

Cinderella.2 によるフラクタルの描画とその教育利用

静岡県立磐田南高等学校 入谷 昭

Akira Iritani
Iwata Minami High school

1. はじめに

コッホ曲線やペアノ曲線、マンデルブロ集合に代表されるフラクタルは、現代数学の華のひとつともいえるべきものである。その数学的内容は高校生には難解であるが、自己相似形によって描かれる図の美しさや不思議さは数学への興味をひきかたてるのに十分であろう。過去に大学入試問題としてコッホの雪片曲線が極限の問題として複数の大学から出題されており、受験問題集にも収録されているが、単なる演習問題として扱われてしまうにはもったいない内容である。コンピュータを用いて実際にフラクタル図形を描いてみれば、その内容についての興味が喚起され、数学への学習意欲が高まることが期待される。また、L-Systemによる植物の成長モデルは、シダやオウム貝などに見られる螺旋や黄金比と並んで、自然界を数学的に解析するものとしての興味が尽きない。授業ではなく、課題研究や数学研究部での研究課題とすることもできる。(螺旋の課題研究については、「数理解析研究所講究録 1735」の「Cinderella.2を用いた授業と課題研究」において紹介済み)

ここでは、Cinderella.2を用いたタートルグラフィクスシステムの構築と、それによるフラクタルの描画を授業に用いた実践例を紹介し、高等学校および大学初年級における教育利用の可能性について論考する。

2. タートルグラフィクスシステムの構築

2.1 タートルグラフィクスシステムの設計

コンピュータでタートルグラフィクスを実現するプログラミング言語には、パパーが提唱したLOGOを起源とする数多くのシステムが存在する。基本は、画面上のタートルに与える前進・後退命令と、描画長・描画方向についての情報で、これにペンの色を指定したり、サウンドを鳴らしたりする命令が付加されている。ユーザーは、これらのコマンドの列を記述してタートルをコントロールして図形を描く。タートルには、1歩進むごとに前と異なる命令を与えることが可能である。

これに対し、ここで設計するタートルグラフィクスシステムは、フラクタルの自己相似形を描くことを念頭に、シンプルに設計する。タートルの進む距離と、向きを変える角度は定数とし、命令はLindenmayerのL-System（文献[1]）に準じて次のように定める。

F,A,B	: 描画しながら前に進む
f	: 描画しないで前に進む
+	: 指定された角度だけ左を向く
-	: 指定された角度だけ右を向く
R,L,S	: なにもしない
[: 現在の状態をスタックに保存する
]	: スタックから情報を取り出す

前進の命令や何もしない命令が複数個あるのは、このタートルグラフィクスシステムの中核となる置き換えシステムのためのものである。置き換えシステムとは、命令文字列のうちの一つの文字を、別の文字列で置き換えるもので、これを繰り返すことにより自己相似な図形を描くことができる。また、スタック命令は、植物の成長モデルにおいて不可欠のものである。なお、命令Sは、Lindenmayerのものではなく、山口大学のテキスト（文献[6]）で用いられている、シェルピンスキーのギャスケットを描くテクニカルな方法を参考に追加したものである。

2.2 プログラミング言語レベルでのシステム

Cinderella.2 は、幾何の描画機能とプログラミング言語CindyScriptを併せ持ったソフトウェアで、描画機能を用いて描画した幾何要素の位置や属性をCindyScriptを用いて制御でき、グラフィクスのための基本的な設定をプログラム側で行う必要がほとんどないのが大きな利点である。最も単純なタートルグラフィクスシステムは、CindyScriptで次の4つの関数を定義することで実現できる。

forward(step)	前にstep分だけ進む
turtle(command,step,angle)	命令文字 command にしたがって亀を動かす stepは1歩の長さ、angleは回転角
makecom(n)	initiatorとgeneratorから命令文字列を生成する
tree(commandrow,step,angle)	命令文字列にしたがって亀を動かす

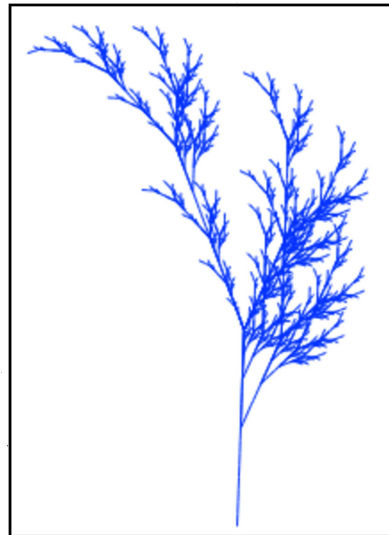
これらの関数は、CindyScriptエディタのInitializationスロットに記述する。

