科学・技術の世界
数式処理システムによる新時代の数学
--- 反復関数系とフラクタル ---

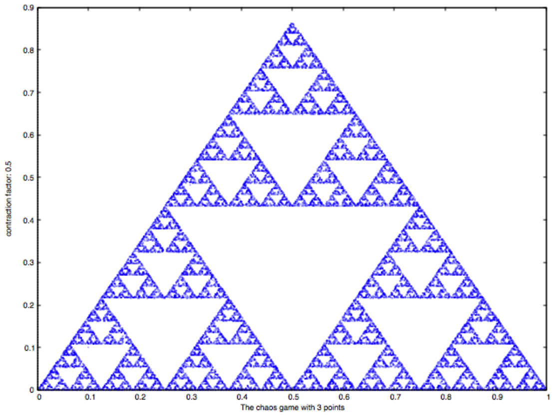
佐藤譲 (Yuzuru Sato)

RIES / Department of Mathematics

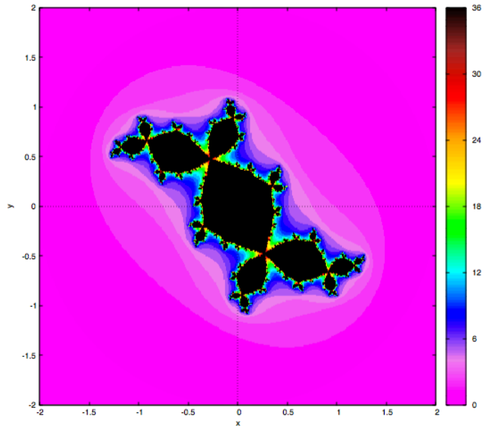
Hokkaido University

第15回 29 January, 2018

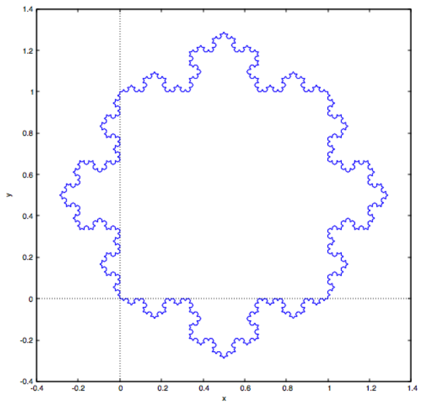
Fractals



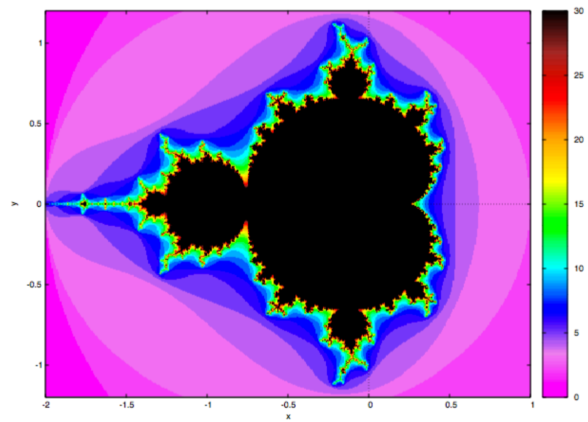
Sierpinski gasket



Duady rabbit



Koch snowflakes



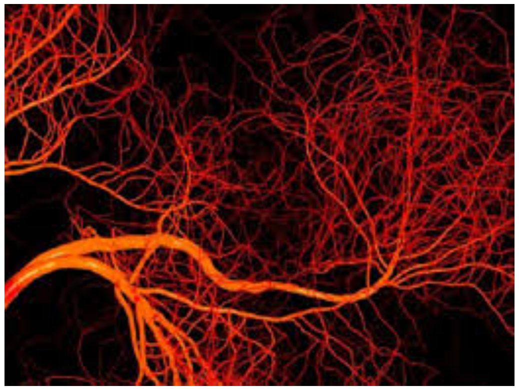
Mandelbrot set

[generated by Maxima sample codes]

Fractals


墨流し

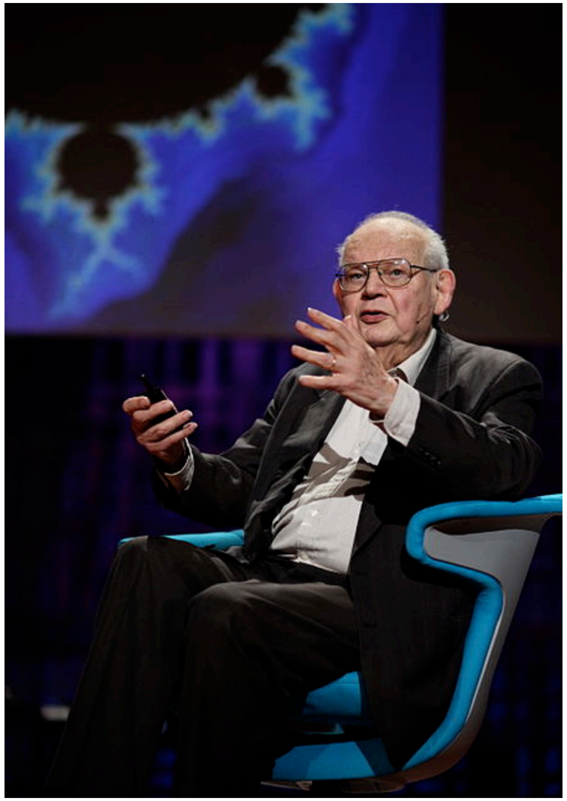

イモガイの外殻


ヒトの血管


ロマネスコの蕾

[from wikipedia]

