College Analysis 参考資料 17

ー フラクタルビューア ー

福井正康・尾崎誠・石丸敬二*

福山平成大学経営学部経営学科
*福山大学経済学部経済学科

概要

我々は教育分野での利用を目的に社会システム分析に用いられる様々な手法を統合化したプログラム College Analysis を作成してきた。最近はグラフィックにも興味を持ち、３Ｄ表示を利用したいくつかのプログラムを開発してきた。この度はこれらのプログラムの中で、フラクタルビューアというプログラムを紹介する。これは２Ｄと３Ｄのフラクタルを描くツールであるが、既存の Mandelbrot 集合や充填 Julia 集合などを描く機能も持っている。

キーワード

College Analysis，社会システム分析，統計，OR，意思決定，フラクタル

URL:　http://www.heisei-u.ac.jp/ba/fukui/

１．はじめに

我々はこれまで、統計、経営科学、意思決定に関するプログラムを統合したソフトウェア College Analysis を作成してきたが、これに少し娯楽性を加えて、一般の人にも科学の楽しさを伝えられるプログラムにしたいと思うようになった。その試みの最初の課題は３Ｄであった。そのため、我々は３Ｄビューアという名前の３Ｄグラフィック表示用のユーティリティを作成し[1)]、最初に２変量関数グラフを表示するプログラムで利用した[2)]。

今回はこれを利用してフラクタル図形が描ける、フラクタルビューア３Ｄというプログラムを作成したので紹介する。また、３Ｄビューアは利用しないが、２次元のフラクタルを表示するフラクタルビューア２Ｄについても紹介する。

フラクタルビューア３Ｄは反復関数を利用する方法や回帰的方法を用いて描かれる３Ｄ画像を作成するツールである[1),2)]。フラクタルは、反復関数法では「点」の集合で表され、回帰的方法では「線」の集合で表される。これに対してフラクタルビューア２Ｄの回帰的方法では、図形は線と面の集合として表される。また、Mandelbrot 集合や充填 Julia 集合などのビューアもついており、複素数列の収束問題の学習と美しいフラクタル画像の観賞用となっている[3),4)]。これらに共通する再帰的方法では、容易にフラクタル画像が描けるように、簡単なフラクタル描画言語を作成した。この報告ではその利用法についても解説する。

ここで紹介した３次元フラクタルは、３Ｄビューアを用いて表示するため、面を組み合わせたような複雑な図形は作れないが、マウスによる回転や移動、色の変更や表示モードの切り替えなど、３Ｄビューアの機能はすべて利用できる[1)]。この論文では最初に、フラクタルビューア２Ｄ、次にフラクタルビューア３Ｄについて紹介する。

２．フラクタルビューア２Ｄ

フラクタルビューア２Ｄの実行画面を図 2.1 に示す。このプログラムは複素数列の収束問題を利用したフラクタルと再帰的な手法によるフラクタルの２種類を扱う。複素数列の収束問題は基本的に$z_n = f(z_{n-1}) + c$の形の代表的な力学系について、Mandelbrot 集合のように、z_0を固定してcの収束、周期振動、発散領域を調べる場合と、充填 Julia 集合のようにcを固定してz_0の収束、周期振動、発散領域を調べる場合を扱っている。収束と周期振動は複数の点の周りで生じることがあるので、よく利用される色付け方法である収束までのループの回数による色分けの他に、どの点に収束するのか、または振動の周期はいくらかによる色付け方法も加えてある。また、色付けに用いる色の種類は予め登録してある色パターンによって指定できる。収束、周期振動、発散を分けて表示したい場合は、それぞれ赤系の「Autumn」、緑系の「Spring」、青系の「Aqua」等の色パターンを選ぶとどの部分がどんな特性を持っているのかをはっきり見ることができる。これらの色パターンのサンプルは、メニュ

ー［ヘルプー色パターン］を選択すれば表示される。

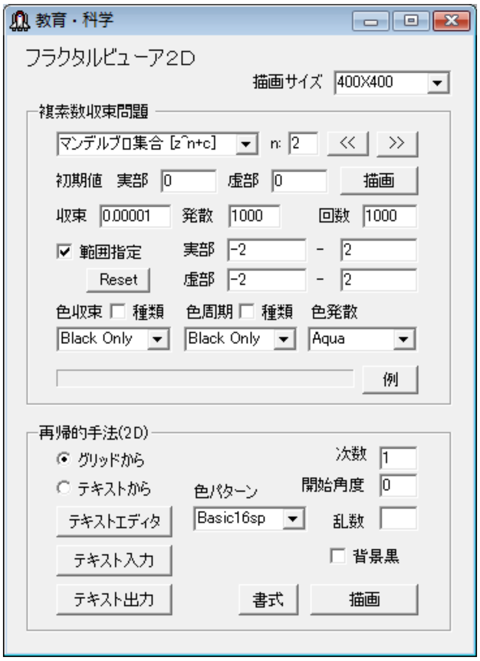


図 2.1 フラクタルビューア２D

２次元の再帰的手法には、線、円、正多角形（正三角形から正六角形）の描画要素によってフラクタルを描く機能がある。描画手順は Logo のタートルグラフィックスに似た独自のフラクタル記述言語によって記述される。その簡単な仕様は「書式」ボタンをクリックすることにより表示される。描画要素の色は、フラクタルの次数と色パターンで指定される。

2.1 複素数収束によるフラクタル

複素数の収束を利用したフラクタルでは、Mandelbrot 集合と充填 Julia 集合の他に我々のプログラムには以下の力学系が含まれている。

１）$z_n = z_{n-1}^k + c$　（Mandelbrot 集合，充填 Julia 集合）

２）$z_n = cz_{n-1}(1 - z_{n-1}^k)$

３）$z_n = cz_{n-1}(1 - 1/z_{n-1}^k)$

４）$z_n = z_{n-1}^k\,(c - z_{n-1})/(1 - \bar{c}\,z_{n-1})$

５）$\mathrm{z_n} = (|\mathrm{Re}(\mathrm{z_{n-1}})| - \mathrm{i}|\mathrm{Im}(\mathrm{z_{n-1}})|)^2 + \mathrm{c}$　(Burning Ship Fractal)