Initializationスロットに書かれたものは、最初に1度だけ実行される。

これに対し、Drawスロットで、1歩の長さ、回転角、イニシエータ、ジェネレータ、置き換え回数を設定して実行する。

次の例は、置き換えシステムによって植物の成長モデルを描画したものである。

<pre> == Initializationスロット == // 前に step だけ進む forward(step):=(draw((0,0),(step,0)); translate((step,0));); // 亀を命令に従って動かす turtle(command,step,angle):=(if(command=="A",forward(step)); if(command=="B",forward(step)); if(command=="F",forward(step)); if(command=="f",translate((step,0))); if(command=="R",); if(command=="L",); if(command=="S",); if(command=="+",rotate(angle)); if(command=="-",rotate(-angle)); if(command=="[",gsave()); if(command=="]",grestore());); // コマンド列を作成する makecom(n):=(workstr=initiator; repeat(n, workstr=replace(workstr,generator););); // コマンド列にしたがって亀を動かす tree(commandrow,step,angle):=(gsave(); repeat(length(commandrow), turtle(commandrow_#,step,angle)); grestore();); </pre>	<pre> == Draw スロット == kaku=22°; nagasa=0.7; initiator="++++A"; generator=[["F","FF"],["A","F-[A]+A] +F[+FA]-A"]; n=5; s=makecom(n); tree(s,nagasa/n,kaku); == 実行結果 == </pre>
--	--



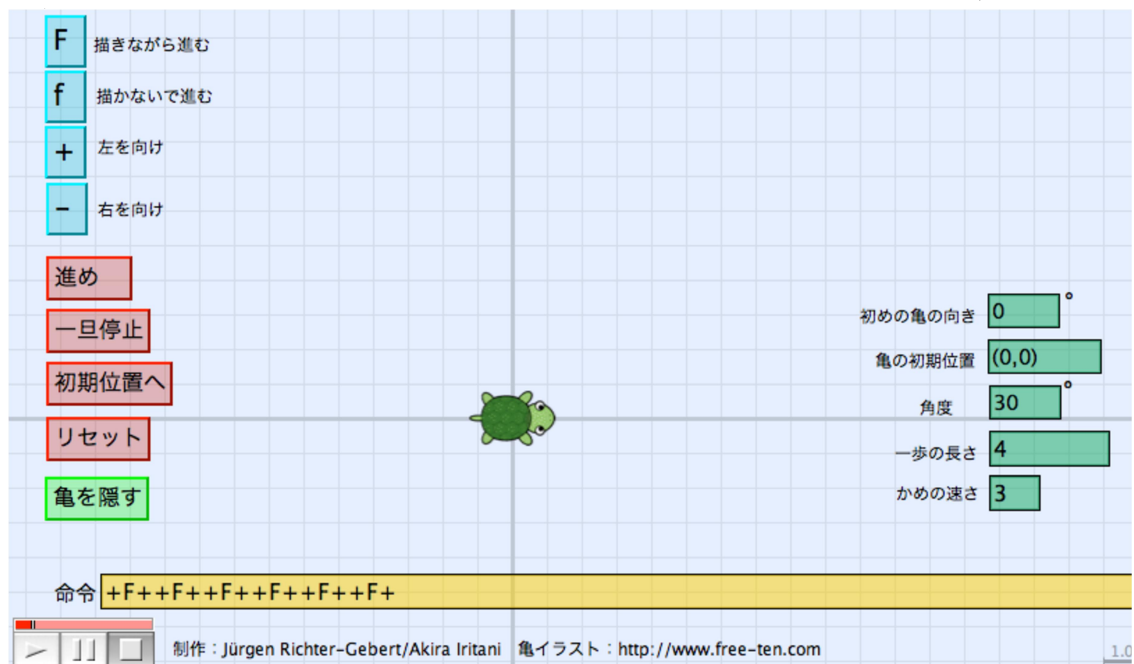
このシステムでは、Initiaizationスロットの関数定義はほとんど変更する必要はない。ユーザーは、Drawスロットの変数を設定して実行するだけである。とはいえ、ジェネレータはリストとして記述する必要があり、次に紹介するアプリケーションレベルでのしてステムのようなビジュアル性もない。一般の高校生には敷居が高いであろう。数学研究部の生徒などに向けたシステムである。