Fractals



Benoit B. Mandelbrot
(1924 –2010)
Applued mathematician at
IBM Thomas J. Watson
Resaerch Center, USA

Scaling

How Long Is the Coast of Britain?
Statistical Self-Similarity and Fractional Dimension

Abstract. *Geographical curves are so involved in their detail that their lengths are often infinite or, rather, undefinable. However, many are statistically "self-similar," meaning that each portion can be considered a reduced-scale image of the whole. In that case, the degree of complication can be described by a quantity* D *that has many properties of a "dimension," though it is fractional; that is, it exceeds the value unity associated with the ordinary, rectifiable, curves.*

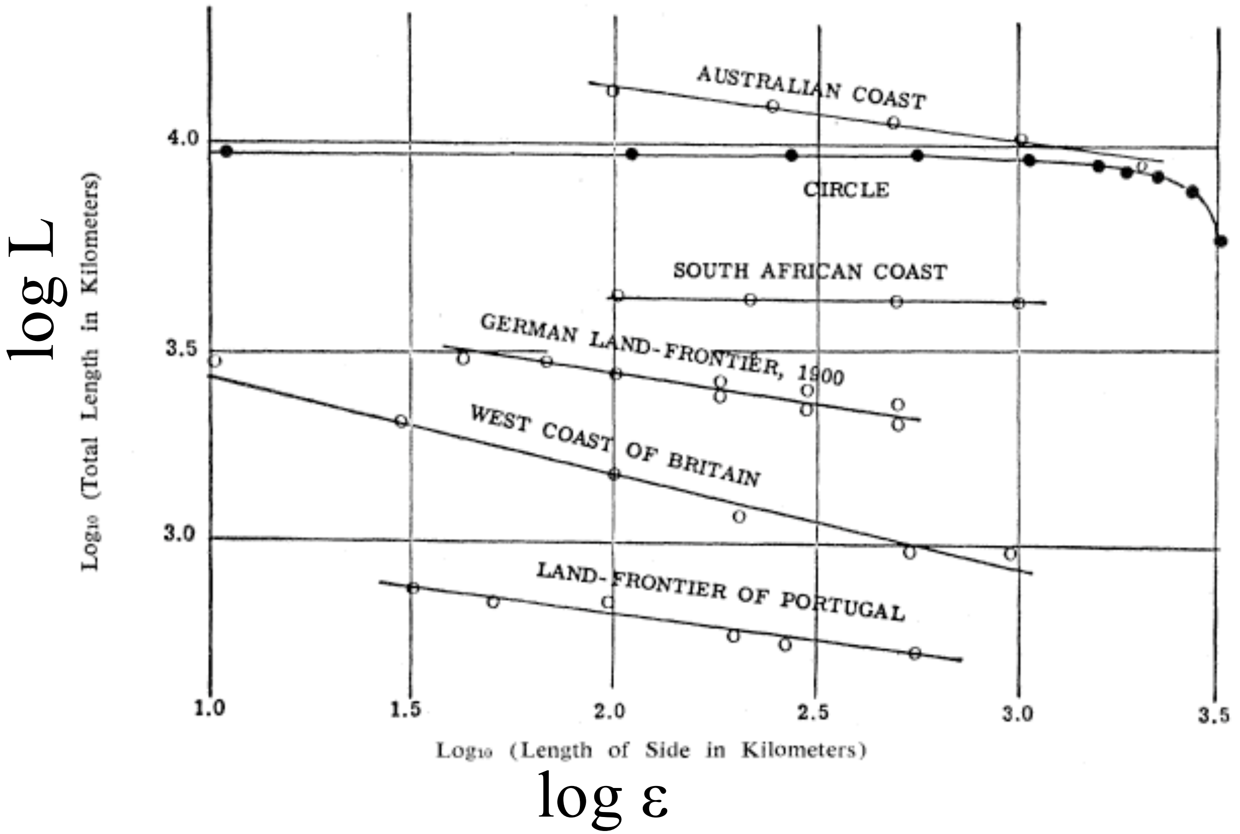
BENOIT MANDELBROT
International Business Machines,
Thomas J. Watson Research Center,
Yorktown Heights, New York 10598

