

テキスタイルデザインのパターン生成展開手法の研究

開発技術部 伊藤通敏、山田 圭

1. はじめに

テキスタイルデザインの高度化と効率化のためには、従来からの感性主導型のデザイン開発のほかに、数式や幾何学を用いたシステマチックなパターン生成の手法を導入する必要がある。近年のコンピュータの高性能化とソフトウェアの進歩により、これらを実現するための複雑な方程式の演算処理やグラフィック表示がパソコンで十分可能になってきた。そこでコンピュータを使用して、テキスタイルデザインに適した新規性のあるパターンを生成展開する手法について研究した。

使用したソフトウェアは近年技術計算だけでなく、自然科学や社会科学の広い分野で利用されている数式処理システム Mathematica である。本研究では、点、線、多角形等の各種幾何図形を表示する数式を組み立て、Mathematica のグラフィックス機能を使用してその生成展開例を作成した。

またカオスやフラクタルなどの複雑系の方程式の反復計算による図形生成と、それらのパターンのテキスタイルデザインへの応用についても検討した。

```
Show[Graphics[{Hue[0], PointSize[0.1],  
Point[{0, 0}]}], Axes -> True,  
AspectRatio -> Automatic]
```

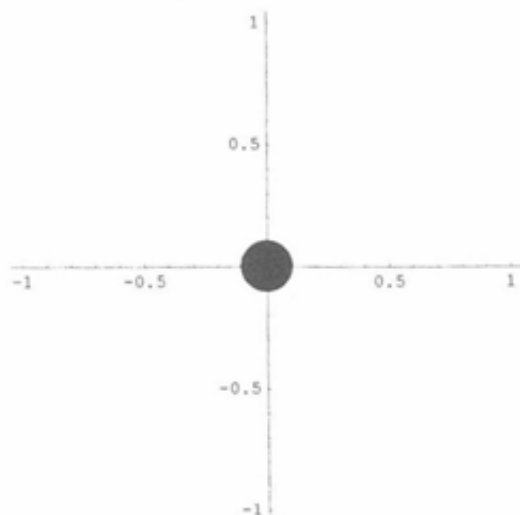


図1 点の作図

2. 点列の構成

Mathematica では扱う対象をすべて数式で表す。図形を表示するためには関数 `Show[Graphics[ ]]` を用いる。`[ ]` の中には種々のグラフィックスの要素が入る。例えば点を描く場合にはこの中に点の色と大きさそして座標値が入る。例を図1に示す。Hue[0]は色の指定であり、表1のように0から1までの値をとる。この場合は赤色で描かれることが分かる。なお、この指定を省略すると黒色

表1 色相対応表

Hue の値	0	0.2	0.4	0.6	0.8	1
色 名	赤	黄緑	緑	青	赤紫	赤

Hue の値は端数でも可 (例 0.08 橙)

で描画される。PointSize[0.1]は点の大きさを表しており、半径が表示エリアの横幅の0.1倍という意味である。Point[{0, 0}]は打つべき点のx座標値が0、y座標値が0であることを示している。Axes -> True は座標軸を表示させるための、AspectRatio -> Automatic は縦と横の目盛りを同じにするためのオプションである。

```
Show[Graphics[Table[{PointSize[0.04],  
Point[{i, j}]}],  
{i, 0, 1, .2}, {j, 0, 1, .2}]],  
Axes -> True, GridLines -> Automatic,  
AspectRatio -> Automatic]
```

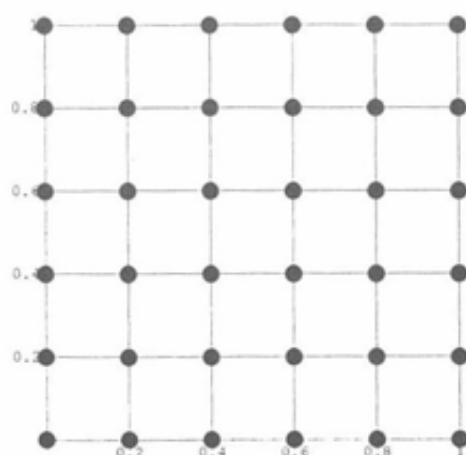


図2 点列の作図

```

a = Table[{Hue[0.5], Point[{i, j}]},
  {i, 1, 10}, {j, 1, 10}];
b = Table[{Hue[0.7],
  Point[{i + .2, j + .2}]}, {i, 1, 10},
  {j, 1, 10}];
Show[Graphics[{PointSize[0.05], a, b}],
  AspectRatio -> 1]

```

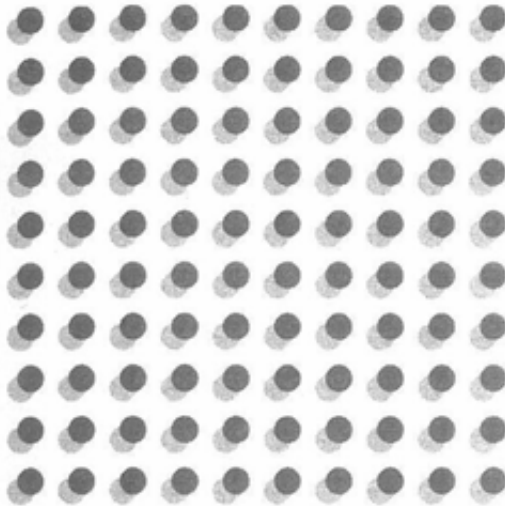


図3 点列のシャドウ表現

1つだけの点ではなく、たくさんの点を打ちたいときは図2のようにTable[]を使う。この場合Table[]でPoint[i, j]に6×6の位置を与えている。なお{i, 0, 1, .2}, {j, 0, 1, .2}はiとjについて0から1まで0.2刻みでという意味である。

図3はTable指定を複数使うことによって同じパターンを色を変えて平行移動させ、シャドウ表現を試みたものである。

また大きさと色は点ごとに指定することもできる。図4に見るように、位置は同じでも点の大きさと色を変えることによって表現力が増す。

3. 線による構成

線分は関数Line[]で指定する。Line[{x₁, y₁}, {x₂, y₂}]で座標x₁, y₁と座標x₂, y₂が直線で結ばれる。線の太さはThicknessで指定する。破線を描くにはDashing[{]}を用いる。{ }の中に線分の長さや空隙の部分の長さの値を入れる(図5)。

点の時と同様Table指定を複数使うことによって同一画面に方向の異なる線を描くことができる。図6~8はこの手法によって描いたものである。図6は1点(中心点)を通る線でTable aで

```

Show[Graphics[Table[{PointSize
  [0.1 (i + j)], Hue[(i + j) / 2],
  Point[{i, j}]}, {i, 0, 1, .2},
  {j, 0, 1, .2}]],
  AspectRatio -> Automatic]