2.3 アプリケーションレベルでのシステム

数学研究会のような、もともとそれなりのモチベーションをもって参加している生徒は、プログラミング言語の習得に対しても自主的・意欲的に取り組むことができるので、「プログラムを書くことでシステムを構築し、それを利用する」ことができるが、一般の生徒にとってはこれは大きなハードルである。そこで、タートルへのコマンド群だけを与えれば描画ができるようなソフトウェアを制作した。ユーザーは画面上の入力エリアにコマンドを入力して実行する。そのためには、ユーザーインターフェースをどうするかが問題であるが、CindyScriptでは画面上に入力した文字列を取得することができ、また、作図機能で描いたスライダを利用して変数の値を変化させることができるので、これを利用する。

次の図がその画面である。右側に1歩の長さや角度などの設定があり、それぞれのボックスをクリックすると変更できる。下段の命令文字列も同様である。

左側はボタンで、これらのボタンを押すことで亀を動かすことができる。



次に、置き換えシステムを追加したのが次の画面である。イニシエータと変換規則を入力する。ここでは、Fだけを置き換える簡単なものを想定している。

この画面は、亀のグラフィックスに置き換えシステムを追加した様子を示しています。左側のメニューには、F（描きながら進む）、f（描かないで進む）、+（左へ向く）、-（右へ向く）、[（ここを覚える）、]（ここに戻る）などのコマンドが並び、下部には「進め」「一旦停止」「初期位置」「リセット」「亀を隠す」などの操作ボタンがあります。右側のパネルには、初期の亀の向き（0°）、亀の初期位置（(-2,0)）、向く角度（60°）、一歩の長さ（2）、亀の速さ（3）、置き換え回数（1）が設定されています。下部の黄色い入力欄には、イニシエータとして「F++F++F++」、F==>として「F+F--F+F」が入力されています。右上隅には「F+F--F+F++F+F--F+F++F+F++F+F++」という置換後の文字列が表示されています。

さらに高度な置き換えシステムが次の図である。それぞれの命令文字に対して置換規則を定めることができる。1歩の長さや置換回数はスライダで設定するようになっており、簡単にシミュレーションを実行できる。

この画面は、より高度な置き換えシステムを示しています。左側のメニューは基本的なコマンドで構成されています。右側のパネルには、初期位置（(0,0)）、初期の向き（0°）、長さ（0.8）、回転角（60°）、置換回数（2）、亀の速さ（3）が設定されています。下部には「スタート」「リセット」「亀を隠す」の操作ボタンがあります。中央には、イニシエータと置換規則を定義する表が示されています。

イニシエータ	置換規則
A	F
f	f
A->	B-A-B
B->	A+B+A
S->	S
R->	R
L->	L

3. 高等学校における利用例

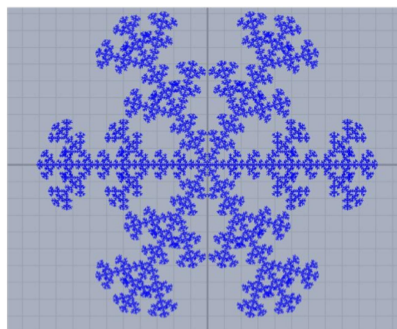
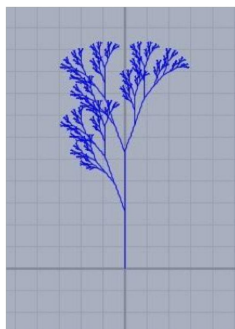
3.1 指導環境

静岡県立磐田南高校は2003年度からスーパー・サイエンス・ハイスクール(SSH)に指定されている。1学年に理数科1クラス、普通科7クラスの規模で、2011年度からは学校設定教科「SSH講座」として、理数科では「磐南スーパーサイエンス」、理数科では「磐南サイエンス探究」という科目を独自に設定し、Mathematicaを利用した曲線描画などの内容の授業を行っている。また、科学系部活動に数学研究部はないが、SSH研究会としてサークル活動的なグループを作って活動する素地が用意されている。このようなことから、通常の数学の授業ではなかなか踏み込むことのできない題材も取り上げることができる環境にある。そこで、まず、SSH研究会の数学部会を立ち上げて生徒に研究させた後、その成果をもとに、磐南サイエンス探究の授業において教材開発を行った。