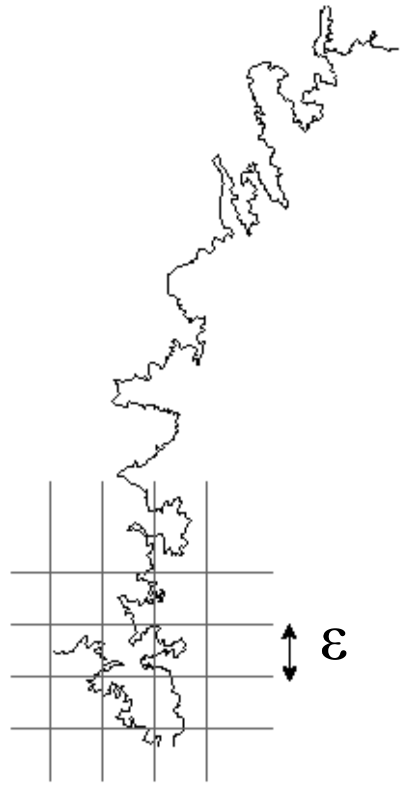
14 November 1966; 27 March 1967

SCIENCE, VOL. 156

Scaling

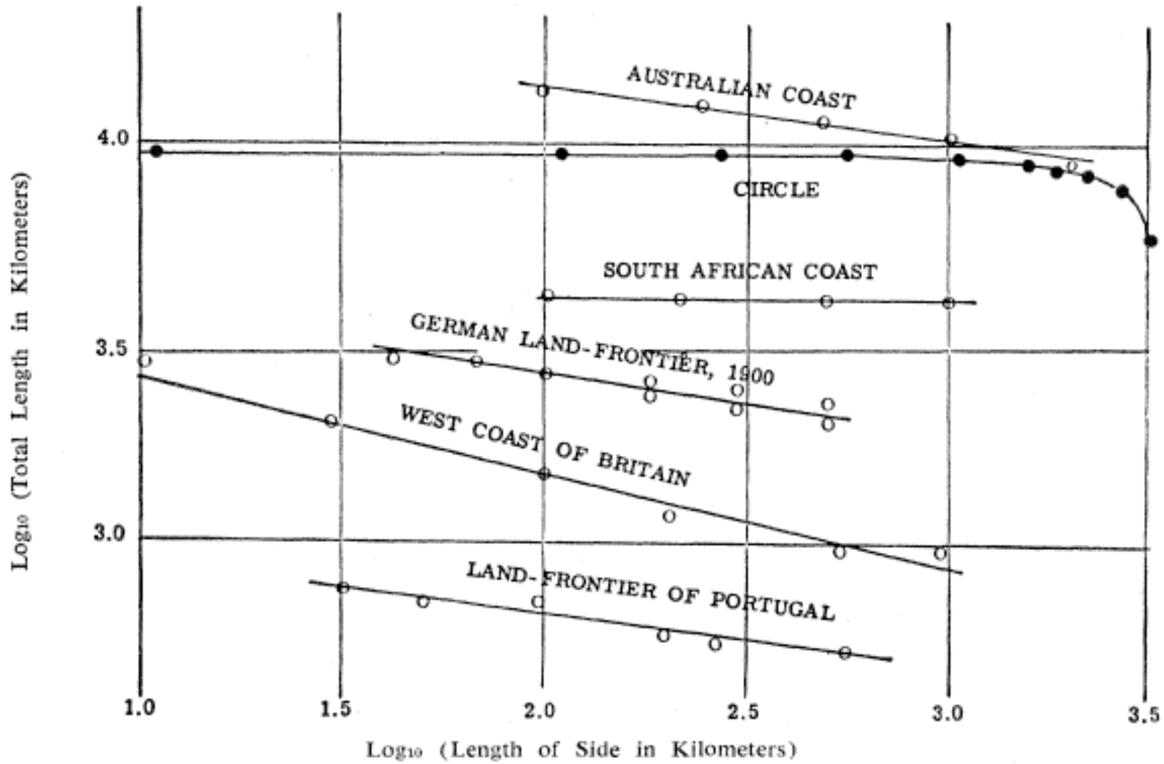
• 海岸線の長さの計測 (Richardson, 1961)

