SASによるフラクタル表現

山下敏治*　奥田達也**
* ファイザー製薬(株)バイオメトリクス部解析グループ
** 住商情報システム(株)システム・マネジメント事業部

1. はじめに

フラクタルは自然界の複雑な地形や構造(海岸線、稜線など)の観察・研究から考え出されたフランス人のベノワ・B・マンデルブロー(1975)によって提唱された概念で、一般的に「部分を拡大しても同じ形→自己相似集合」と理解されています。

理論式(IFSや複素関数族、離散モデルなど)を扱ったコンピュータがつくりだすフラクタルについてはすでにいくつかのソフトウエアやWebサイトが公開されていますが、今回、SASによるフラクタル表現を試みましたのでその事例を紹介いたします。

2. 一般的なフラクタル

コッホ曲線、葉脈曲線、シェルピンスキーのギャスケット、レヴィのC曲線、楓の葉など一般的なフラクタルは、いずれも複素数 $z=x+iy$ として基本的に下記に示す写像Fを、無限回を考えて等確率的に繰り返し反復計算し（反復関数系IFS：Iterated Function System）、その合併集合Kをプロットすることでグラフ化できます。

■ 葉脈曲線、レヴィのC曲線（図1、3）

$$\begin{pmatrix} F_1(z) \\ F_2(z) \end{pmatrix} = \begin{pmatrix} \alpha_1 & \beta_1 \\ \alpha_2 & \beta_2 \end{pmatrix} \begin{pmatrix} z \\ \bar{z} \end{pmatrix} + \begin{pmatrix} 0 \\ 1-\alpha_2-\beta_2 \end{pmatrix}, \quad K = F_1(K) \,\mathrm{Y}\, F_2(K), \quad (|\alpha_i|+|\beta_i|<1, i=1,2)$$

葉脈曲線は$\alpha_1=\alpha_2=0, \beta_1=\frac{1}{2}+i\frac{\sqrt{3}}{6}, \beta_2=1-|\beta_1|^2$ Λ *Fig*.1

レヴィのC曲線は$\alpha_1=\alpha_2=0, \beta_1=\frac{1}{2}+i\frac{1}{2}, \beta_2=\overline{\beta_1}$ Λ *Fig*.3

■ シェルピンスキーのギャスケット（図2）

縮小写像F_3が加わりますが基本原理は同じです。（Appendix 1 参照）

■ 楓の葉（図4）

さらに縮小写像F_4が加わります。写像F_1～F_4の発生確率はそれぞれ1/2, 1/6, 1/6, 1/6 になります。（Appendix 1 参照）

SASでグラフを描くには（図3：レヴィのC曲線）

```
data levy;
 a=1/2; b=1/2;
 x=1  ; y=1  ;

 do i=1 to 2**15;
    j=ranuni(i);
    f=1*(0<=j<1/2)+2*(1/2<=j<=1);
    if f=1 then do;
       u=a*x-b*y;
       v=b*x+a*y;
       x=u;
       y=v;
       output;
       end;
    if f=2 then do;
       u= a*x+b*y+a;
       v=-b*x+a*y+b;
       x=u;
       y=v;
       output;
       end;
    end;

symbol1 v=dot i=none h=0.05 color=cx00ffff;
symbol2 v=dot i=none h=0.05 color=cx00ff00;

proc gplot data=levy;
 plot y * x = f / nolegend;
run;
```