```



図4 点列のサイズと色相変化

天地を結ぶ対角線、Table bで左右を結ぶ対角線を指定している。図7は縦横斜め線の組み合わせであり、Table aで右上がりの斜線、Table bで左上がりの斜線、Table cで垂直線、Table dで水平線を指定している。図8は2点から放射する破線の線列である。破線の長さ指定の数字は描画される図の巾を1としたときの値であるので、同心円状のパターンが現れる。

```

Show[Graphics[{Thickness[0.02],
  Line[{0, 1}, {1, 1}],
  Thickness[0.01],
  Line[{0, 0.5}, {1, 0.5}],
  Thickness[0.005],
  Dashing[{0.1, 0.05}],
  Line[{0, 0}, {1, 0}]}]]]

```

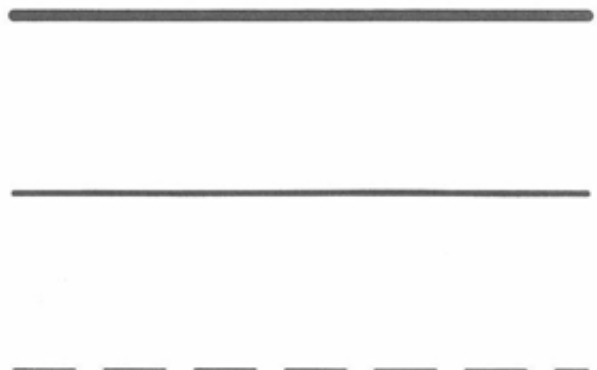


図5 線の作図

```

a = Table[{Hue[i / 10], Line[{(i, 50 - k),
    {50 - i, k}]}], {i, 0, 50}, {k, 0, 50, 50}];
b = Table[{Hue[i / 10], Line[{(k, 50 - i),
    {50 - k, i}]}], {i, 0, 50}, {k, 0, 50, 50}];
Show[Graphics[{Thickness[.005], a, b}],
    AspectRatio -> 1];

```

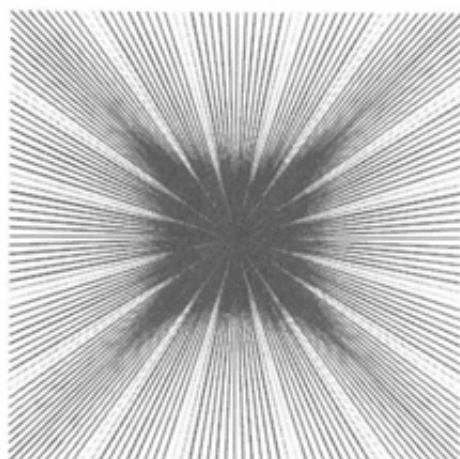


図6 1点を通る線

```

a = Table[
    {Hue[i / 10], Line[{(k, 30 - i), {i, 30 - k}]}],
    {i, 0, 30}, {k, 0, 30, 30}];
b = Table[{Hue[i / 10], Line[{(k, i), {i, k}]}],
    {i, 0, 30}, {k, 0, 30, 30}];
c = Table[{Hue[i / 10],
    Line[{(i, 0), {i, 30}]}], {i, 0, 30}];
d = Table[{Hue[i / 10],
    Line[{(0, i), {30, i}]}], {i, 0, 30}];
Show[Graphics[{Thickness[.005], a, b, c, d}],
    AspectRatio -> 1];

```

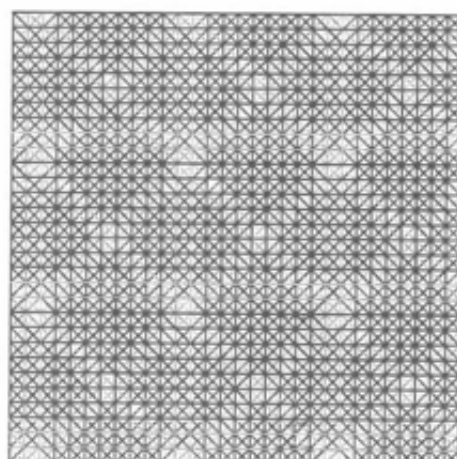


図7 線の複合

```

a = Table[{Hue[i], Line[{(0, 0), {i, 1}]}],
    {i, 0, 1, 0.04}];
b = Table[{Hue[i],
    Line[{(1, 0), {i, 1}]}], {i, 0, 1, 0.02}];
Show[Graphics[{Thickness[.005],
    Dashing[{0.2, 0.05}],
    a, b}], AspectRatio -> 1];

```

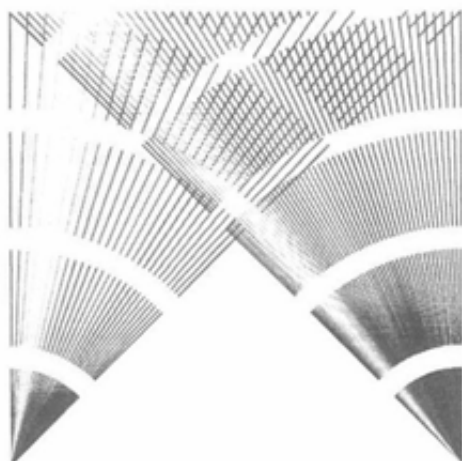


図8 破線の放射

```

r = 0.3; unit[x_, y_] :=
(Circle[{x, y}, r, {0, Pi / 2}],
    Circle[{x + 1, y}, r, {Pi / 2, Pi}],
    Circle[{x + 1, y + 1}, r, {Pi, 3 Pi / 2}],
    Circle[{x, y + 1}, r, {3 Pi / 2, 2 Pi}],
    Circle[{x + 0.5, y + 0.5}, r]);
p = Table[unit[x, y], {x, 0, 1}, {y, 0, 1}];
Show[Graphics[p], Axes -> True,
    GridLines -> Automatic,
    AspectRatio -> Automatic,
    PlotRange -> {{0, 1}, {0, 1}}];

```

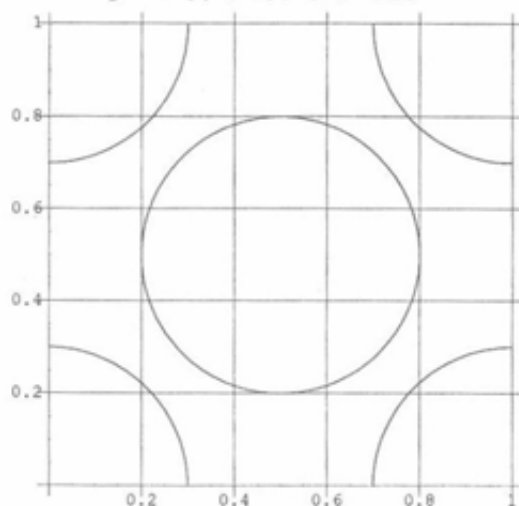
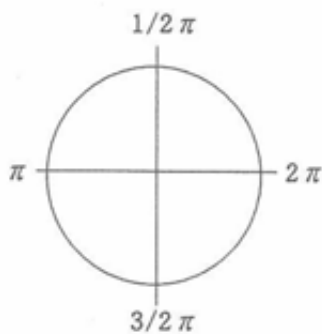