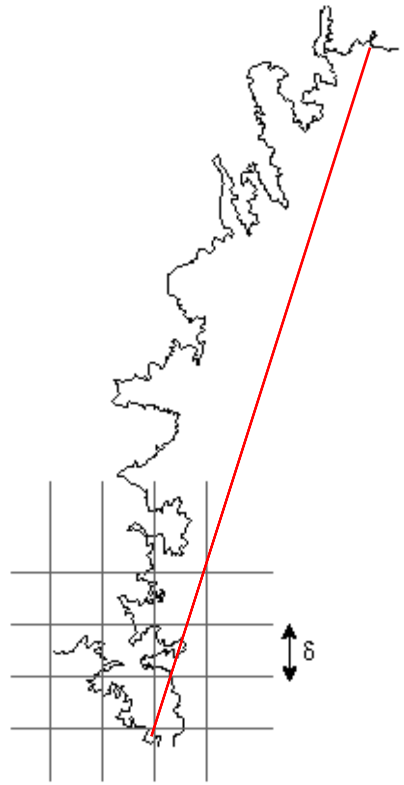




Scaling

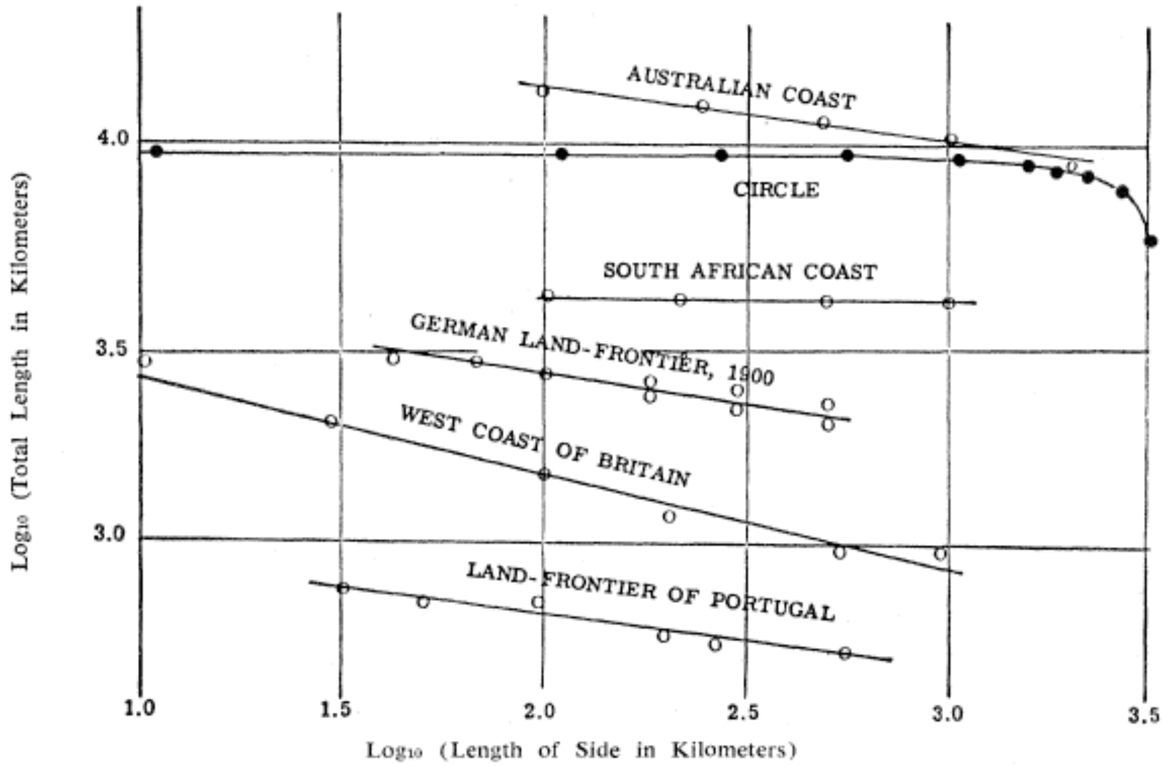
- 海岸線の長さの計測 (Richardson, 1961)

