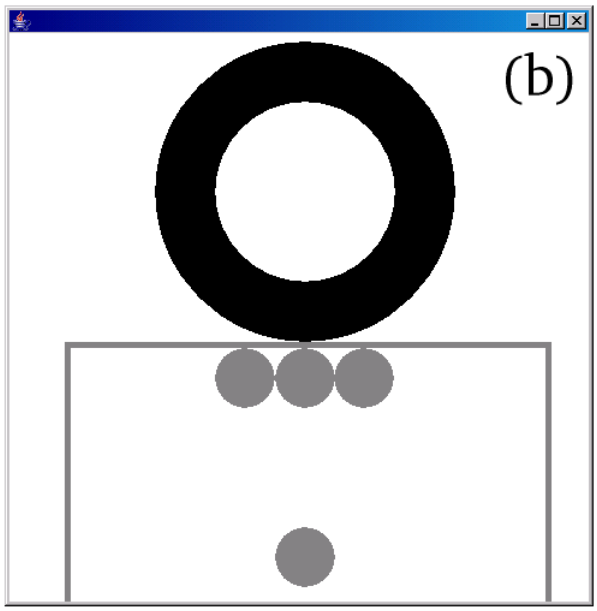


図 3.12 (a) 永久磁石の磁力を強くした場合 (b) 永久磁石の磁力を弱くした場合

第 4 章 結論と今後の課題

4.1 結論

今回シーケンシャルではない、コンカレントな動作を行うシミュレーションを実際に Java 言語で作成し検証を行った。その結果以下のことがわかった。

- Java 言語を用いることでコンカレントで動作可能なシミュレーションを実行できる。このことはマルチスレッドプログラミングを行うことで実証できた。
- シーケンシャルなプログラムを用いたシミュレーションに比べコンカレントなプログラムを用いたシミュレーションは、最終的に得られる結果はほぼ同じであったが得られる情報量が多く、同じ条件でも全く同じ結果にならないことからより現実を忠実に再現しているシミュレーションであると言える。
- マルチスレッドを用いたコンカレントなプログラムではそれぞれの動きを別のスレッドで動かすため、自然なプログラミングができ、容易に条件を与え動きを変更することができる。

4.2 今後の課題

今回コンカレントに動作するシミュレーションの利点を確認できた。しかし現実との比較を行っていないため、どの程度現実に近いシミュレーションを再現できているかわからない。そこで実際の実験から得られたデータとコンカレント、シーケンシャルなプログラムから得られたデータを比較し検討する必要がある。

謝辞

本研究を行うにあたり、多大な御指導、御助言を頂いた小田部荘司助教授に深く感謝します。また並行処理の基本、及び個々の現象のシミュレーションについての助言や指導をして頂いた有明高専電子情報工学科松野哲也助教授、松下照男教授、木内勝助手に深く感謝します。

最後に色々と支援を頂いた小田部研究室、及び松下研究室の皆さんに深く感謝します。

参 考 文 献

1) 結城浩: Java 言語で学ぶデザインパターン入門［マルチスレッド編］、ソフトバンクパブリッシング (2002)

2) 橋口幸司:Java 言語を用いた物理・化学・生物学的現象のシミュレーション [九州工業大学卒業論文 平成 15 年]

3) 岡村和憲:Bi2212 超伝導体の次元性と凝縮エネルギー密度 [九州工業大学修士論文 平成 16 年]

4) S. Sasaki, I. Yagi, M. Murakami, J. Appl. Phys. 95(2004)2090.