初期値の設定

確率1/2でそれぞれ写像 F_1, F_2 にランダムに割り当てます。

ここではこの反復操作（IFS）を約3万回行っています。

GPLOTプロシジャで複素平面上にプロット。写像毎にカラーを割り振ります。完成！

Fig.1 Stem Leaf Curve

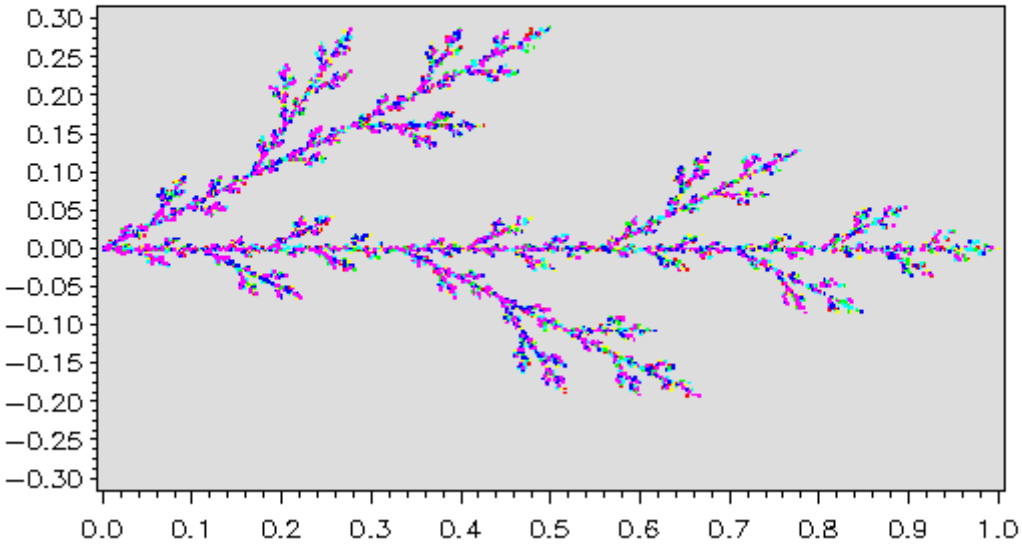


Fig.2 Sierpinski Gasket

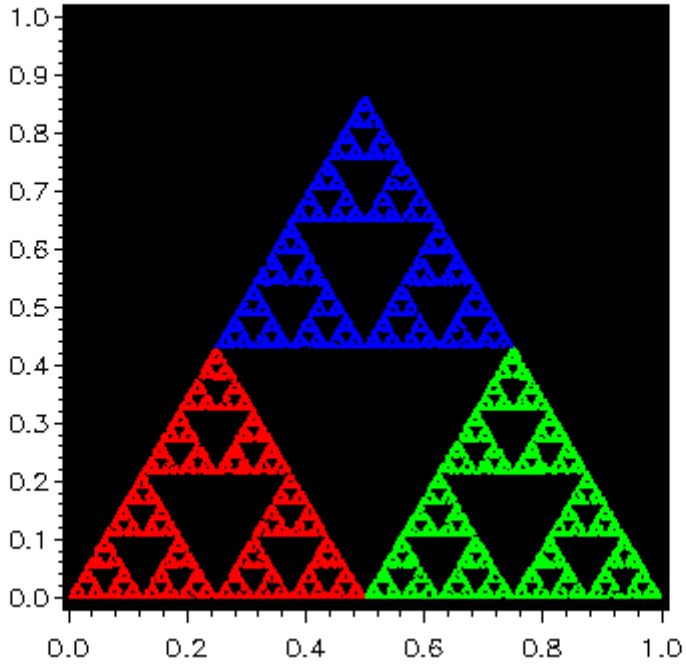


Fig.3 Levy's C

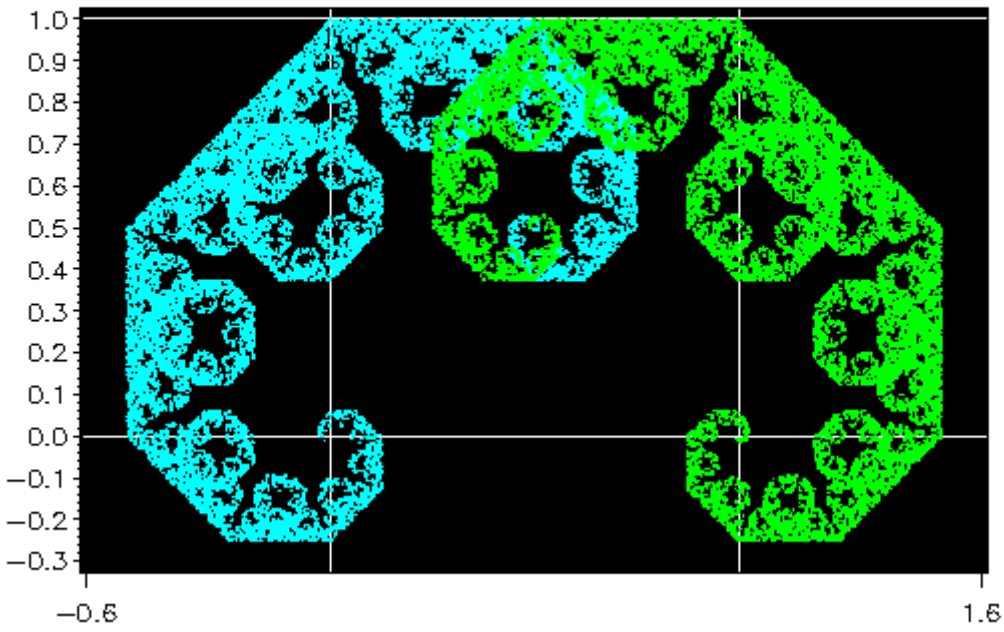
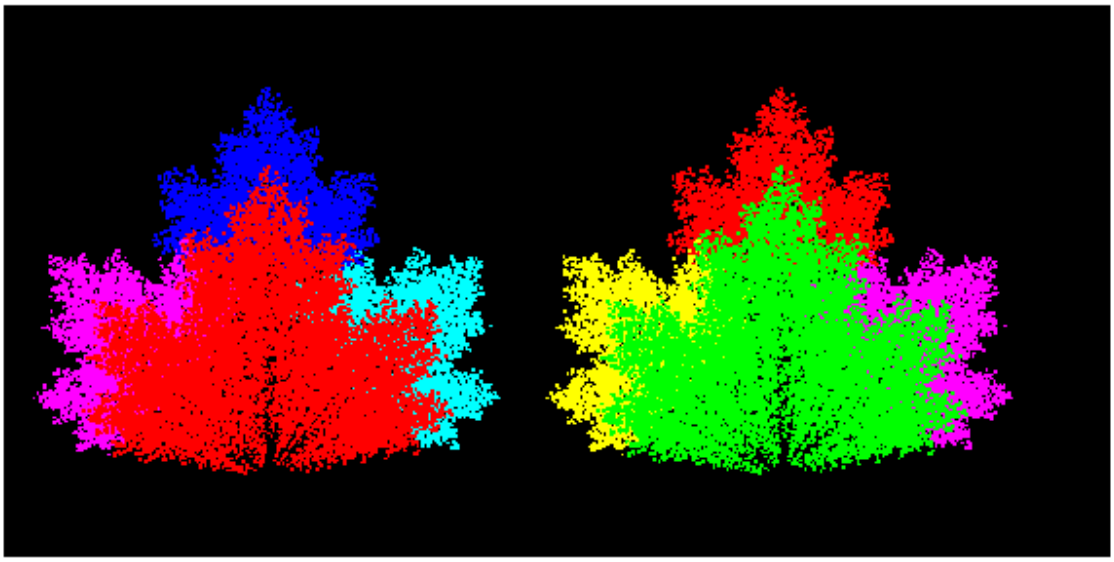