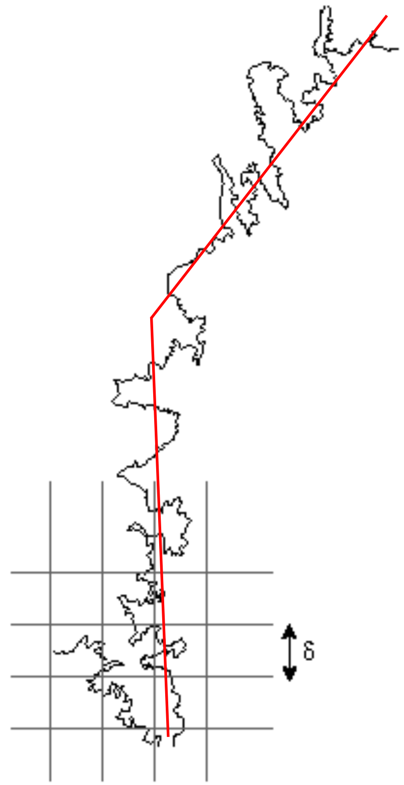




Scaling

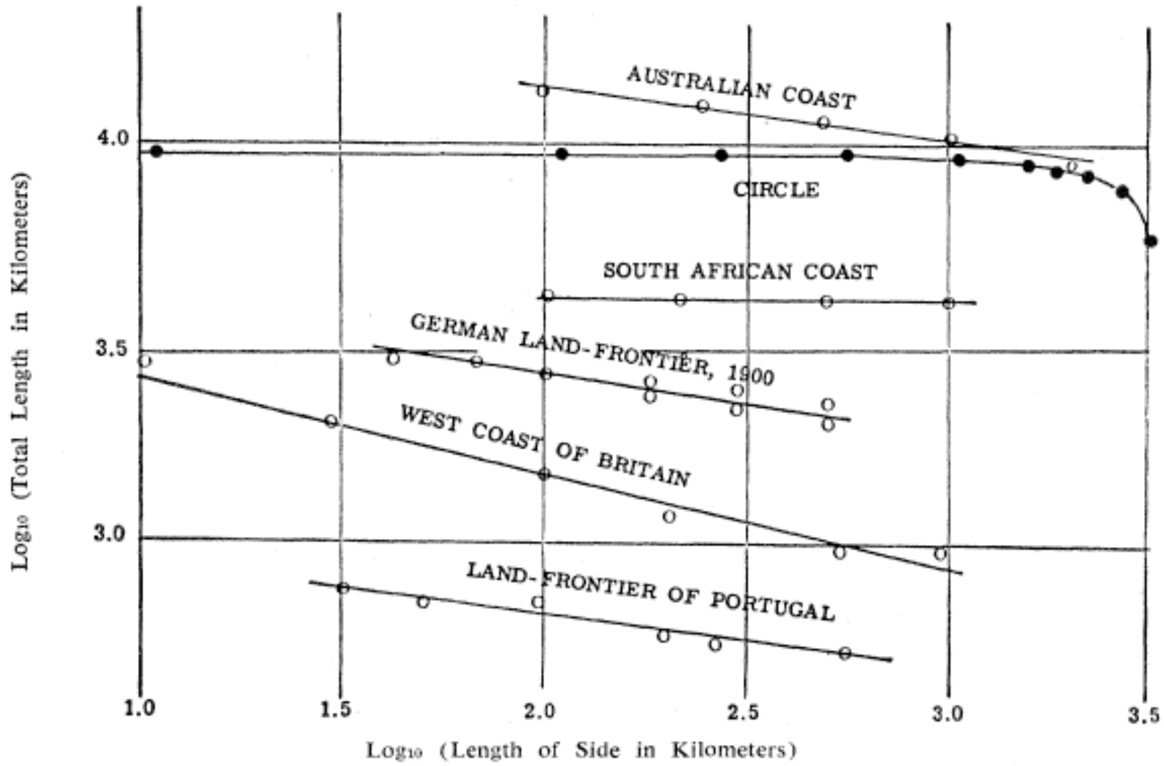
• 海岸線の長さの計測 (Richardson, 1961)

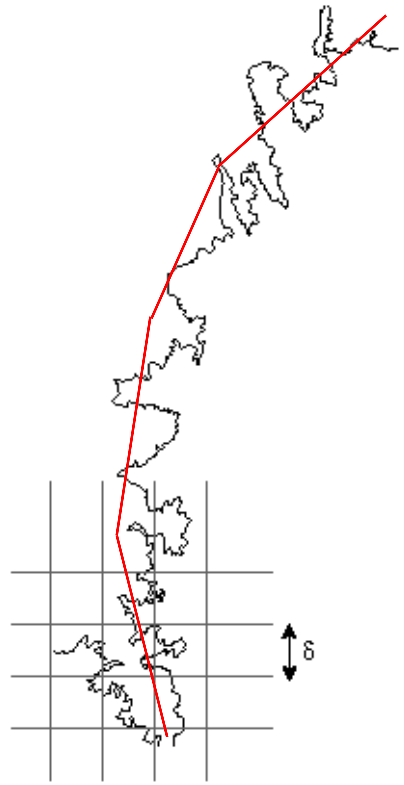




Scaling

- 海岸線の長さの計測 (Richardson, 1961)