ここに、$k = 1,2,\cdots$，$z_i \in \boldsymbol{C}$，$(i = 1,2,\cdots)$，$c \in \boldsymbol{C}$である。また、１）～４）については、Mandelbrot 集合のように、z_0を固定してcの収束、周期振動、発散領域を調べる場合と、充填 Julia 集合のようにcを固定してz_0の収束、周期振動、発散領域を調べる場合が含まれている。

これらを図に描く際、美しさを強調するために、例えば$|z_n - z_{n-1}| < 0.00001, |z_n| > 1000$のように収束と発散の値を定め、これに到達した$n$の値を用いて色分けする。繰り返し回数$n$の最大値は、「回数」テキストボックスに記入する。これを超える場合は収束も発散もせず、周期振動する場合と判定するが、これにも繰り返しの最終的な到達点の近傍に何回目で近づくかを再計算することによって色を指定する。色の指定は、「色収束」、「色周期」、「色発散」コンボボックスを使って既定の色の組み合わせから選ぶ。繰り返し回数については、描画の正確さと美しさを優先させた値を取っているが、描画に時間がかかる場合は、回数を 1000 回から 100 回程度に減らしてよい場合もある。

この他に収束と周期振動については、もう 1 つ色付けの方法がある。収束については収束点が複数ある場合、どちらの収束点に収束するか、周期振動については周期がいくつかということによる色付けである。これは、「色収束」、「色周期」コンボボックスの上にある「種類」チェックボックスをチェックすることで選択される。

図 2.1 のメニュー画面では、まず集合の種類を選択する。力学系の数式の他に、上で説明した Mandelbrot 集合的な描画であれば「M 集合」、充填 Julia 集合的な描画であれば「J 集合」になっている。その他特殊なものもある。これらの集合の選択をすると次数nの値、描画範囲、色の種類など代表的な例の設定が行われるので、そのまま「描画」ボタンをクリックしてもサンプル画像が見られる。これ以外はユーザーの設定となる。

「描画」ボタンで表示された画面上の一部を拡大したい場合は、画面上でマウスをドラッグすると、描画範囲が赤い四角形で選択され、そのまま「描画」ボタンをクリックすると選択範囲が拡大表示され、その領域が描画画面の他に、メニュー画面のテキストボックスにも表示される。「範囲指定」チェックボックスにチェックがない場合、領域を選択していないときにはデフォルトの範囲になるが、「範囲指定」はチェックしておく方が分り易い。実部と虚部の範囲をデフォルトに直す時には「Reset」ボタンをクリックする。

何度も拡大するといくつか前の段階の画像を見たくなることがある。その際には、「<<」、「>>」ボタンで前後の画像を表示することができる。

例として一部分を拡大していった Mandelbrot 集合の図を図 2.2a、図 2.2b、図 2.2c、図 2.2d に示す。

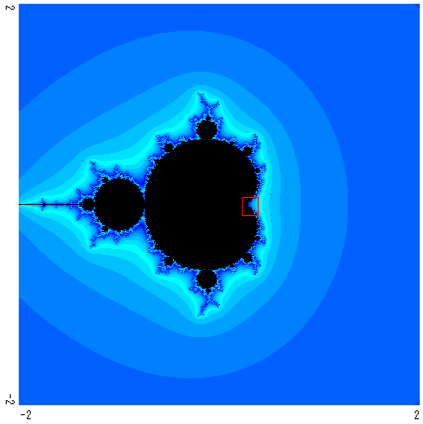


図 2.1.1a Mandelbrot 集合

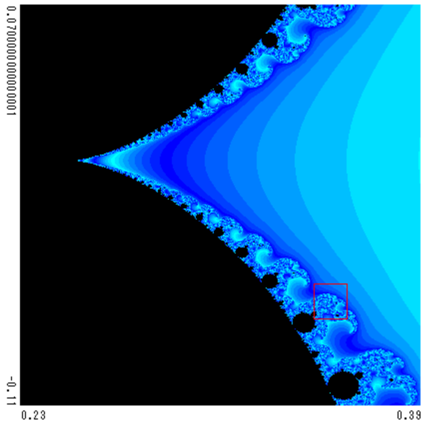


図 2.1.1b　拡大 1

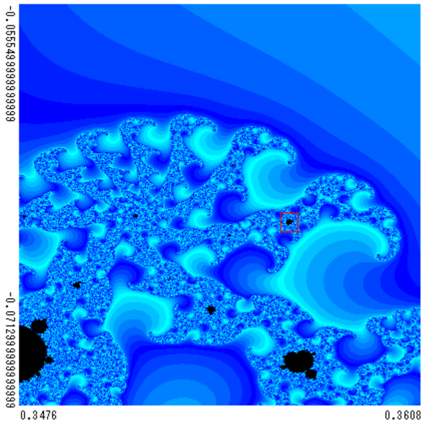


図 2.1.1c　拡大２

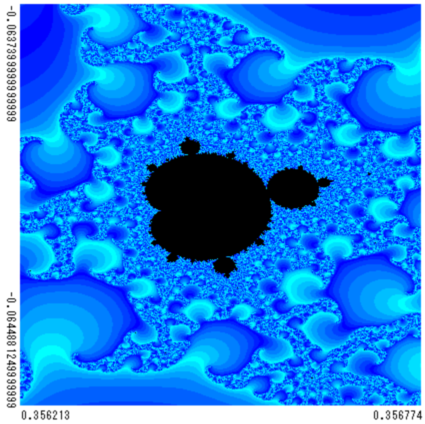


図 2.1.1d　拡大３

次に収束の種類による塗り分けの例を示す。図 2.1.2 は Mandelbrot 集合で、収束、1 周期、2 周期、・・・のように塗り分けた例である。中央左側（2 周期）や上下の丸い部分（3 周期）の色が変わっていることが分る。16 周期以上はすべて白色にするように設定している。