3.2 SSH研究会数学部会での活動

2011年度に、第3期のSSH指定校として新規事業を立ち上げたのを機に、SSH研究会数学部会への参加者を1年生から公募し、4名の参加者を得た。彼らひとりひとりにCinderellaを渡し、L-Systemを題材とした課題を示して家と学校でできるときに研究を進めさせた。タートルグラフィクスとL-Systemについては、生徒の研究の進捗状況に合わせてWeb上で解説を行った。この手法は、理数科2年生の課題研究でも行っていたもので、Web上の掲示板を使ってヒントを与えたり、生徒の質問に答えるものである。その結果、11月のSSH成果発表会において、8月から10月までの3ヶ月間で行った研究の成果を発表することができた。CindyScriptを用いたプログラミング言語レベルのシステムを使ったものである。

下図左は、生徒が「パセリ」と名付けた植物の成長モデル、右は「雪印」と名付けた自己相似形のフラクタル絵柄である。



3.3 授業での指導例

SSH研究会数学部会での成果から、2012年度に、普通科2年生の磐南サイエンス探究B「数学シミュレーション」でフラクタルを取り上げた。フラクタルは、高校数学のカリキュラムにはない内容だが、大学入試問題にはコッホの雪片曲線が過去に極限の問題として出題されている。また、タートルグラフィクスにより、平面図形についての感覚を磨いたり、三角関数などを現実的に応用する場面を設定することができる。

3.3.1 タートルグラフィクスによる多角形の描画とその応用

タートルグラフィクスの考え方を学ぶには、多角形の描画が入門として最適であろう。凸型多角形と星形多角形をタートルグラフィクスで描く。使用する命令も、F、+、- の3つだけである。向きを変える角度、1歩の大きさなどは画面上のボックスに入力する。ここには、三角関数も入力できる。これを応用して、円周率の近似値を求めさせた。ただし、面積ではなく、多角形の周長を用いた。円に内接する多角形と外接する多角形を描く。このとき、タートルの1歩の長さとして多角形の1辺の長さを指定するので、これを用いて周長が計算できる。数値計算はCindyScriptでできるが、ここではCindyScriptは使わない前提なので、Mathematicaを用いさせた。(右下) 計算式の正誤をタートルグラフィクスで確かめるという算段である。

The screenshot shows a turtle graphics environment. On the left, there is a command list with the following items:

- F 描きながら進む
- f 描かないで進む
- + 左を向け
- 右を向け
- 進め
- 一旦停止
- 初期位置へ
- リセット
- 亀を隠す

At the bottom, a command input field contains: `+F++F++F++F++F++F++F++F++F++F++F++F++F++F++F++F+`

On the right, the control panel has the following settings:

- 初めての亀の向き: °
- 亀の初期位置:
- 角度: °
- 一步の長さ:
- かめの速さ:

In the bottom right corner, a Mathematica calculation box shows:

```
In[1]:= N[12 * 8 * Sin[Pi / 12]] / 8
Out[1]= 3.10583
In[2]:= N[12 * 8 * Tan[Pi / 12]] / 8
Out[2]= 3.21539
```

At the bottom of the interface, it says: 制作: Jürgen Richter-Gebert/Akira Iritani 亀イラスト: http://www...

3.3.2 自己相似形のフラクタル図形の描画

置き換えシステムを用いて自己相似な図形を描かせる。まず、多角形を描くのに、同じような命令を続けなくても、置き換えシステムを用いれば簡単に表現できることを学ぶ。次に、フラクタルの例として、コッホの雪片曲線をあげる。また、シェルピンスキーのギャスケットやヒルベルト曲線を紹介し、試行錯誤でよいので置き換えシステムを使って図形を描くことを課題とする。ここでは、フラクタルに対する興味づけが目的なので、論理的な設計までは要求をしない。線の長さの極限值や、凸型図形になったときの面積の極限值の計算は次の段階とする。また、平面充填やフラクタル次元にも踏み込まなかった。（高校生にとっては内容が高度である）