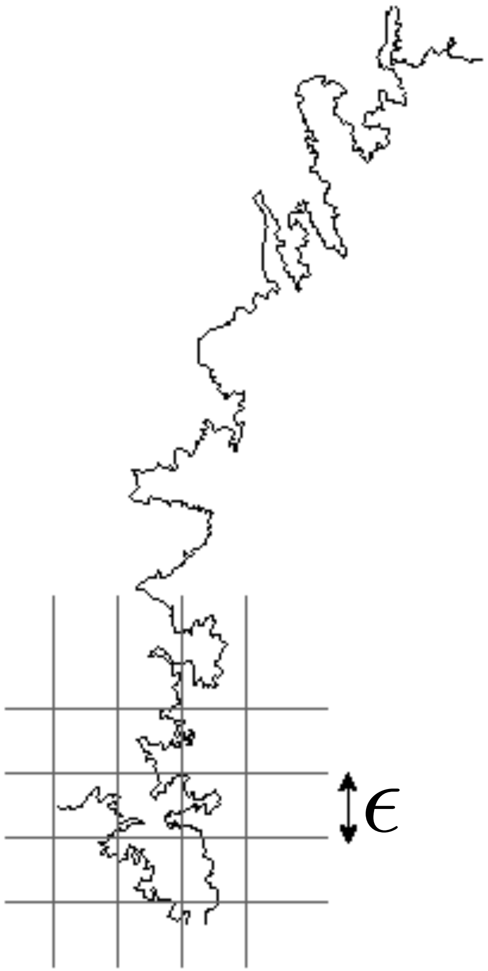


Scaling

- 境界線を折れ線で近似するとき、長さ ϵ の区間が約 $C\epsilon^{-D}$ 個必要となる場合。

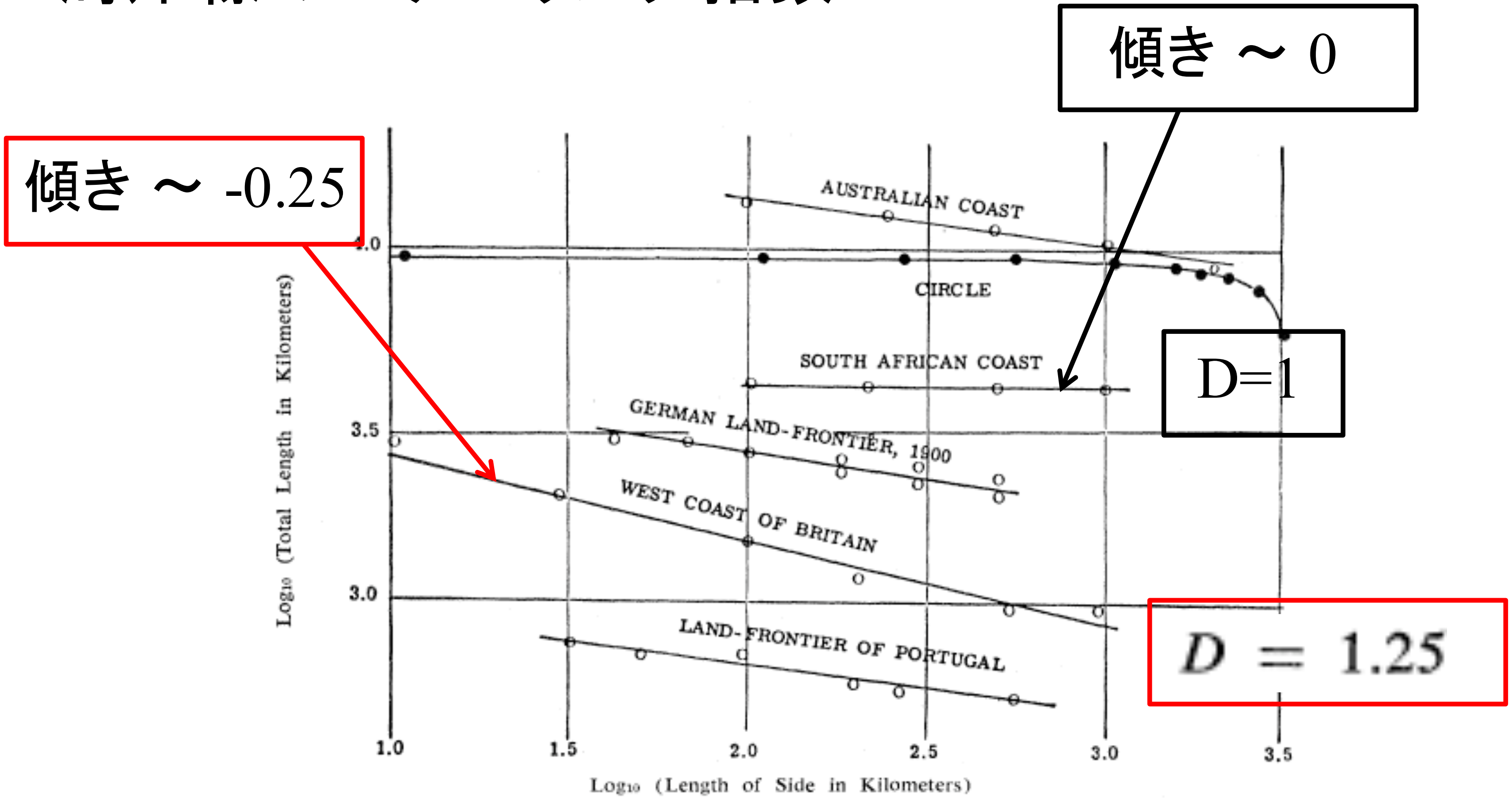
- 総計で長さは以下。

$$L(\epsilon) \sim C\epsilon^{-D} \cdot \epsilon = C\epsilon^{1-D}$$



Scaling

- 海岸線のスケーリング指数



Scaling

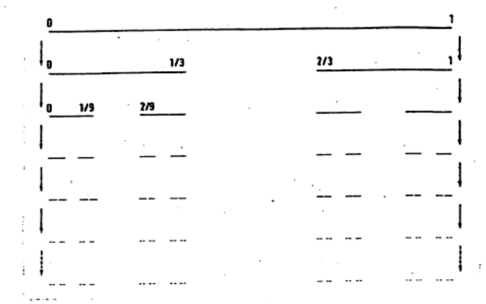
- スケーリング指数の普遍性
  - スペインとポルトガルによって主張されている両国間の国境の長さ(987kmと1214km)、オランダとベルギーによって主張されている両国間の国境の長さ(380kmと449km)の違いは、隣り合う国同士で異なる ϵ を使っていることで説明される。