通常、充填 Julia 集合の図は、例えば$c = 0.3 + 0.3i$などのように、定数の値を図 2.1.2 の Mandelbrot 集合の中央部分の収束領域に与えて描くことが多いが、ここでは２周期振動となるその左側の領域内の値$c = -1$に取って図 2.1.3 を描いてみる。これは周期振動の種類を色分けする設定で実行したが、２周期点が 1 種類であることから、収束領域は単一色となっている。

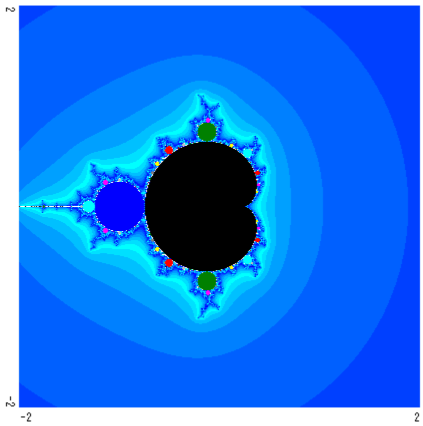


図 2.1.2 収束・周期で色分けした Mandelbrot 集合

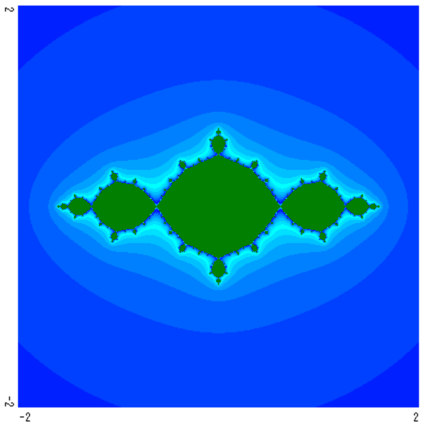


図 2.1.3 ２周期の定数を用いた充填 Julia 集合

図 2.1.4 は J 集合$z_n = c(z - 1/z)$, $c = 0.2 + 0.5i$ の場合の２周期振動領域の周期振動の種類（収束する周期振動点）によって塗り分けた例である。この場合画面全体が２周期振動領域で、収束点の種類が２種類（周期振動する２つの点が２種類）あることが分る。印刷の画像では見えないと思うが、図の中に無限遠の引力圏が点在する。抜き出して表示すると図 2.1.5 のようになる。これらは図 2.1.4 の複雑な模様の部分に点在していることが分る。この図もフラクタルのメニューに含まれている。

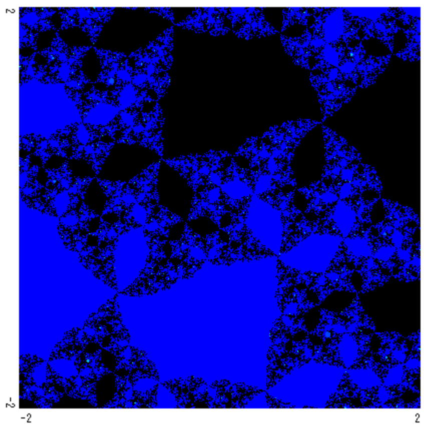


図 2.1.4 周期振動の収束点による色分け

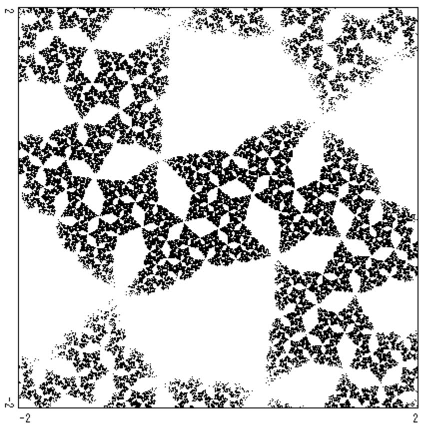


図 2.1.5 無限遠の引力圏

これらの集合以外で、我々のプログラムに含まれる集合とそのサンプルを以下に示す。図 2.1.6 は M 集合で、$z_n = cz_{n-1}(1 - z_{n-1}^3)$, $z_0 = 0.63$の発散領域とそれ以外の塗り分け、図 2.1.7 は M 集合で、$z_n = cz_{n-1}(1 - z_{n-1}^{-2})$, $z_0 = i$の周期による塗り分けである。

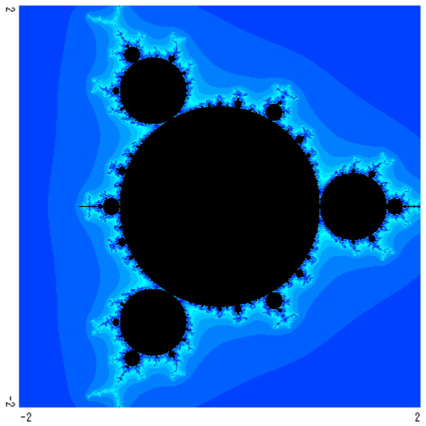


図 2.1.6 サンプル 1

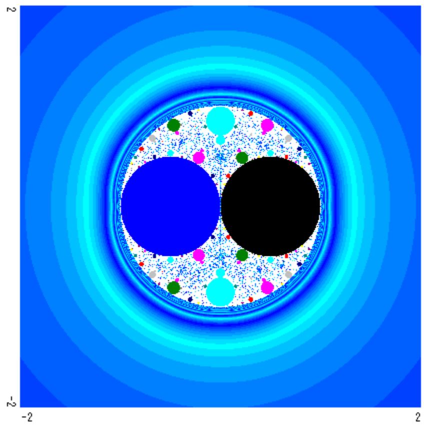


図 2.1.7 サンプル 2

図 2.1.8 は M 集合で、$z_n = z_{n-1}\,(z_{n-1} - c)/(1 - \bar{c}z_{n-1})$, $z_0 = 0.1i$の 3 領域による塗り分け、図 2.1.9 は充填 Julia 集合で、$z_n = z_{n-1}^2 + c$, $c = 0.3 + 0.3i$の発散と他の領域による塗り分けである。ここに$\bar{c}$はcの複素共役である。