次に示すのは、生徒が描いた図の一例である。テキストの例（シェルピンスキーのギャスケットなど）を改造したレベルのものではあるが、高校生としてはこの程度ができればよいだろう。

イニシエータ: $F+F+F+F+F+F+F+F+F+F$
 置換規則 $F \rightarrow$ $F+F+F+$
 $f \rightarrow$ f
 $A \rightarrow$ $B-A-B$
 $B \rightarrow$ $A+B+A$
 $S \rightarrow$ S
 $R \rightarrow$ R
 $L \rightarrow$ L

初期位置: $(10,0)$ 初めの向き: 0°
 長さ: 1.5
 回転角: 30° 置換回数: 2
 亀の速さ: 4

スタート リセット 亀を隠す

置換規則 $F \rightarrow$ $SS-FF-SS-FF-SS-FF-SS-FF$
 $f \rightarrow$ FFF
 $f \rightarrow$ f
 $A \rightarrow$ $A++B$
 $B \rightarrow$ $B++A$
 $S \rightarrow$ $F-S+S+S-F$
 $R \rightarrow$ R
 $L \rightarrow$ L

初期位置: $(-3,8)$ 初めの向き: 0°
 長さ: 0.2
 回転角: 120° 置換回数: 3
 亀の速さ: 4

スタート リセット 亀を表示

4. 3Dタートルグラフィクス・複素平面によるフラクタルについて

4.1 3Dタートルグラフィクス

Lindenmayerの「The Algorithmic Beauty of Plants」には、3Dのタートルグラフィクスコマンドも用意してある。平面の場合と異なるのは回転だけなので、+、-の代わりに次のものを用いる。

```

+      : z 軸周りに正の回転
-      : z 軸周りに負の回転
&     : y 軸周りに正の回転
^     : y 軸周りに負の回転
\     : x 軸周りに正の回転
/     : x 軸周りに負の回転
|     : z 軸周りに180度反転

```

3Dグラフィクスは各軸周りの回転行列を用意し、透視変換を行うもので、次のようなわずかなスクリプトで実現できる。

```

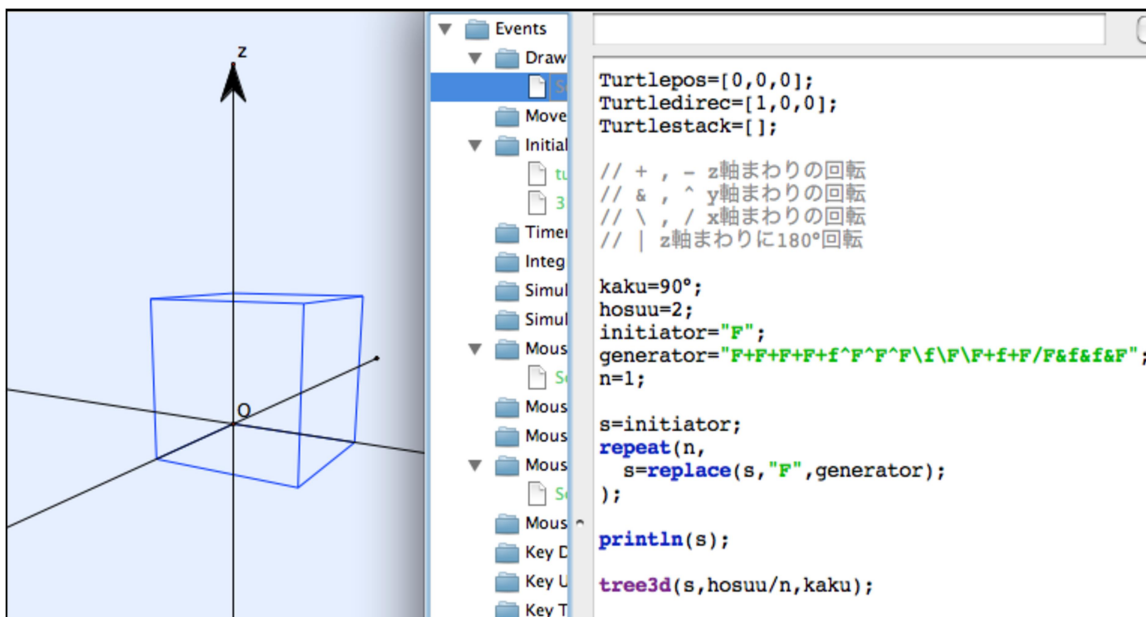
// 角  $\theta$  の回転行列
rtx( $\theta$ ):=[[1,0,0],[0,cos( $\theta$ ),-sin( $\theta$ )],[0,sin( $\theta$ ),cos( $\theta$ )]];
rty( $\theta$ ):=[[cos( $\theta$ ),0,sin( $\theta$ )],[0,1,0],[-sin( $\theta$ ),0,cos( $\theta$ )]];
rtz( $\theta$ ):=[[cos( $\theta$ ),-sin( $\theta$ ),0],[sin( $\theta$ ),cos( $\theta$ ),0],[0,0,1]];

// 回転行列初期値
mat=rtx(pi/12)*rty(-2*pi/3)*rtx(-pi/2);

viewpoint=9.001; // 視点のy座標 整数にしないこと
zoom=15;        // 拡大率

// 透視変換。使い方は p=[1,2,3] という3次元座標を
// A.xy=map(p) で2次元座標にしてAの(x,y)座標に返す
map3d(p):=(
  p=mat*p;
  mp=p/(viewpoint-p_3);
  [mp.x,mp.y]*zoom;
);