図9 円と円弧の作図

4. 円による構成

円を描くには関数 Circle を使う。Circle $[(x, y), r]$ で中心が (x, y) で半径 r の円を表す。また円弧も Circle で描ける。Circle $[(x, y), r, [\theta_1, \theta_2]]$ で中心が (x, y) 、半径 r 、中心角 $\theta_2 - \theta_1$ の円弧を表す。図9に円と円弧の作図を示す。なおこの例では後の作図の便のために半径の値を最初に与え、描画する位置と範囲を5つのユニットからなる Table で指定する手法をとった。また中心角を $\text{Pi}(\pi)$ で指定している。 π と角度の関係は図10に示すとおりである。

図10 π と角度

このプログラムで円の半径の値をいろいろ変えると図11のような面白い図形が描ける。半径0.5は日本の伝統文様の七宝繋ぎである。数値の変化に伴って、円弧の交差によって生まれる複合的なパターンが様々に変化する。

円の作図においては、Circle を Disk と置き換えることによって円盤や扇形が描ける。図12の扇の右方向へのグラデーションは、扇の中心角の指定の際に π を暫増する x 座標の値で除すことによって生み出される。

5. 多角形による構成

多角形を描くには Polygon を使う。Polygon $[[\{x_1, y_1\}, \dots, \{x_n, y_n\}]]$ で各頂点の位置を指定する。最初の点 (x_1, y_1) から順番に結んで行き、最後の点 (x_n, y_n) と最初の点とが結ばれて多角形ができる(図13)。なお多角形の内部は塗り潰される。

図14は図13の多角形を図9と同じようにユニットを含む Table で指定することによって四方連続展開したものである。

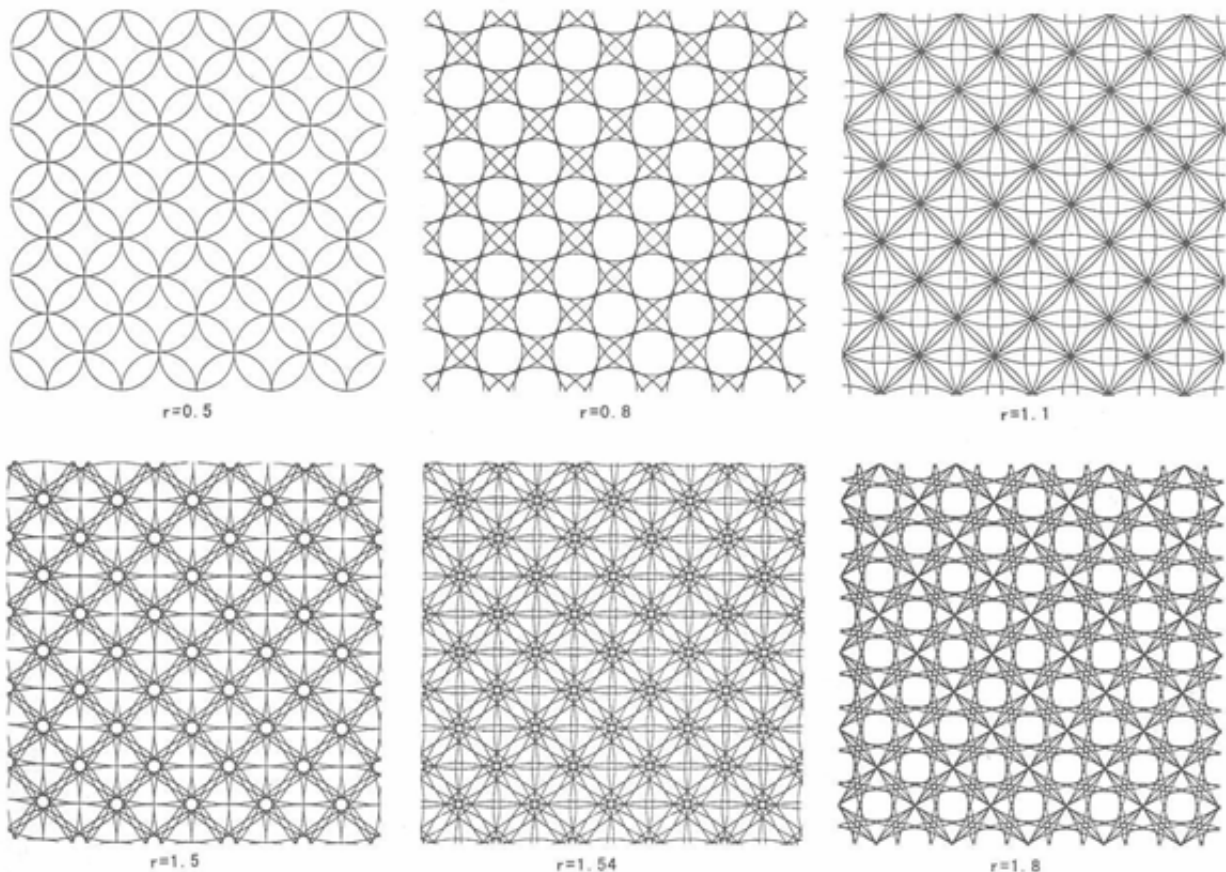


図11 円の半径とパターン変化

多角形の頂点の位置をある規則で動かし、連続させると変化のあるパターンを作ることができる。図15は図16Aで示す四辺形の頂点の位置を決めるユニットの原点、終点からの距離を一方で暫増、一方で暫減することによって、格子柄を作

```
Show[Graphics[Table[{Hue[i/5*j/5],
  Disk[{i, j}, 0.5, {0, Pi/i}]}],
  {i, 1, 5, 0.6}, {j, 1, 5, 0.6}]],
  AspectRatio -> Automatic]
```

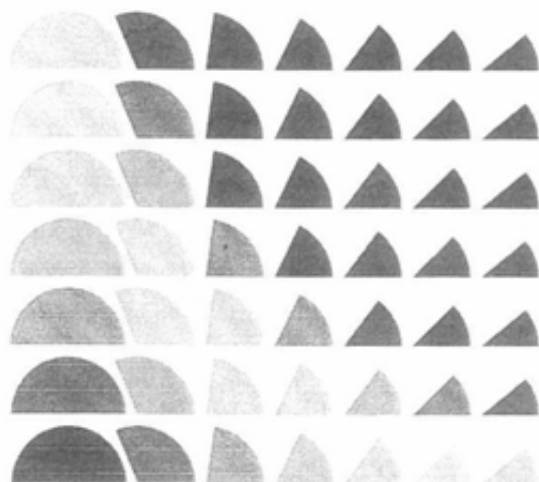
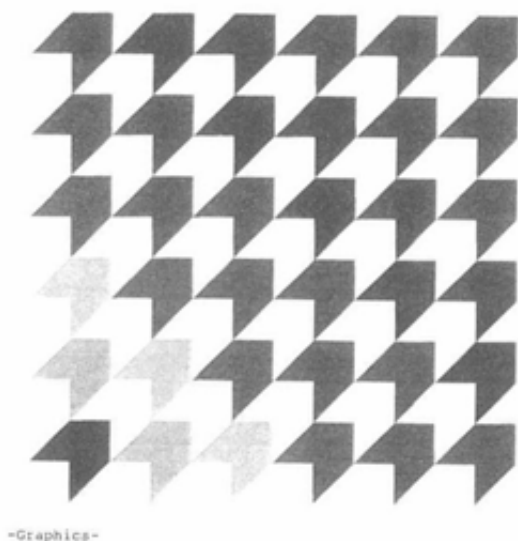