Fig.4 Maple Leaf



3. 複素関数族のフラクタル

マンデルブロー集合 M、ジュリア集合（充填ジュリア集合）K_γは、$z=x+iy$ として下記の複素関数式を反復計算することで与えられ、収束条件を判定して得られる極限集合を複素平面上にプロットすることでグラフ化できます。

■マンデルブロー集合（図5）

$$z = x + iy$$

$$f(z) = z^2 + \gamma, \quad \gamma = a + ib, \quad M = \left\{ \gamma \mid f(z), z_0 = 0, \lim_{n\to\infty} |z_n| < \infty \right\} \ \Lambda \ Fig.5$$

ここで

$$f(z) = (x+iy)^2 + (a+ib) = (x^2 - y^2 + a) + i(2xy + b)$$

■ジュリア集合（図6～8）

$$z = x + iy$$

$$f(z) = z^2 + \gamma, \quad \gamma = a + ib, \quad K_\gamma = \left\{ z_0 \mid f(z), \lim_{n\to\infty} |z_n| < \infty \right\} \ \Lambda \ Fig.6 \ \& \ 7$$

$$f(z) = \gamma z(1-z), \quad \gamma = a + ib, \quad K_\gamma = \left\{ z_0 \mid f(z), \lim_{n\to\infty} |z_n| < \infty \right\} \ \Lambda \ Fig.8$$

ここで

$$f(z) = \gamma z(1-z) = (a+ib)(x+iy)(1-x-iy)$$
$$= (ax-by)(1-x) + (bx+ay)y + i\{(bx+ay)(1-x) - (ax-by)y\}$$

SASでグラフを描くには（図6：ジュリア集合）

```
data julia;

 a=0.285; b=0.012;

 do x0=-1.2 to 1.2 by 0.01;
 do y0=-1.2 to 1.2 by 0.01;
    x=x0; y=y0;
    do i=1 to 2**5 until(r>2);
       u=x**2-y**2+a;
       v=2*x*y+b;
       x=u;
       y=v;
       r=sqrt(u**2+v**2);
       end;
    output;
    end;
    end;
run;
```

パラメータ a, b の設定

$Re(z_0)$と $Im(z_0)$の定義域、増分（分割）を設定します。

複素数 z_0 の反復計算を行い、$|z_0|$で収束・発散の判定を行います。

定義域にある複素数 z_0 全てにこの操作を行います。

```
symbol1  v=dot i=none h=0.01 color=h000 lss;/* deg   0 */
symbol2  v=dot i=none h=0.01 color=h00b lss;/* deg  11 */
symbol3  v=dot i=none h=0.01 color=h016 lss;/* deg  22 */
(continue)
symbol32 v=dot i=none h=0.01 color=h15d lss;/* deg 349 */

proc gplot data=julia;
 plot y0 * x0 = i / nolegend;
run;
```

GPLOTプロシジャを用いて z_0 を複素平面上にプロット。収束の反復回数毎にカラーを割り振っています。部分を拡大しても同じ形をしており「自己相似集合」と呼ばれています。