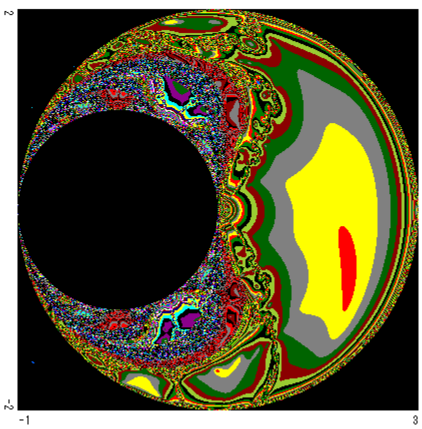


図 2.1.8 サンプル 3

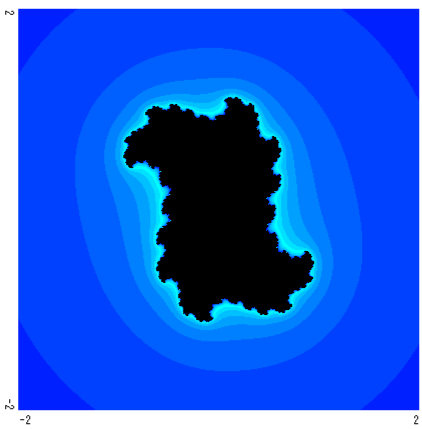


図 2.1.9 サンプル 4

図 2.1.10 は J 集合で、$z_n = cz_{n-1}(1 - z_{n-1}^3)$, $\mathrm{c} = 1.2 + 0.1\mathrm{i}$の発散と他の領域による塗り分けで、図 2.1.11 は J 集合で、$z_n = z_{n-1}^2\,(z_{n-1} - c)/(1 - \bar{c}z_{n-1})$, $z_0 = 1.1 + 2.1i$の 2 つの領域による塗り分けである。

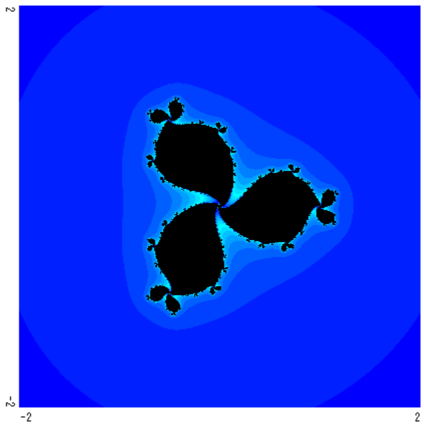


図 2.1.10 サンプル 5

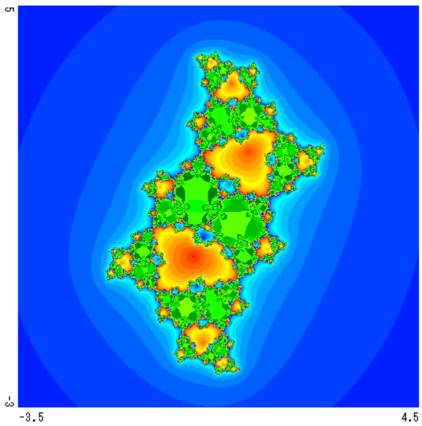


図 2.1.11 サンプル 6

2.2 再帰的方法によるフラクタル

再帰によるフラクタル画像の描画はプログラミングの基礎として多くのプログラマが経験する。しかし Windows 環境ではプログラム処理系のグラフィックの扱いに多少の基礎知識を必要とするので、多くの人が手軽に、というわけには行かない。そこで我々はこれらの知識に煩わされることなく、再帰処理だけを頭に入れてフラクタル画像が作れる簡単なマクロ言語を開発した。これを用いることで、何の準備もなく２Ｄのフラクタル画像を作成することができる。また、この言語はかなりの部分で３Ｄのフラクタルに応用できるので、次章の内容と合わせて読んでもらいたい。

まず簡単な例を示す。図 2.1 のメニューで「テキストから」ラジオボタンをチェックし、「テキストエディタ」ボタンをクリックする。テキストエディタが開かれるが、この中にプログラムを書き込み、完成したら「テキスト入力」ボタンをクリックして、プログラムをグリッドへコピーし保存するのがよい。グリッドからテキストへプログラムを書き出すのは、「テキスト出力」ボタンを利用する。

プログラムは基本的に LOGO のタートルグラフィックスのようになっており、左端 (0,0) から右端 (1,0) へ向けてスタートする。命令はコマンドとパラメータからなっているが、パラメータがない場合もある。パラメータには数式も使える。プログラムのサンプルを図 2.2.1 に示す。

```
turn 45
frac 1/2^0.5
turn -90
frac 1/2^0.5
```

図 2.2.1a サンプル 1

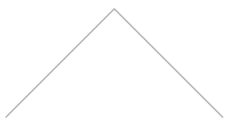


図 2.2.1b 次数 1

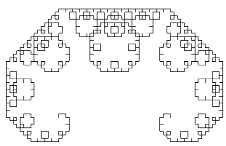


図 2.2.1c 次数 12

最初の turn 45 は反時計回りで角度 45°、frac 1/2^0.5 はスケール$1/\sqrt{2}$ の相似図形を現在の角度で貼り付けることを意味する。メニューに「開始角度」テキストボックスがあるが、これを 90° にすると、上向きに描画が始まり、図 2.2.1b と図 2.2.1c が反時計回りに 90° 回転した図形となる。