図12 扇のグラデーション

```
unit[x_, y_] := {Hue[(x+y)/10],
  Polygon[{(x+0.5, y), (x+1, y+0.5),
    (x+1, y+1), (x+0.5, y+1),
    (x, y+0.5), (x+0.5, y+0.5)}]}];
p = Table[unit[x, y], {x, 0, 5}, {y, 0, 5}];
Show[Graphics[p], AspectRatio -> Automatic]
```



-Graphics-

図14 多角形の連続

ったものである。

図17は図16Bで示す四辺形の各頂点の元の位置からの距離sを暫増することによって四辺形に回転を与えている。

```
Show[Graphics[{Hue[0.1],
  Polygon[{(0.5, 0), (1, 0.5), (1, 1),
    (0.5, 1), (0, 0.5), (0.5, 0.5)}]}],
  Axes -> True, GridLines -> Automatic,
  AspectRatio -> Automatic]
```

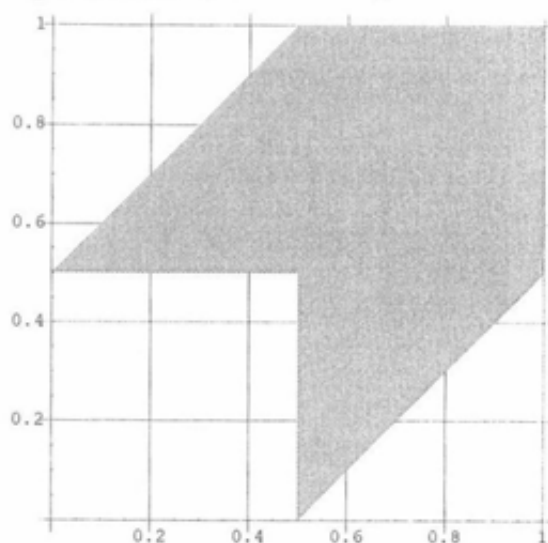


図13 多角形の作図

```
unit[x_, y_] := {Hue[0.9], Polygon[
  {(x+x/8, y+y/8), (x+1-x/8, y+y/8),
    (x+1-x/8, y+1-y/8),
    (x+x/8, y+1-y/8)}]}];
p = Table[unit[x, y], {x, 0, 8}, {y, 0, 8}];
Show[Graphics[p], AspectRatio -> Automatic,
  Axes -> True]
```

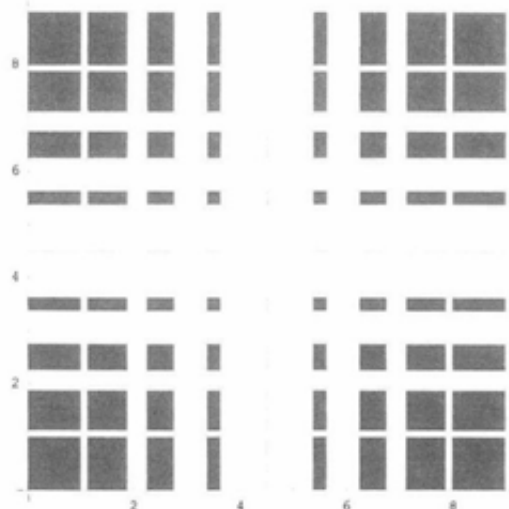


図15 四辺形の頂点移動による格子柄

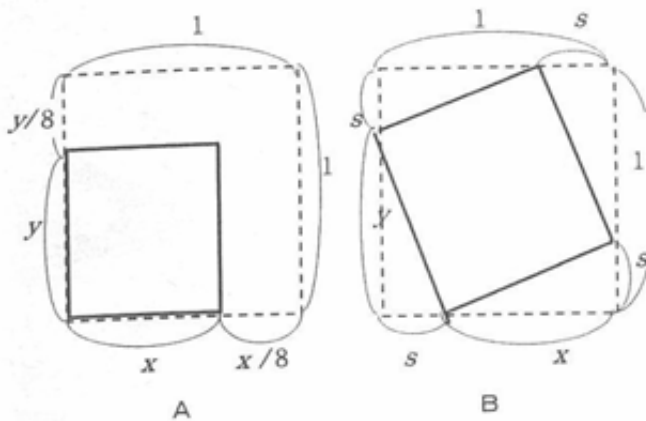


図16 四辺形頂点位置決め説明図

```
quad[s_, x_, y_] :=
  {{x+s, y}, {x+1, y+s}, {x+1-s, y+1},
   {x, y+1-s}}; n=8;
p = Table[Polygon[quad[i/n, i, j]],
  {i, 0, n-1}, {j, 0, n}];
Show[Graphics[{Hue[0.7], p}],
  AspectRatio -> 1, PlotRange ->
  {{1, 8}, {0, 7}}, Background -> Hue[0.15]]
```

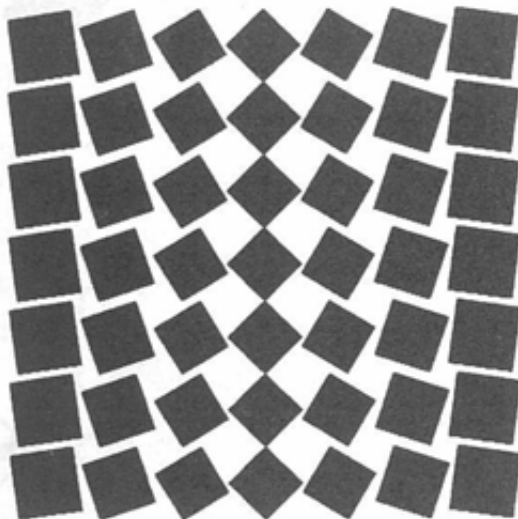


図17 四辺形の回転

6. カオス図形の生成

あらかじめ定めておいた計算手順を何度も繰り返し、計算結果の状態がどのように変化していくかを観察すると、複雑な現象に出会うことがある。カオスはこのような状況のもとに見いだされる。

数列の変化の規則を表す方法の1つとして、次の数が前の数からどのように定められるかを表す漸化式がある。方程式の解が直線に乗らない非線形の漸化式は、簡単な方程式であっても初期値の取り方や係数の値によっては繰り返し演算を行うと結果が複雑に分かれたり、ばらついたりする場

```
Module[{f, data}, f[{x_, y_}] :=
  {y + 0.6 x + 5 / (1 + x^2), -x};
data = NestList[f, {1, 0}, 10000];
ListPlot[data, AspectRatio -> Automatic,
  PlotStyle -> {PointSize[0.006]}]
```