```



これを用いると、3Dの植物成長モデルを描画することができる。しかし、高校の授業の中での活用となると今後研究していく必要があるだろう。

4.2 複素数平面でのフラクタル

2012年度入学生より実施されている学習指導要領では、数学Ⅲに複素数平面が復活した。（なお、教科書では「複素平面」ではなく「複素数平面」という用語が以前から用いられている）CindyScriptには、点の座標と複素数を相互に変換する関数 `complex()` と `gauss()` が用意されており、複素数平面を簡単に扱うことができる。

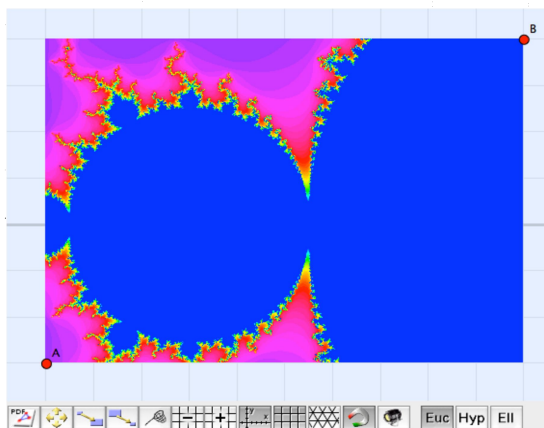
これを用いて、複素数平面の基礎を学び、さらに応用として複素関数を用いた自己平方フラクタルの描画をおこなうことが可能である。MathematicaのDensityPlot関数に相当する関数 `colorplot` が用意されており、次のような数行のスキプトで描画ができる。描画範囲は画面上の2点A,Bをドラッグすることにより変更できる。

==== マンデルブロ集合 ====

```

f(x,n):=(
  c=complex(x);
  z=0;
  k=0;
  while(|z|<2.0 & k<n,
    z=z^2+c;
    k=k+1;

```



```
);
col=k/n;
);

// 動かすモードで点A,Bを設定
colorplot(hue(f(#,60)+0.66),A,B,pxlres->1);
=====
```

本年度（2013年度）の2年生が新学習指導要領で学ぶ最初の学年であるので、マンデルブロ集合やジュリア集合などの複素関数によるフラクタルを紹介することができるだろう。現在、その教材を開発中である。

参考文献

- [1] Przemyslaw Prusinkiewicz , Aristid Lindenmayer, 「The Alogorithmic Beauty of Plants」 , Springer -Verlag ,1990
- [2] J.Mishra , S.Mishra, 「L-System Fractals」 , Elsevier Science, 2007
- [3] 佐藤幸悦, 「LOGOで学ぶCGと複素数の世界」, 森北出版, 1995
- [4] 淵上季代絵, 「フラクタルCGコレクション」, サイエンス社, 1987
- [5] 安居院猛,中嶋正之,木見尻秀子, 「C言語による3次元コンピュータ・グラフィクス」, 昭晃堂, 1987
- [6] 山口大学, 「情報処理テキストWeb版ーMathematicaの使い方ーフラクタル図形を作る」, http://www.edu.yamaguchi-u.ac.jp/~mis/www-page/mis/kaisetu/note_text_2003/index.html