Fig.5 Mandelbrot Set

f(z)=z*z+r, z0=0, r=a+ib

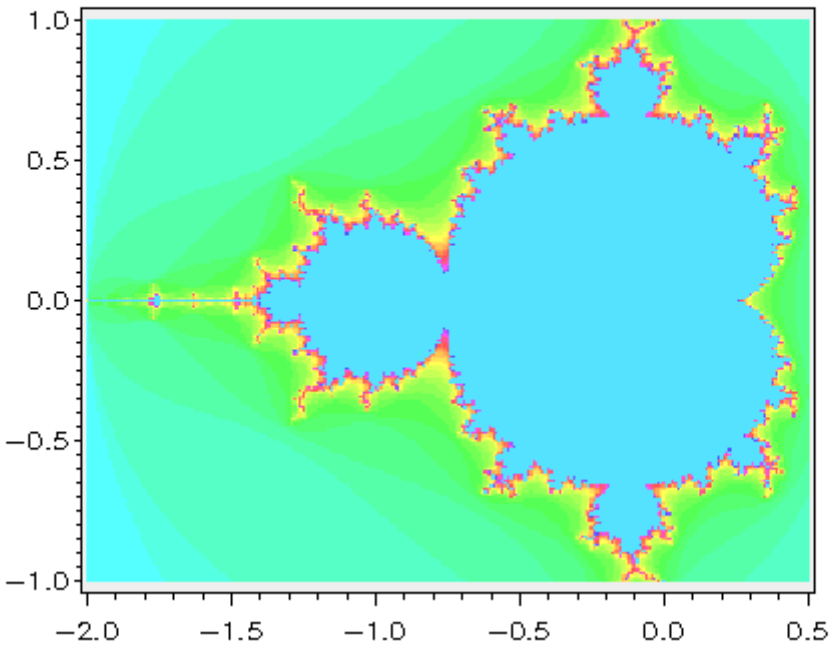


Fig.6 Julia Set

f(z)=z*z+r, r=0.285+0.012i

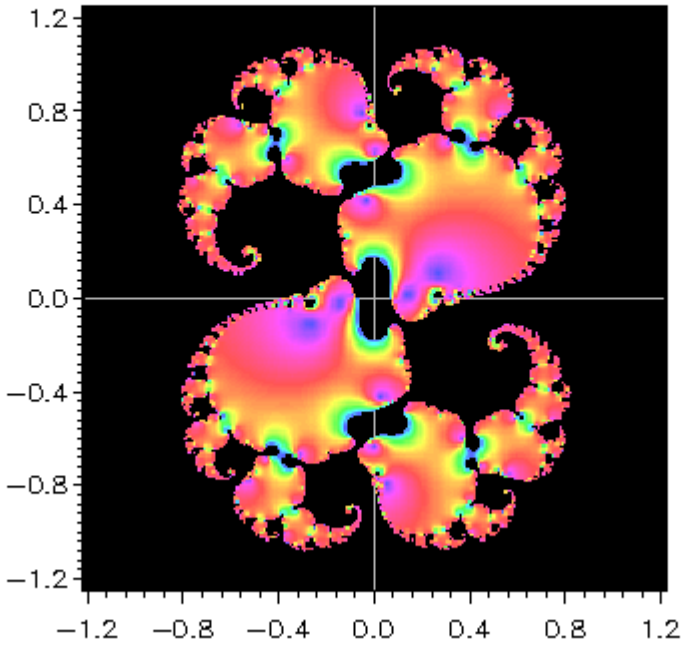


Fig.7 Julia Set

f(z)=z*z+r, r=−0.75+0.00i

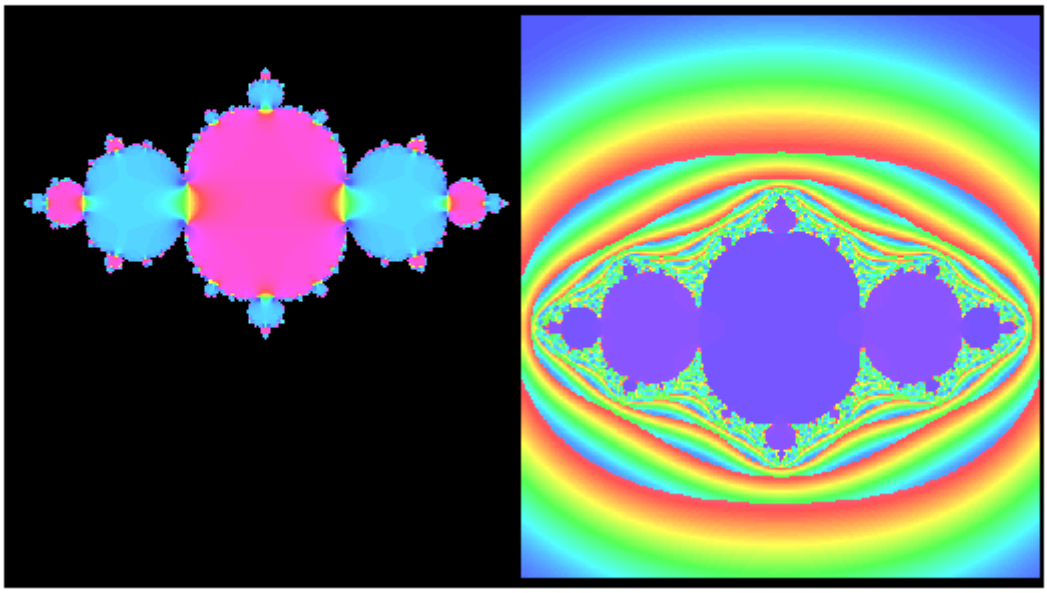
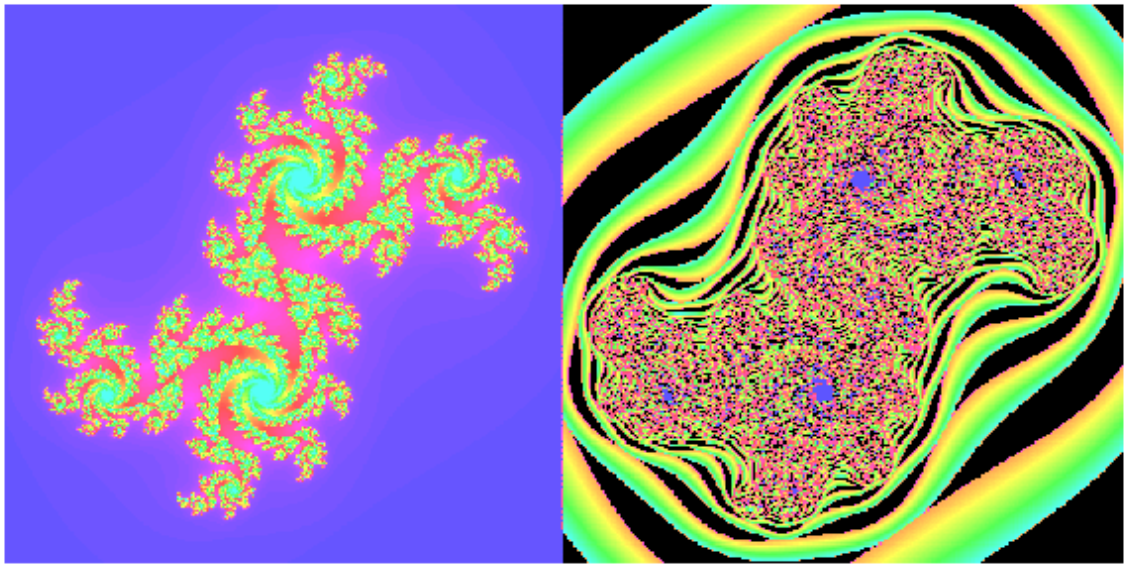