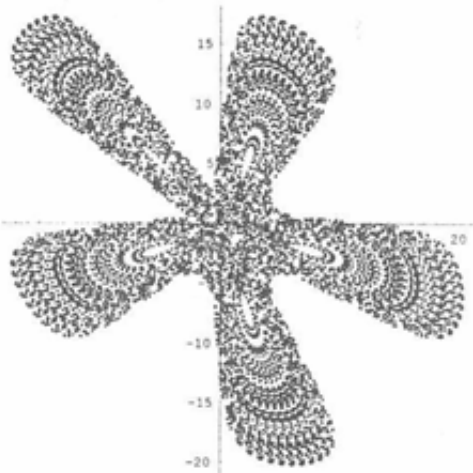


図18 カオス図形

合がある。これがカオスである。一例を次式に示す。

$$\begin{cases} x_{n+1} = y_n - 0.6x_n + \frac{5}{1+x_n^2} \\ y_{n+1} = -x_n \end{cases} \quad (1)$$

この式は平面上の点 (x_n, y_n) から点 (x_{n+1}, y_{n+1}) が定められる漸化式で、最初に与える点 $(x_1, y_1) = (1, 0)$ としてこの計算を10,000回繰り返した時、全ての点を図示すると図18のような図形が得られる。ここで係数0.6を k と置き、 k を一定間隔で連続的に変化させると、得られるパターンは図19のような展開を見せる。ただ1箇所のパラメータの変化によって、全く異なった図形が得られることが分かる。

カオス図形をもう一例示す。

$$\begin{cases} x_{n+1} = -y_n \\ y_{n+1} = x_n - k \sin(6y_n) \sin(10y_n + 10) \end{cases} \quad (2)$$

この漸化式で x, y の初期値を $(2, 0)$ とし、 k の値を0.1から0.3の範囲でいろいろ変えて演算を5,000回繰り返すと図20のように4弁の回転対称図形が得られる。

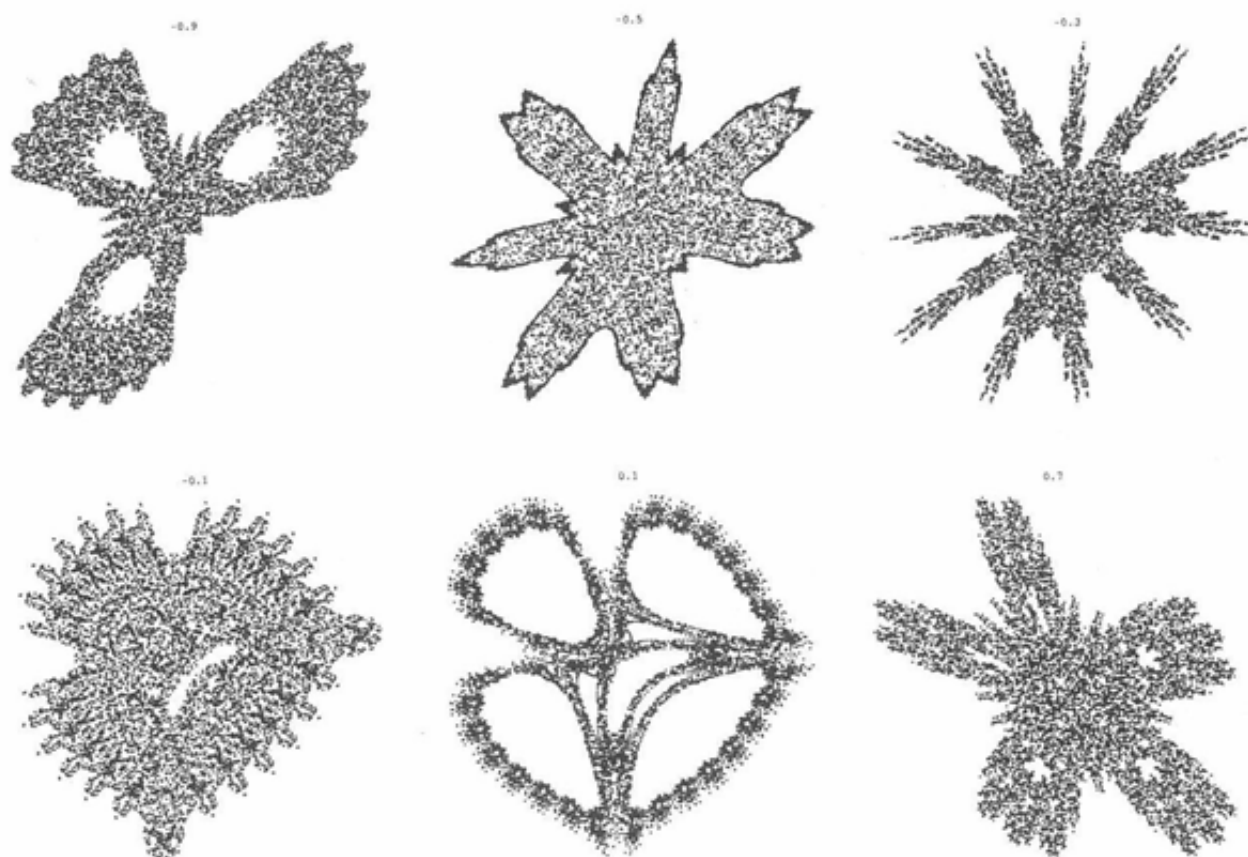


図19 (1) 式をもとにしたカオス図形の展開 (数字はkの値)