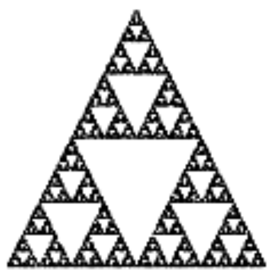
[B. Mandelbrot, 1977]

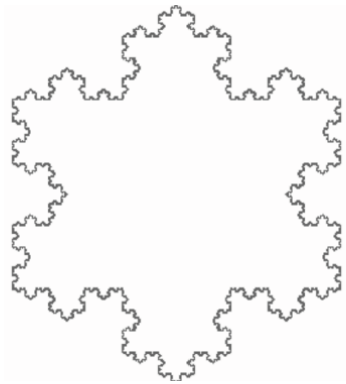
Concept of fractals

- 観測スケールに対して不変な幾何構造をフラクタルとよぶ。(Mandelbrot, 1977)

  - 特徴的スケールをもたない幾何構造
  - 自己相似性をもつ幾何構造
  - 非整数の次元をもつ幾何構造







Iterated function systems

$\mathbf{R}^n$ 上の反復関数系とは、確率 $p_1, p_2, \ldots, p_m$ を伴う $v \in \mathbf{R}^n$ に作用するm個の写像の組

$$f_1, f_2, \ldots, f_m$$

$$f_i(v) = L_i v + c_i$$

L_i : 固有値が全て1より小さい n × n 行列

c_i : n次元ベクトル

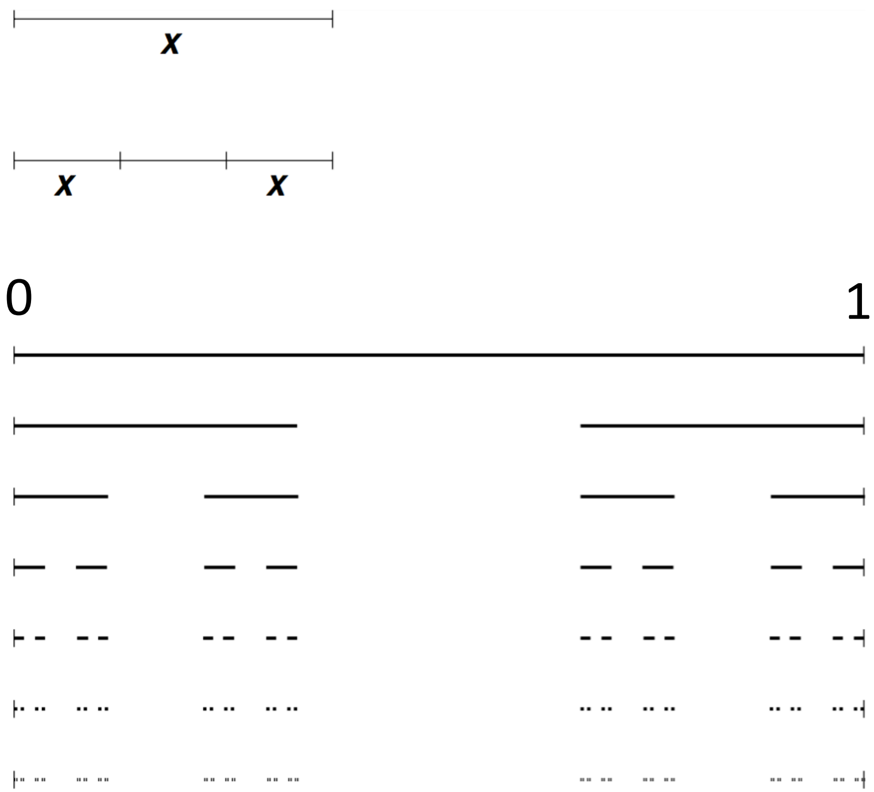
のことをいう。それぞれの(縮小)写像 f_i は確率 p_iで選ばれ、初期値に反復適用される。