Fig.8 Julia Set

f(z)=r*z*(1−z), r=1.64+0.96i



4. 離散方程式のダイナミックス

ローレンツ方程式などのストレンジアトラクタ、審美的カオスと呼ばれるエノン写像やグモフスキーとミラの写像はそれぞれ次式で与えられ、繰り返し計算された軌道を3次元あるいは2次元の実数空間・平面上にプロットしていくことでビジュアル化することができます。

■ローレンツ・アトラクタ（図9～11）

$dx/dt = -cx + cy$

$dy/dt = -xz + rx - y$

$dz/dt = xy - bz$

ここでx, y, zは時間tの関数、σ, r, bは物理的常数

■エノン写像（図12）

$$x_{n+1} = ax_n - b(y_n - x_n^2)$$

$$y_{n+1} = bx_n + a(y_n + x_n^2)$$

ここでa, bは$e^{i\theta} = \cos\theta + i\sin\theta$の第1成分（実部）Reと第2成分（虚部）Im

■グモフスキーとミラの写像（図13～15）

第1式

$$x_{n+1} = by_n + F(x_n)$$

$$y_{n+1} = -x_n + F(x_{n+1})$$

$$F(x_n) = ax_n + \frac{2(1-a)x_n^2}{1+x_n^2}$$

第2式

$$x_{n+1} = y_n + a(1-by_n^2)y_n + G(x_n)$$

$$y_{n+1} = -x_n + G(x_{n+1})$$

$$G(x_n) = \mu x_n + \frac{2(1-\mu)x_n^2}{1+x_n^2}$$

SASでグラフを描くには（例：ローレンツアトラクタ）

```
data lorenz;

 s =10;
 r =28;
 b =8/3;
 dt=0.005;

 x0=10; y0=12; z0=15;
 x =x0; y =y0; z =z0;
 c='h000aaff';
 output;

 do i=1 to 2**11;
    _x=x+dt*(-s*x+s*y);
    _y=y+dt*(-x*z+r*x-y);
    _z=z+dt*(x*y-b*z);
    x=_x;
    y=_y;
    z=_z;
    c='h' || put(mod((360/60)*(i-1),360),hex3.) || '88ff';
    output;
    end;
run;

proc g3d data=lorenz;
 scatter y * x = z
       / shape='balloon' size=0.1 noneedle color=c;
run;
```

パラメータ σ,r,b および時間 t の微小幅 *dt* を設定。

時間 t の関数 x,y,z の初期値を設定します。

微小時間 *dt* の微小幅 *dx,dy,dz* を反復計算します。ここでは $dt*2^{11}$=約 10 秒を設定しています。

時間 t の変化に合わせてカラーを HLS コードで設定。

G3Dプロシジャを用いてアトラクタを3次元空間にプロット。図9～11ではさらに annotate 機能を用いて X-Y 平面、X-Z 平面への投射を行っています。けっして交わることなく織りたたまれていく軌道が確認できます。