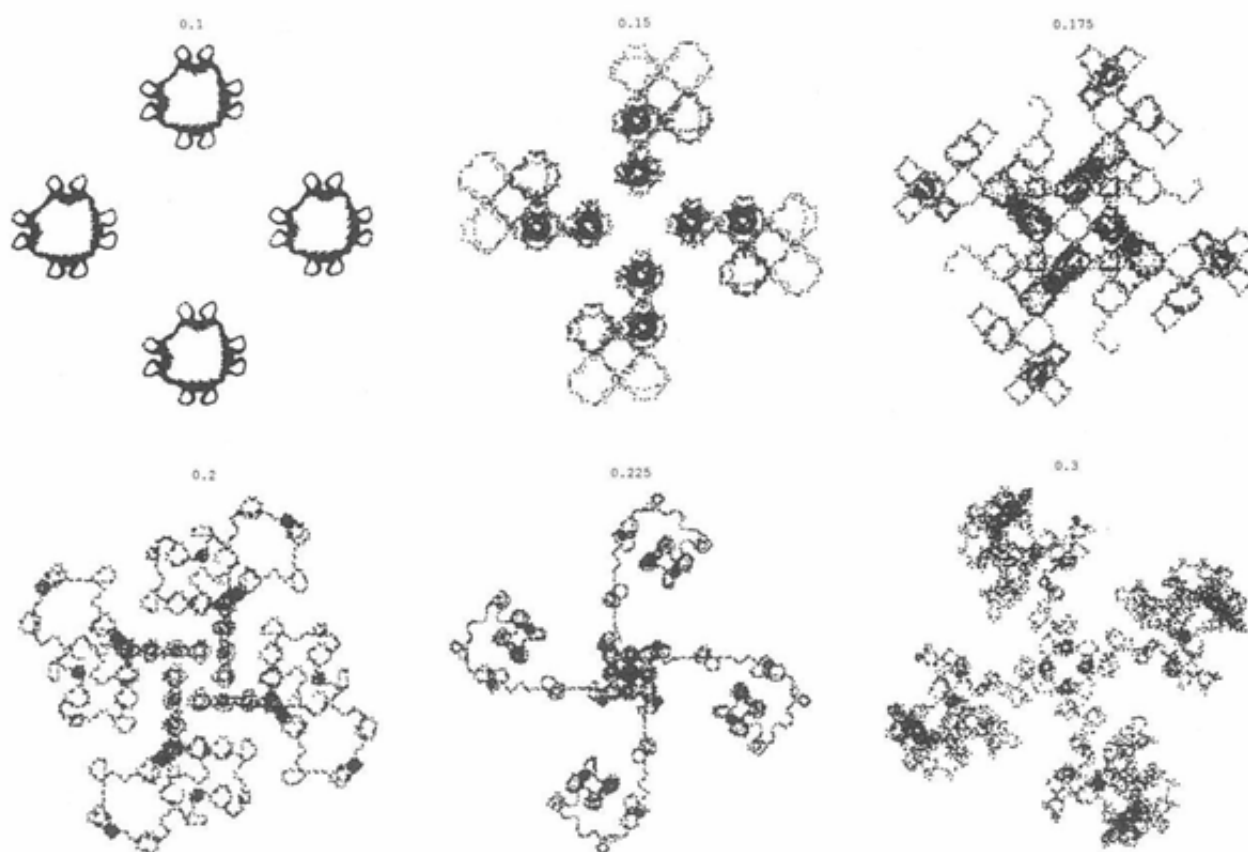


図20 (2) 式によるカオス図形の展開 (数字はkの値)

7. フラクタル図形の生成

フラクタルとは「断片」のことであり、フラクタル図形とは自己相似構造を持つ図形を指す。フラクタルを説明する場合によく引用されるコッホ曲線は、微分の極限として得られる図形が、どの部分をとってみても、それとそっくり同じ構造のものがいくらかでもミクロなスケールでその部分自身の中に含まれている (図21)。しかし得られる図形の面白さの点から見るとフラクタル図形はマンデルブロ集合図形とジュリア集合図形に代表される。

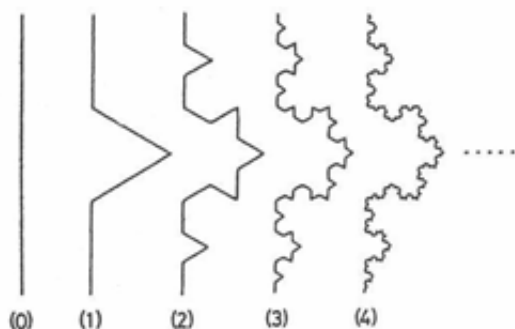


図21 コッホ曲線

7.1 マンデルブロ集合

フラクタル幾何学の創始者B. Mandelbrot (米)によってこのように呼ばれるようになったこの図形は (3) 式によってあらわされる。

$$Z_{n+1} = Z_n^2 + r \quad (3)$$

ここで取り扱う数を実数である限りこの計算を何回繰り返しても結果は予測のつく値となるが、複素数を用いると多様な変化が起きる。複素数とは実数と虚数の和 $x + yi$ であらわされる数で実数部 x と虚数部 y とから成り立っている。複素数は実数部を横軸、虚数部を縦軸とした複素平面上の点として表すことができる。この実数部、虚数部は正でも負でもよいし、整数でも小数でもよい。

マンデルブロ集合の計算は (3) 式の r に複素数を用い、 Z の初期値を0として始める。 Z_n は複素平面の上で原点からの距離が2より大きくなるとどんどん遠くへ行き、発散してしまう。50回くらいまでに2より大きくななければその後も遠くへは行かない。 Z_n が遠くへ行かないような (x, y) の集合がマンデルブロ集合と呼ばれる。

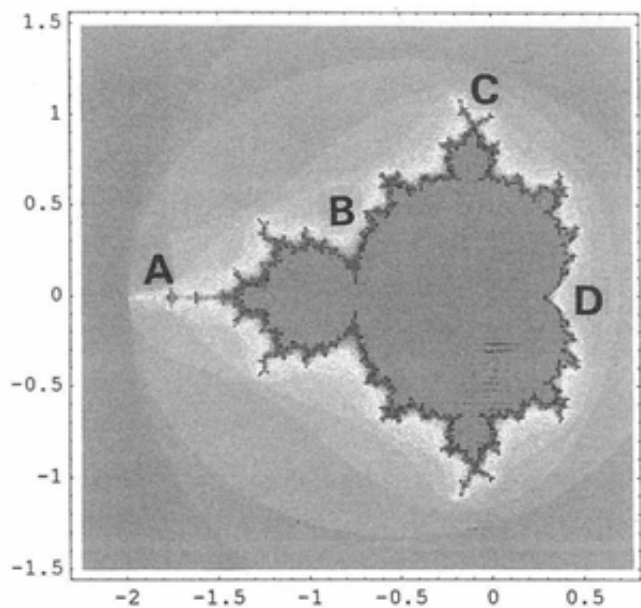


図22 マンデルブロ集合

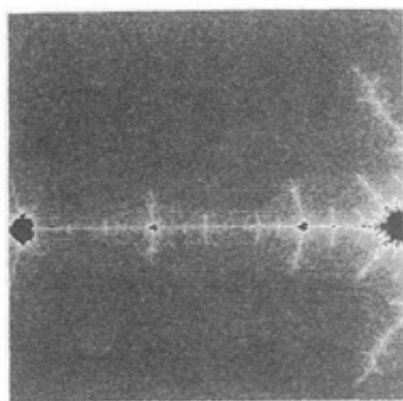
図22は (3) 式において Z の初期値を0として点 Z の絶対値が2を越えるまで反復演算を行い、その反復回数に応じてその点を色付けしたものである。図の黒く塗られた部分は50回の繰り返し演算において2より大きくならなかった領域であり、マンデルブロ集合である。

この場合、図形的に面白いのは、集合図形と発散する領域との境界の部分である。この部分を拡大して観察すると、無限に同じような図形が複雑に反復されているのが見られ、さらにその一部を拡大すると、また同じような図形が見られるという自己相似構造を持った図形が観察される。図23に図22のA~D各部の拡大画像を示す。

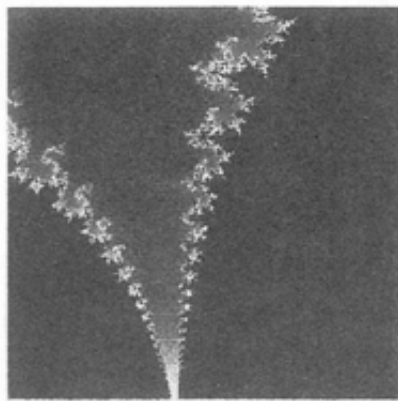
7.2 ジュリア集合

ジュリア集合もマンデルブロ集合と同じように (3) 式から生まれる。ジュリア集合は複素数の研究で有名な数学者M. Julia (仏) にちなんで付けられたもので、マンデルブロ集合が Z の初期値を0として複素数 r の値をいろいろ動かしてみるのに対して、ジュリア集合は加える複素数は一定にして、 Z を変化させたときの収束する点の集合を求めたものである。加える複素数を変えることにより集合の形態も異なる。