次の例は有名なコッホ曲線である。図 2.2.2 にプログラムと描画サンプルを示す。

```
frac 1/3
turn 60
frac 1/3
turn -120
frac 1/3
turn 60
frac 1/3
```

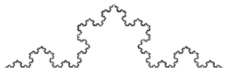
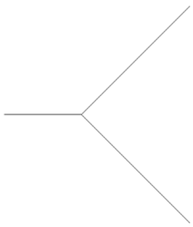


図 2.2.2a サンプル２　図 2.2.2b 次数 1　図 2.2.2c 次数 7

これは説明の必要がないであろう。同じような例が続いたので、次は分岐がある場合の例である。

```
go 0.3
separate
turn 45
frac 0.6
return
turn -45
frac 0.6
```



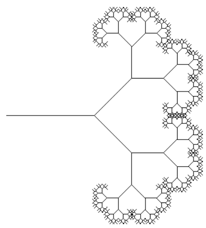


図 2.2.3a サンプル３　図 2.2.3b 次数 1　図 2.2.3c 次数 9

goto 0.3 は 0.3 のスケールで線を引く命令である。次の separate はこの時のタートルの位置と向きを記憶する命令である。分岐する場合によく使われるので separate にした。5 行目にある return は、前の separate で記憶した状態に戻す命令である。残念ながら現在は separate ～ return のネスト構造には対応していない。

次は、４角形を用いたサンプルである。

```
cfix
polygon4 1
turn 20
fracp 0.7
```

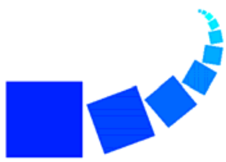


図 2.2.4a サンプル４　図 2.2.4b 次数 1　図 2.2.4c 次数 10

cfix は描画の際に図形の中心を基準にすることを意味する。これがなければ左下が基準である。次の polygon4 1 はスケール 1（1 辺の長さが 1）の四角形を描画する命令である。描画が終わった段階でタートルは図形の中心から、その時のタートルの向きに図形のスケールだけ進んでいる。最後の fracp 0.7 は通常の frac 0.7 とは異なる。frac の場合は次数 1 の場合にも 2 つの四角形が描かれてしまう。

fracp はこれを止めるためのフラクタルの予定地のようなものである。フラクタルは置くが、次の次数から表示する命令である。ちなみに色パターンは「Aqua」を利用している。

次の例は 3 角形で構成されるよく知られたシェルピンスキーのギャスケットである。

```
separate
fract3 0.5
fract3 0.5
return
turn 60
jump 0.5
turn -60
fract3 0.5
```

図 2.2.5a サンプル 5　　図 2.2.5b 次数 1　　図 2.2.5c 次数 6

ここでは fract3 0.5 があるが、これは 3 角形も書いてフラクタルも貼り付けるという意味である。このような書式を利用すると、次数 1 で 3 角形が 3 つになり、それに合わせて各次数で 3 角形が増える。また jump 0.5 は線を引かずにタートルを飛ばす命令である。

次数 1 で 3 角形を 1 つにしたい場合は、少し長くなるが、図 2.2.6 のようなプログラムにする。4 行目の separate はなくてもよい。

```
separate
polygon3 1
return
separate
fracp 0.5
fracp 0.5
return
turn 60
jump 0.5
turn -60
fracp 0.5
```

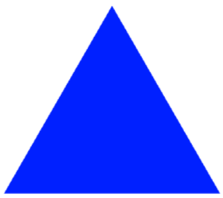
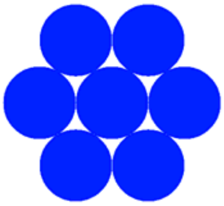


図 2.2.6a サンプル 6　　図 2.2.6b 次数 1　　図 2.2.6c 次数 6

図 2.2.7 は円を使ったサンプルである。cfract 1/3 は比率 1/3 の円を書いて、フラクタルを貼り付ける命令である。この場合比率は直径となる。cfix は図形の中心を基準にすることを意味する。プログラムは途中から省略している。

```
cfix
separate
cfract 1/3
return
jump 1/3
cfract 1/3
return
turn 60
jump 1/3
cfract 1/3
return
…
```



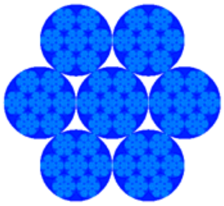


図 2.2.7a サンプル 7　　図 2.2.7b 次数 1　　図 2.2.7c 次数 4

図 2.2.8 は色に乱数を使った例である。

```
drep
cfix
separate
color int(16*rnd)
circle 1
return
color 14
pentagon 1
return
fracp 0.81
```

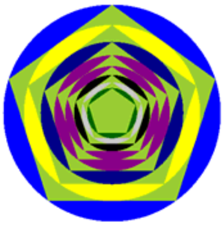


図 2.2.8a サンプル 8　　図 2.2.8b 次数 1　　図 2.2.8c 次数 8

最初の drep は正多角形の比率を外接円の直径にする宣言である。これがなければ比率は 1 辺の長さとなる。図形の色は色パターンと描画の次数によって自動的に決まるが、自分で指定することもできる。color 命令は直後の図形描画の色を指定する。ここでは円の色指定で、パラメータの中に乱数 rnd を使っている。

以上いくつかサンプルを示したが、マクロの簡単なまとめを表 2.1 に示す。

表 2.1 フラクタルビューア２Ｄの再帰処理書式

CFIX		正多角形の起点を中心に設定（お勧め）【宣言】
DREP		多角形の比率を外接円の直径表示に設定【宣言】
CONNECT	[色番号]	フラクタル同士を色番号の線でつなぐ　【宣言】
FRACP	比率	描画をしない再帰処理
FRACT3	比率	常に正三角形を描画する再帰処理
POLYGON3	比率	正三角形の描画
TRIANGLE	比率	正三角形の描画
FRACT4	比率	常に正方形を描画する再帰処理
POLYGON4	比率	正方形のを描画
SQUARE	比率	正方形のを描画
FRACT5	比率	常に正五角形を描画する再帰処理