Fig.9 Lorenz Attractor
s=10, r=8.0, b=2.67

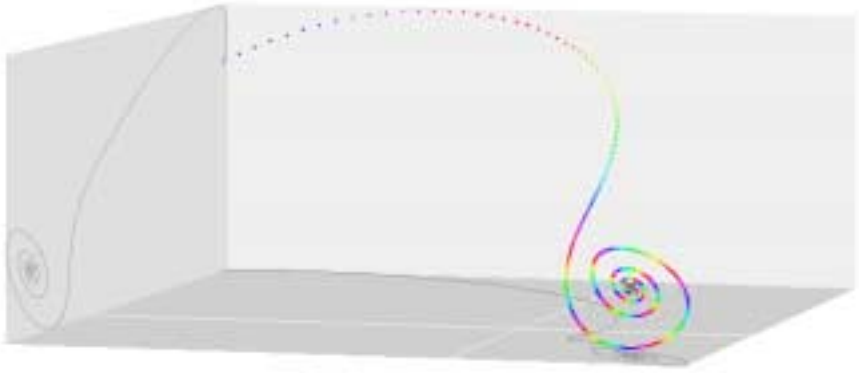


Fig.10 Lorenz Attractor
s=10, r=15.0, b=2.67

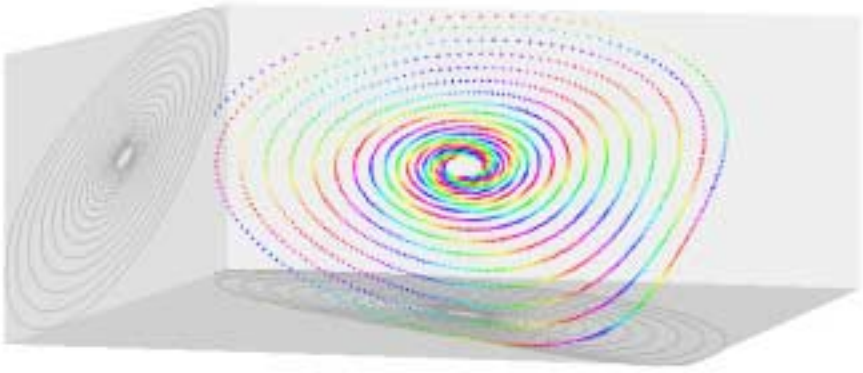


Fig.11 Lorenz Attractor
s=10, r=28.0, b=2.67

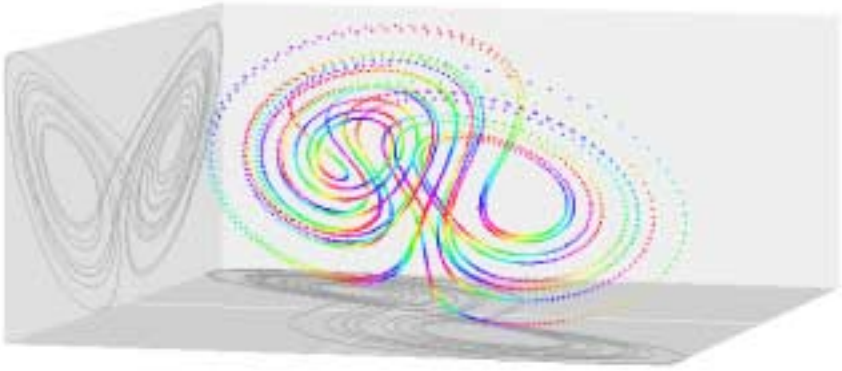


Fig.12 Henon's Orbits

a=0.24, b=0.9708

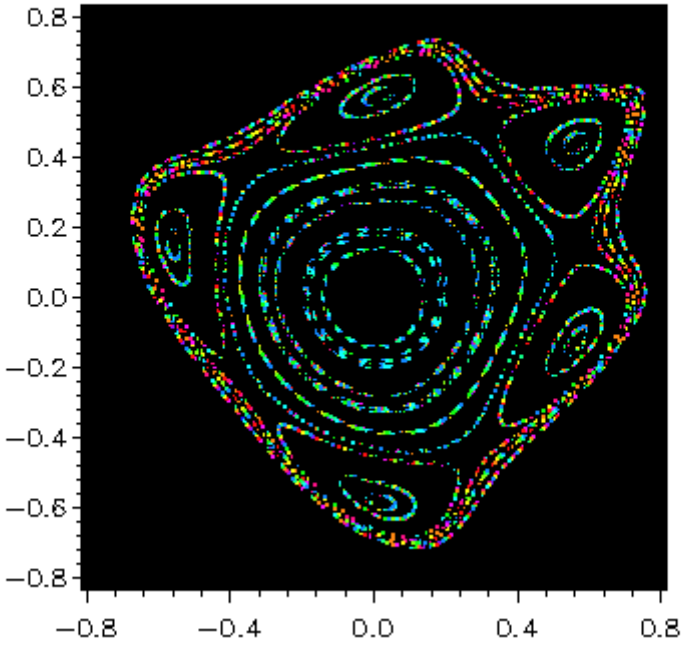


Fig.13 Mira's Map

a=0.7, b=0.9998

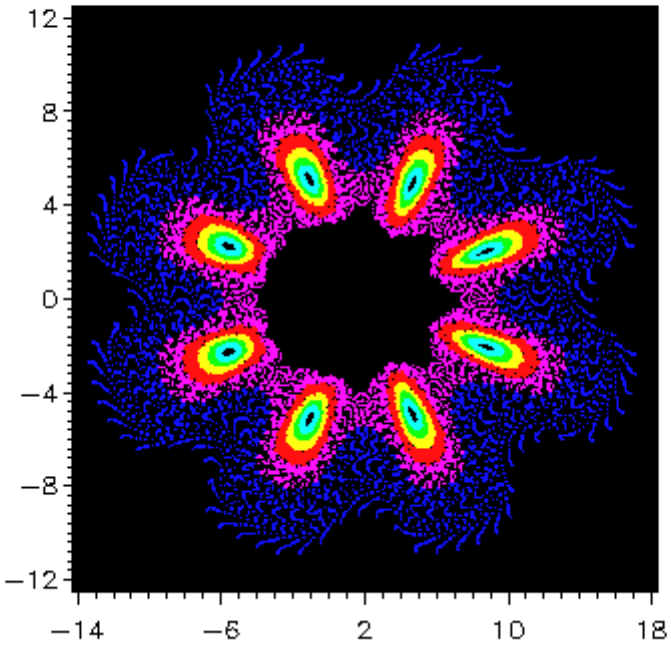


Fig.14 Mira's Map

a=0.31, b=0.9998

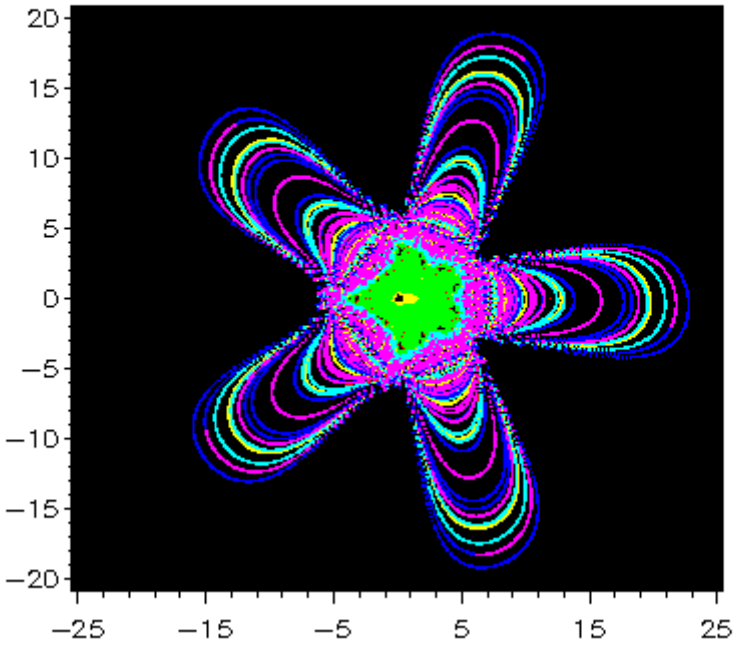
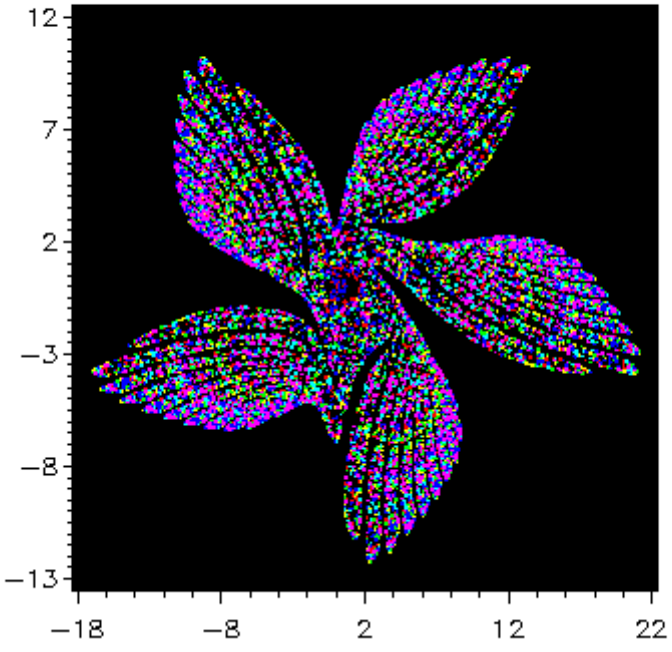


Fig.15 Mira's Map

u=−0.8, a=0.008, b=0.05



5. おわりに

SASによるフラクタル表現の事例紹介を行いました。

まとめとして、

1）SASのコーディングは比較的かんたん

2）SAS/GRAPHのプロシジャ群（GPLOT、G3D、G CONTOUR）はフラクタルの表現にたいへん有効

3）カラーコード（RGB、HLS）は多彩な表現に有効

4）G3Dプロシジャをもっとパワーアップしてほしい

5）複素関数の計算機能がほしい…ときがある

フラクタルの概念そのものはまだ新しいものであり、その医薬・生物学、生命情報科学分野での今後の研究・成果を楽しみにしています。SASのコーディングも比較的簡単ですので、どうぞ皆さん試してみてください。

Good Fractals! 

参考書籍

1)**「フラクタル幾何学」**
B. B. Mandelbrot 著、広中平祐監訳(日経サイエンス社 1985)
2)**「フラクタル」**高安秀樹(朝倉書店 1986)