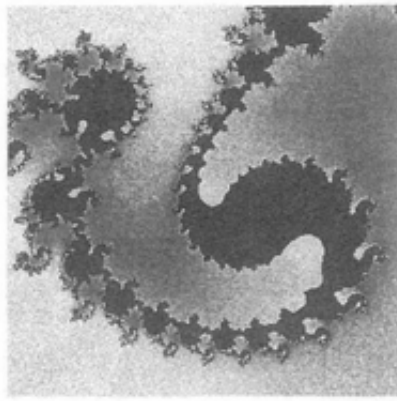
図24は加える複素数の実数部と虚数部の違いによって得られる図形がどのように変化するかをMathematicaを使用して実数部、虚数部とも-3



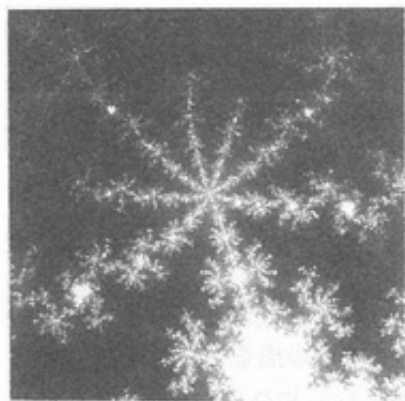
1 図22A部拡大



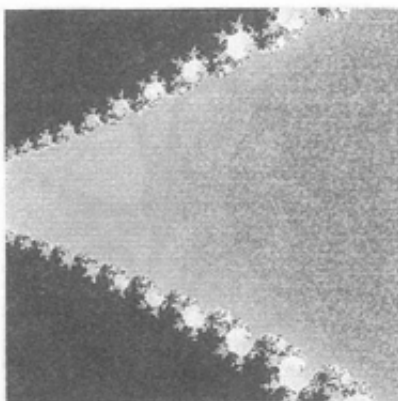
2 図22B部拡大



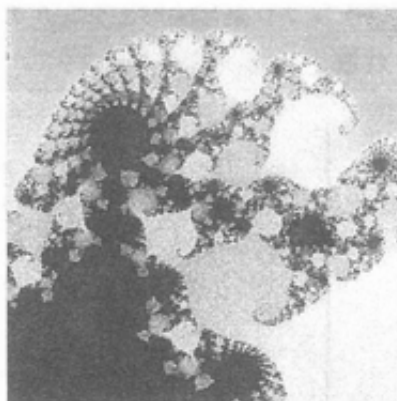
3 2の細部拡大



4 図22C部拡大



5 図22D部拡大



6 5の細部拡大

図23 マンデルブロ集合拡大図

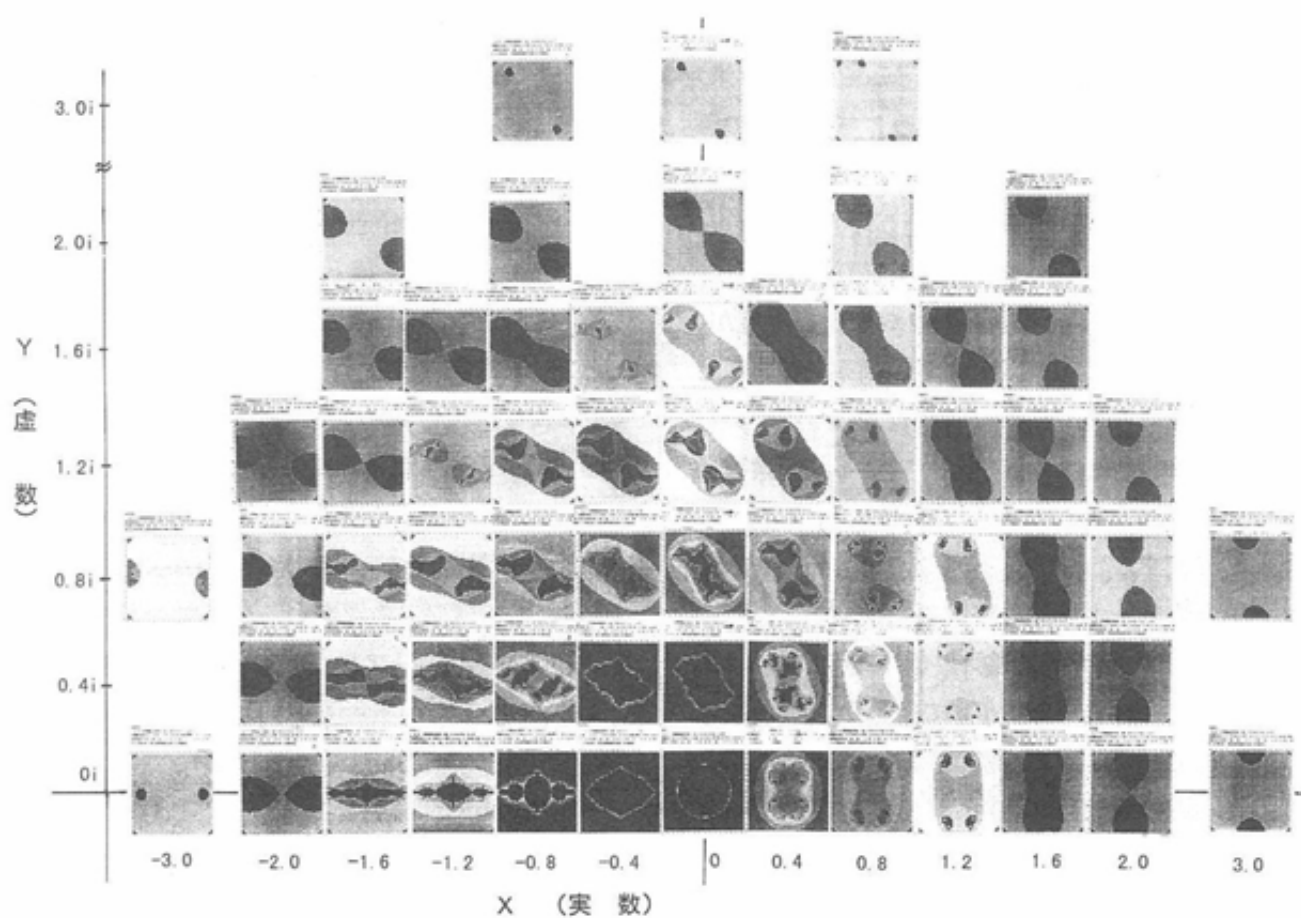


図24 ジュリア集合マップ

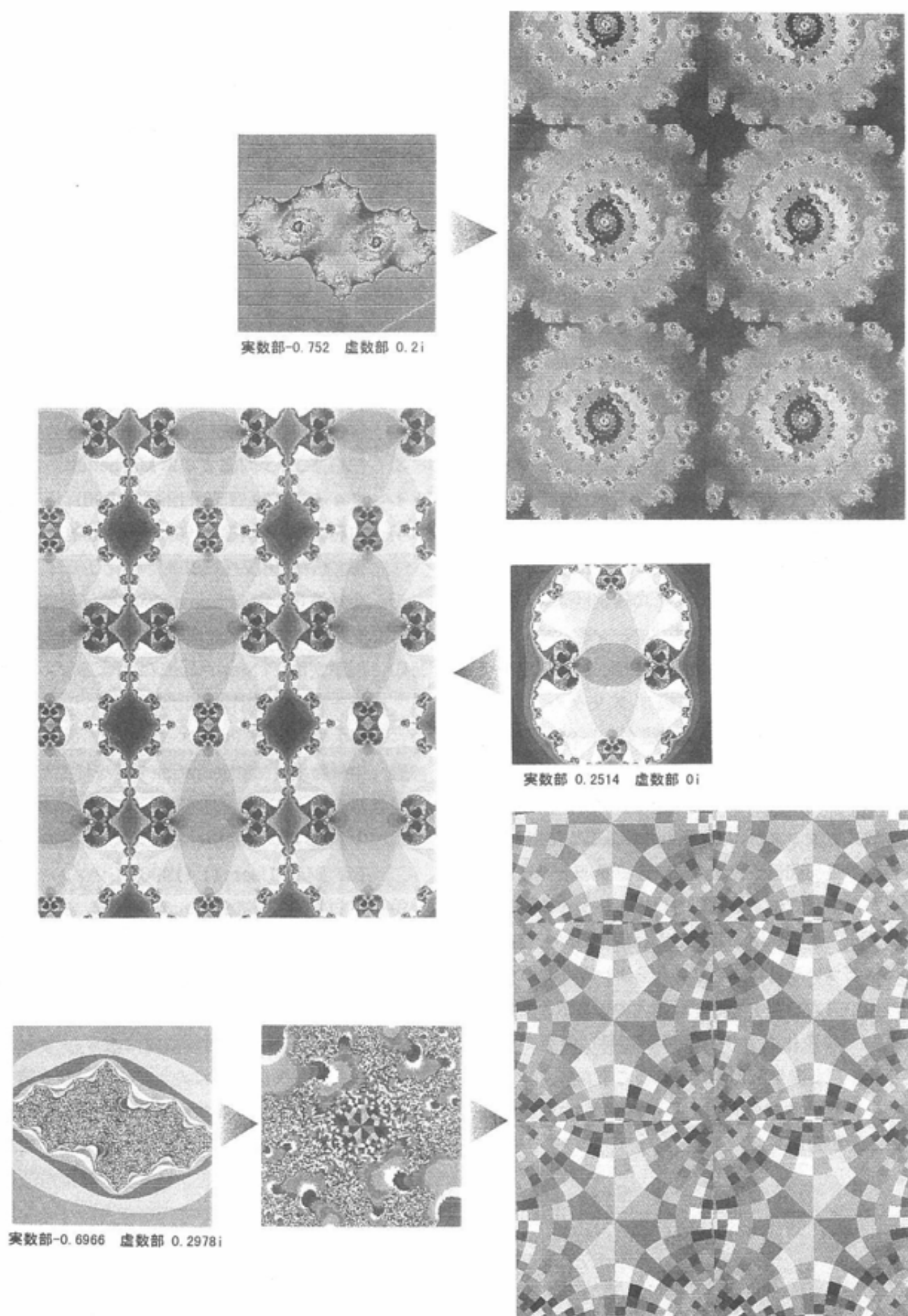


図25 フラクタル図形によるテキスタイルパターン

から3の範囲について0.4刻みで調べたものである。実数部は符号の正負によって形態が全く異なるが、虚数部は符号が異なっても図形が0i軸に対して鏡面对照形になるだけであることが分かったので下半分の負数部分は省略した。このマップはジュリア集合を使った図形生成のアウトラインを把握するためのガイドとして使用できる。

ジュリア集合もフラクタル図形であるので細部を拡大すると自己相似構造になっている。拡大してみても面白いのは実数軸で-1.2から0.8の範囲、虚数軸で0.8iから-0.8iあたりまでであり、複素平面上のこの範囲はマンデルブロ集合の黒い図形の範囲と重なる。

このマップを手掛かりにテキスタイルデザインとして応用できそうなパターンを選び、テキスタイルデザインの基本的なパターン展開である四方連続模様を作成した。図25に、もとになったパターンとその着目部分、そして連続展開をかけて出来上がったテキスタイルパターンを示す。なおフラクタル図形の生成においては、計算速度の点からフラクタル図形作成アプリケーションMandel Explorer及び「眼蔵くん」を使用した。

8. まとめ