POLYGON5	比率	正五角形の描画
PENTAGON	比率	正五角形の描画
FRACT6	比率	常に正六角形を描画する再帰処理
POLYGON6	比率	正六角形の描画
HEXAGON	比率	正六角形の描画
CFRACT	比率	常に円を描画する再帰処理
CIRCLE	比率	円の描画
FRAC	比率	最後の次数だけ直線描画の再帰処理
FRACT	比率	常に直線を描画描画する再帰処理
FRACC		強制的な終点へ連結する再帰処理
GO	比率	常に直線描画（標準/GOT）
GOF	比率	最後の次数だけの直線描画
GOC		強制的な終点へ連結する直線描画
TURN	角度	進行方向の角度変化[度]）
REVERSE		裏返し
SEPARATE		分岐の開始点・状態の保存（ネスト構造はまだ未対応）
RETURN		分岐への状態の戻り
RESET		進行方向・回転角の初期値再設定

注）多角形の場合、比率は１辺の長さで表す。
（DREP があるときは外接円の直径）
注）円の場合、比率は直径で表す。

３．フラクタルビューア３Ｄ

３次元空間へのフラクタルの描画は、反復関数による点の描画と再帰的手法による直線の描画で行われる。図 3.1 にフラクタルビューア３Ｄのメニュー画面を示す。

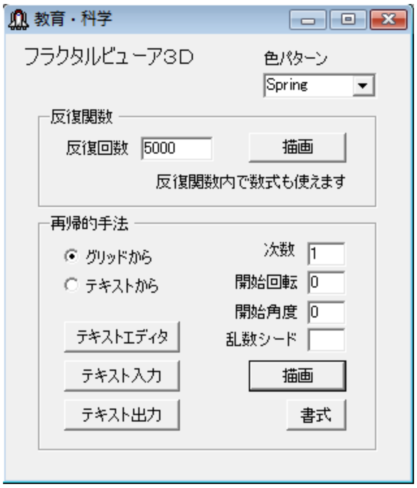


図 3.1 フラクタルビューア３Ｄメニュー

ここでは反復関数による方法と再帰的手法を順番に説明する。

3.1 反復関数による方法

２次元の場合r種類の反復関数は以下の行列計算で表される。

$$\begin{pmatrix} x_n \\ y_n \end{pmatrix} = \begin{pmatrix} a_{11}^{\alpha} & a_{12}^{\alpha} \\ a_{21}^{\alpha} & a_{22}^{\alpha} \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \end{pmatrix} + \begin{pmatrix} b_1^{\alpha} \\ b_2^{\alpha} \end{pmatrix} \qquad (\alpha = 1,2,\cdots,r)$$

これは複素平面上における以下の演算と同等であり、古くからその性質が調べられてきた。

$$z_n = a^{\alpha} z_{n-1} + b^{\alpha} \bar{z}_{n-1} + c^{\alpha} \qquad (\alpha = 1,2,\cdots,r)$$

ここに

$$z_n = x_n + i y_n, \quad z_{n-1} = x_{n-1} + i y_{n-1}, \quad \bar{z}_{n-1} = x_{n-1} + i y_{n-1},$$

$$a^{\alpha} = (a_{11}^{\alpha} + a_{22}^{\alpha})/2 + i\,(a_{21}^{\alpha} - a_{12}^{\alpha})/2, \quad b^{\alpha} = (a_{11}^{\alpha} - a_{22}^{\alpha})/2 + i\,(a_{12}^{\alpha} + a_{21}^{\alpha})/2,$$

$$c^{\alpha} = b_1^{\alpha} + i b_2^{\alpha}.$$

これを３次元に拡張すると、反復関数は以下のように表される。

$$\begin{pmatrix} x_n \\ y_n \\ z_n \end{pmatrix} = \begin{pmatrix} a_{11}^{\alpha} & a_{12}^{\alpha} & a_{13}^{\alpha} \\ a_{21}^{\alpha} & a_{22}^{\alpha} & a_{23}^{\alpha} \\ a_{31}^{\alpha} & a_{32}^{\alpha} & a_{33}^{\alpha} \end{pmatrix} \begin{pmatrix} x_{n-1} \\ y_{n-1} \\ z_{n-1} \end{pmatrix} + \begin{pmatrix} b_1^{\alpha} \\ b_2^{\alpha} \\ b_3^{\alpha} \end{pmatrix} \quad (\alpha = 1,2,\cdots,r)$$

このr種類の反復関数から、確率的に 1 つ選んで計算を実行し、それを繰り返して点を打って行く。

以下具体的に例を示しながら結果を見て行こう。図 3.1.1 に非常に有名な C 曲線のデータと実行例を示す。実行例は 10000 点を打ったものである。描画結果は紙面（スクリーン面）上方向が z 軸正の方向、紙面右方向が x 軸正の方向である。

	確率	a1	a2	a3	a0
▶ 初期値		1	0	1	
反復関数1	0.5	0.5	0	-0.5	0
		0	0	0	0
		0.5	0	0.5	0
反復関数2	0.5	0.5	0	0.5	0.5
		0	0	0	0
		-0.5	0	0.5	0.5

図 3.1.1a C 曲線データ

図 3.1.1b 実行結果

データでは、1 行目に初期値として z_0 の値、1 列目に反復関数を選択する確率、2 行 2 列目以降が反復関数の係数行列の値である。同様にして、コッホ曲線の例を図 3.1.2 に示す。