3)**「フラクタル数学」**石村貞夫、石村園子(東京図書 1990)
4)**「カオスとフラクタル入門」**
山口昌哉ほか(日本放送出版協会 1992)
5)**「複素力学系序説」**上田哲生ほか(培風館 1995)
6)**「初めてのフラクタル」**
H. A. Lauwerier 著、西村利男訳(丸善 1996)
7)**「SAS/GRAPH リファレンスガイド」**
SAS インスティチュートジャパン(1990)

Appendix 1

■ シェルピンスキーのギャスケット

$z = x + iy$

$F_1(z) = Az$

$F_2(z) = Az + \begin{pmatrix} 1/2 \\ 0 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$

$F_3(z) = Az + \frac{1}{2} \begin{pmatrix} 1/2 \\ \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$

$K = F_1(K) \cup F_2(K) \cup F_3(K)$

$A = \begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}, \quad (x_0, y_0) = (1,0)$

発生確率 $p(F_1) = p(F_2) = p(F_3) = 1/3$

■ 楓の葉

$z = x + iy$

$F_1(z) = \begin{pmatrix} 0.8 & 0 \\ 0 & 0.8 \end{pmatrix} z + \begin{pmatrix} 0.1 \\ 0.04 \end{pmatrix}$

$F_2(z) = \begin{pmatrix} 0.5 & 0 \\ 0 & 0.5 \end{pmatrix} z + \begin{pmatrix} 0.25 \\ 0.4 \end{pmatrix}$

$F_3(z) = \begin{pmatrix} 0.355 & -0.355 \\ 0.355 & 0.355 \end{pmatrix} z + \begin{pmatrix} 0.266 \\ 0.078 \end{pmatrix}$