数式処理システムMathematica等を使用して、各種の幾何図形やカオス、フラクタルといった複雑系の方程式による図形を作成し、テキスタイルデザインへの応用を検討した。所望の図形を得るにはどういう方程式を立てたらよいかという点に相当の労力を要するが、数値入力であるのでパラメータの変化により、図形が容易に変更できたり、予期しない図形が得られることが多く、新規性のあるテキスタイルデザインの開発に有効であることが分かった。

作成した図形は大半がカラーであるが、本報では印刷上の制約からモノクロの図版を掲載している。またできるだけ作図の基となったリスト（プログラム）を図に附したが一部はスペースの点から割愛せざるを得なかった。興味を持たれた方は本所までお問い合わせいただきたい。

文 献

- 1) S. Wolfram, Mathematica A System for Doing Mathematics by Computer (日本語版), アジソン ウエスレイ (1996)
- 2) 小林 道正, 小林 研, 「はじめてのMathematica」 BNN (1995)
- 3) 白石修二, 「例題で学ぶMathematicaグラフィックス編」, 森北出版 (1996)
- 4) 白石修二, 「例題で学ぶMathematica数学編」, 森北出版 (1995)
- 5) 小林道正, 「Mathematicaによる関数グラフィックス」, 森北出版 (1996)
- 6) 川瀬宏海, 「Mathematicaによるプレゼンテーション創作グラフィックス」, 東京電機大学出版局 (1997)
- 7) 中村勝平, Mathematicaで絵を描こう 1, Mac User, 10 (1995), ソフトバンク
- 8) 中村勝平, Mathematicaで絵を描こう 14, Mac User 11 (1996), ソフトバンク
- 9) 中村勝平, Mathematicaで絵を描こう 15, Mac User, 12 (1996), ソフトバンク
- 10) A. Peterson, 奥田訳「CGがひらく現代数学ワンダーランド」, 新曜社 (1990)
- 11) 伊東敬祐, 「カオスって何だろう」, ダイアモンド社 (1993)
- 12) 佐藤幸悦, 「LOGOで学ぶCGと複素数の世界」 森北出版 (1995)