	確率	a1	a2	a3	a0
▶ 初期値		1	0	0	
反復関数1	0.5	1/2	0	3^0.5/6	0
		0	0	0	0
		3^0.5/6	0	-1/2	0
反復関数2	0.5	1/2	0	-3^0.5/6	1/2
		0	0	0	0
		-3^0.5/6	0	-1/2	3^0.5/6

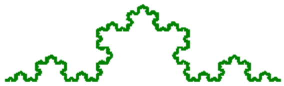


図 3.1.2a コッホ曲線データ　　　　図 3.1.2b 実行結果

図 3.1.2 ではデータに数式を用いている。以上 2 つは 2 次元の例であったが、図 3.1.3 に 3 次元の例を示す。

	確率	a1	a2	a3	a0
▶ 初期値		1	1	1	
反復関数1	0.25	0.5	0	0	0
		0	0.5	0	0
		0	0	0.5	0
反復関数2	0.25	0.5	0	0	60
		0	0.5	0	0
		0	0	0.5	0
反復関数3	0.25	0.5	0	0	30
		0	0.5	0	60
		0	0	0.5	0
反復関数4	0.25	0.5	0	0	30
		0	0.5	0	30
		0	0	0.5	60

図 3.1.3a 3 次元ギャスケットデータ　　　　図 3.1.3b 実行結果

これらの他にも参考文献 3）などで紹介されている例を図 3.1.4、図 3.1.5 にあげておこう。

	確率	a1	a2	a3	a0
▶ 初期値		0	1	1	
反復関数1	0.1	0.05	0	0	0
		0	1	0	0
		0	0	0.6	0
反復関数2	0.2	0.05	0	0	0
		0	1	0	0
		0	0	-0.5	1
反復関数3	0.2	0.46	0	-0.32	0
		0	1	0	0
		0.39	0	0.5	0.6
反復関数4	0.2	0.47	0	-0.15	0
		0	1	0	0
		0.17	0	0.42	1.1
反復関数5	0.2	0.43	0	0.28	0
		0	1	0	0
		-0.25	0	0.45	1
反復関数6	0.1	0.42	0	0.26	0
		0	1	0	0
		-0.35	0	0.31	0.7

図 3.1.4a 木のようなフラクタル　　　　図 3.1.4b 実行結果

	確率	a1	a2	a3	a0
▶ 初期値		0.5	0	0.5	
反復関数1	0.73	0.856	0	0.0414	0.07
		0	1	0	0
		-0.0205	0	0.858	0.147
反復関数2	0.13	0.244	0	-0.385	0.393
		0	1	0	0
		0.176	0	0.224	0.102
反復関数3	0.13	-0.144	0	0.39	0.527
		0	1	0	0
		0.181	0	0.259	-0.014
反復関数4	0.01	0	0	0	0.486
		0	1	0	0
		0.355	0	0.216	0.05

図 3.1.5a 葉のようなフラクタル

図 3.1.5b 実行結果

3.2 再帰的方法

再帰的方法は 2 章でも述べたが、2 次元でタートルは平面上を移動するので左右に向きを変えるとき turn 命令を用いるが、3 次元ではタートルの背中方向（初期値は y 軸負の方向）に法線ベクトルを考え、この向きを rotate 命令で変更する。ここで反復関数のときと同様に紙面（スクリーン面）上方向が z 軸正の方向、紙面右方向が x 軸正の方向である。しかし考えにくい場合は、タートルを 90° ひねり、背中を z 軸方向に向け、法線ベクトルの方向をタートルの右脇方向と考え、利用者はタートルに乗った状態をイメージすればよい。これで、rotate はタートルの回転（右ねじ方向）、turn は上下方向への向き変えとなる。以後この考え方に基づいて解説する。

この方法を用いた 3 次元フラクタルの例を図 3.2.1 に示す。

```
go 0.4
separate
turn -30
frac 0.6
return
rotate 60
turn 30
frac 0.6
return
rotate -60
turn 30
frac 0.6
return
```

図 3.2.1a サンプル 1

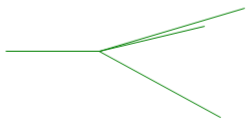


図 3.2.1b 次数 1

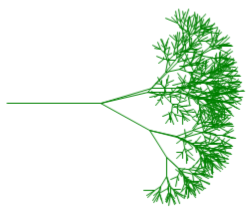


図 3.2.1c 次数 6

初期状態で進行方向は右横（x 軸方向 0 から 1 へ）で、タートルの右脇に当たる法線ベクトルは、y 軸負の方向である。紙面は x-z 平面で、タートルを右横から見ている状態になる。go 0.4 は 0.4 だけ進んで線を描く命令である。separate はその状態を記憶する。turn -30 で 30° 下を向き、frac 0.6 で、0.6 倍に縮小したフラクタルを貼り付ける。return で記憶した位置に戻り、rotate 60 で 60° 右

ねじの方向に回転、turn 30 で 30° 上を向いて、0.6 倍に縮小したフラクタルを貼り付ける、等々である。

次に図 3.2.2 にフラクタルを貼り付ける際に直線を表示しない fracp を用いた例を示す。

```
separate
rotate 90
go 1
turn 90
go 1
turn 90
go 1
turn 90
go 1
return
jump 0.5
turn -90
jump 0.25
turn 90
rotate 90
turn 45
rotate -90
fracp 0.7071
```

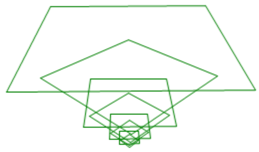


図 3.2.2a サンプル 1　図 3.2.2b 次数 1　図 3.2.2c 次数 8

今回のサンプルは真横からだと見にくいので、少し傾けて表示してある。最初に rotate 90 することによって、描画の方向が x-y 平面上になり、正方形を描いている。始点まで戻ったら、最初の辺の中央にジャンプし、90° 傾けて 0.25 ジャンプする。元に戻して 45° 傾けて 0.7071 倍のフラクタルを表示せずに貼り付ける。