$F_4(z) = \begin{pmatrix} 0.355 & 0.355 \\ -0.355 & 0.355 \end{pmatrix} z + \begin{pmatrix} 0.378 \\ 0.434 \end{pmatrix}$

$K = F_1(K) \cup F_2(K) \cup F_3(K) \cup F_4(K)$

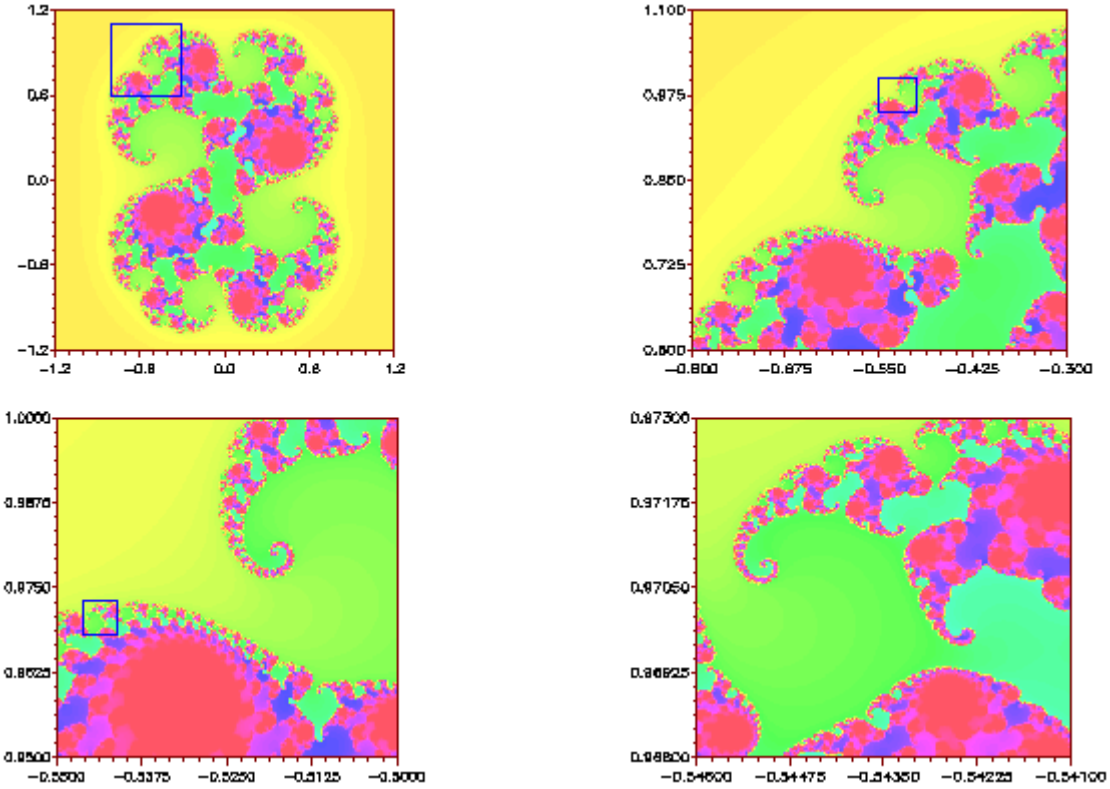
発生確率 $p(F_1) = 1/2, \quad p(F_2) = p(F_3) = p(F_4) = 1/6$

初期値 $(x_0, y_0) = (0,0)$

Appendix 2

ジュリア集合の部分拡大

部分を拡大しても同じ形をしている



Appendix 3

使用したプロシジャとカラーコード

番号	集合	プロシジャ	着色方法	備考
Fig.1	葉脈曲線	GPLOT	写像毎に RGB コード	一様乱数使用
Fig.2	シェルピンスキーのギャスケット	GPLOT	写像毎に RGB コード	一様乱数使用
Fig.3	レヴィのC曲線	GPLOT	写像毎に RGB コード	一様乱数使用
Fig.4	楓の葉	GPLOT	写像毎に RGB コード	一様乱数使用
Fig.5	マンデルブロー集合	GPLOT	反復回数毎に HLS コード	収束・発散解とも
Fig.6	ジュリア集合	GPLOT	解の大きさ$\|z_n\|$毎に HLS コード	収束解のみ
Fig.7	ジュリア集合	GPLOT	解の大きさ$\|z_n\|$毎に HLS コード	L)収束解のみ R)収束・発散解とも
Fig.8	ジュリア集合	GCONTOUR	収束・発散判定を行い HLS コード	L)収束回数で着色 R)解の大きさで着色
Fig.9	ローレンツアトラクタ	G3D	時間 t 毎に HLS コード	R=8
Fig.10	ローレンツアトラクタ	G3D	時間 t 毎に HLS コード	R=15,ベナール対流
Fig.11	ローレンツアトラクタ	G3D	時間 t 毎に HLS コード	R=28
Fig.12	エノン写像	GPLOT	HLS コード	θ =76.11°
Fig.13	ミラの写像	GPLOT	HLS コード	第1式
Fig.14	ミラの写像	GPLOT	RGB コード	第1式
Fig.15	ミラの写像	GPLOT	RGB コード	第2式
App2	ジュリア集合拡大図	GCONTOUR	収束・発散判定を行い反復回数毎に HLS コード	

Appendix 4

Esthetique Chaos グモフスキーとミラの写像

パラメータのわずかな変化で様々に変形(a=0.79～-0.39)