最後にマクロは付けないが、いくつかの例を示しておく。次数はいずれも 5 である。図 3.2.3 は図 3.1.3 で描いたギャスケットを再帰的手法で描いたものである。描画の要素は直線になっている。図 3.2.4 はフラクタル同士をつなぐ connect 命令を利用したヒルベルト曲線である。図 3.2.5 は同じフラクタルを複数同時に表示するために作られた copy 命令を使ったチリのような図形である。

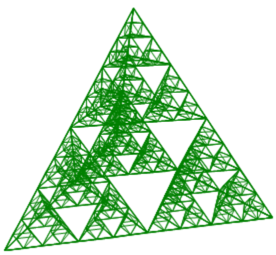


図 3.2.3 ギャスケット

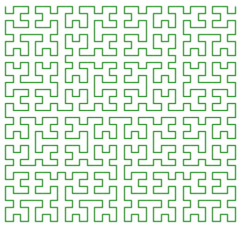


図 3.2.4 ヒルベルト曲線

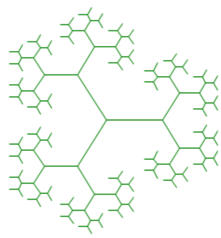


図 3.2.5 チリ

これらのマクロはホームページ上の Sample.zip 中のフラクタル 3D（再帰）.txt に含まれている。

以上いくつかサンプルを示したが、マクロの簡単なまとめを表 3.1 に示す。

表 3.1 フラクタルビューア３Ｄの再帰処理書式

CONNECT		フラクタル同士を線でつなぐ　【宣言】　例：ヒルベルト曲線
FRAC	比率	最後の次数だけ直線描画の再帰処理（標準/FRACF）
FRACP	比率	直線描画をしない再帰処理
FRACT	比率	常に直線描画する再帰処理
GO	比率	常に直線描画（標準/GOT）
GOF	比率	最後の次数だけの直線描画
TURN	角度	進行方向の角度変化[度])
ROTATE	角度	進行方向の回転[度](/ROUND)
FRACC		強制的な終点へ連結する再帰処理
GOC		強制的な終点へ連結する直線描画
SEPARATE		分岐の開始点・状態の保存（ネスト構造はまだ未対応）
RETURN		分岐への状態の戻り
RESET		進行方向・回転角の初期値再設定
START		進行方向・回転角・位置の初期値再設定
COPY	比率	同じフラクタルの描画（2 つ目以降）

４．考察

我々のフラクタルビューアでは、２次元は通常のビットマップ画像として、３次元は３Ｄビューアを用いた空間データとして出力した。当然２次元も平面データとして出力することができるが、画面を動かしてもあまり効果的でないこと、ドットの数を増やしてより細かい絵を描きたかったことからビットマップへの出力を選択した。３次元では動きを重視するために、描画要素数 10000 程度までが望ましい。

我々は、２次元のフラクタル描画手法として、複素数の収束による方法と再帰的手法、３次元の描画手法として反復関数による方法と再帰的手法を提供してきたが、反復関数による方法は２次元でも利用可能であるので、これを加えることも考えたい。ただその際には、３次元ではあまり考えていなかった色を重視した追加にすべきであろう。またその他にもフラクタルの表示方法はいくつかあり、これらについても検討したい。我々のプログラムの目的はあくまでフラクタル描画原理の学習支援であるので、機能強化を重ねプログラムを複雑化することには疑問がある。適度なバランスが重要であると考える。

フラクタルビューア２Ｄのマクロに比べ、フラクタルビューア３Ｄのマクロは単純である。描画も直線を集めたもので、面の概念もない。そのため四角形１つにも 10 行近くの命令が必要である。我々は、単純な多角形を描画要素に加えたいと考えるが、面にはその法線方向のデータが必要となり、拡張の仕様が固まっていない。これは今後の課題である。

我々のプログラムはフラクタルアートを作成するようなものではなく、基本的に原理を学ぶための

ツールである。それゆえ作る画像は、完全な自己相似形であり、それらを組み合わせて表示することや視覚的な効果を加えることは殆ど考えていない。ただ、３Ｄのフラクタルについては「Copy 比率」命令を加えて、大きさを変えた同じフラクタルを組み合わせることができるようにしている。

参考文献

１）社会システム分析のための統合化プログラム 16 －３Ｄビューアとその応用－, 福井正康, 尾崎誠, 石丸敬二, 福山平成大学経営学部経営研究, 7 号, (2011) 掲載予定.

２）College Analysis における「２変量関数表示ユーティリティ」の構造と機能, 石丸敬二, 福井正康, 福山大学経済学論集, 第 35 巻, 2 号, (2010) 87-106.

３）フラクタルで描く魅惑的な画像の世界, 白井豊, ゆたか創造舎, 2009.

４）カオスとフラクタル－Excel で体験, 臼田昭司他, オーム社, 1999.

５）カオスとフラクタル―非線形の不思議, 山口昌哉, 講談社, 1986.

６）複素数とフラクタル, 芹沢浩, 東京図書, 1995.

Multi-purpose Program for Social System Analysis 17

- Fractal Viewer -

Masayasu FUKUI, Makoto OZAKI and Keiji ISHIMARU*

Department of Business Administration, Faculty of Business Administration,
Fukuyama Heisei University
* Department of Economics, Faculty of Economics,
Fukuyama University

Abstract

We have been constructing a unified program on the social system analysis for the purpose of education. We are interested in graphics and made some programs which use 3D viewer. We will introduce fractal viewers in these programs. These programs are tools which draw 2D or 3D fractal graphics including famous Mandelbrot or filled Julia set.

Keywords

College Analysis, social system analysis, statistics, management science, decision making, fractal, 3D

URL: http://www.heisei-u.ac.jp/ba/fukui/