Example: Cantor set

$$f_1(x) = x/3$$

$$f_2(x) = x/3 + 2/3$$

$$p_1 = p_2 = 1/2$$



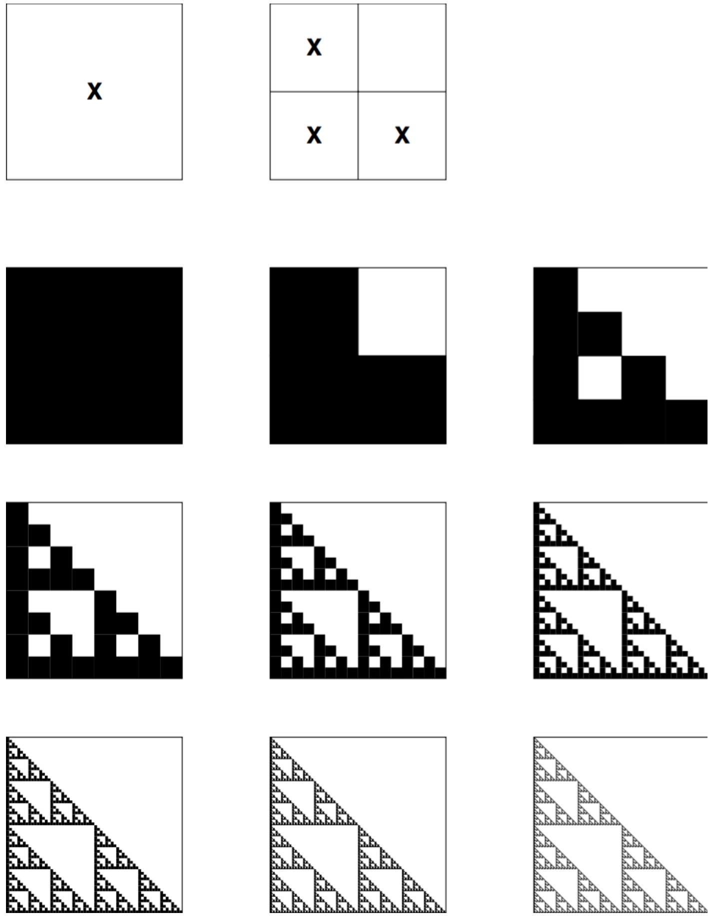
Example: Sierpinski gasket

$$f_1(x, y) = (x/2, y/2)$$

$$f_2(x, y) = ((1 + x)/2, (y/2))$$

$$f_3(x, y) = (x/2, (1 + y)/2)$$

$$p_1 = p_2 = p_3 = 1/3$$



Collage theorem

[IFSの不動点定理, Barnsley (1985)]

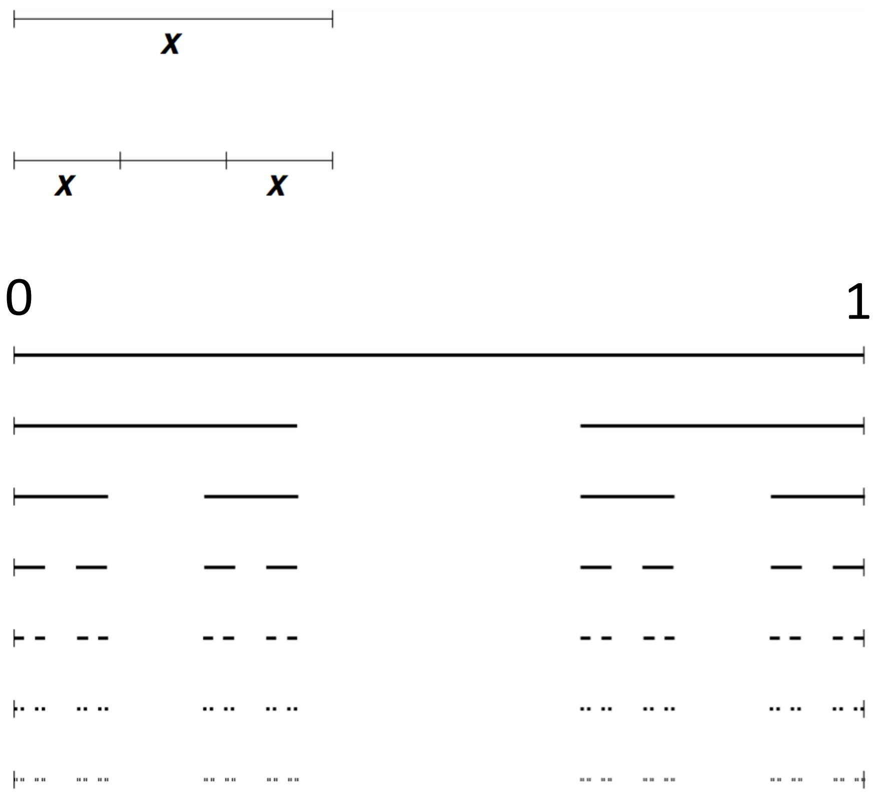
$f_1, f_2, \ldots, f_m$ が全て縮小写像であるIFSにはただ一つの極限集合が存在する。

Cantor set

$$f_1(x) = x/3$$

$$f_2(x) = x/3 + 2/3$$

$$p_1 = p_2 = 1/2$$

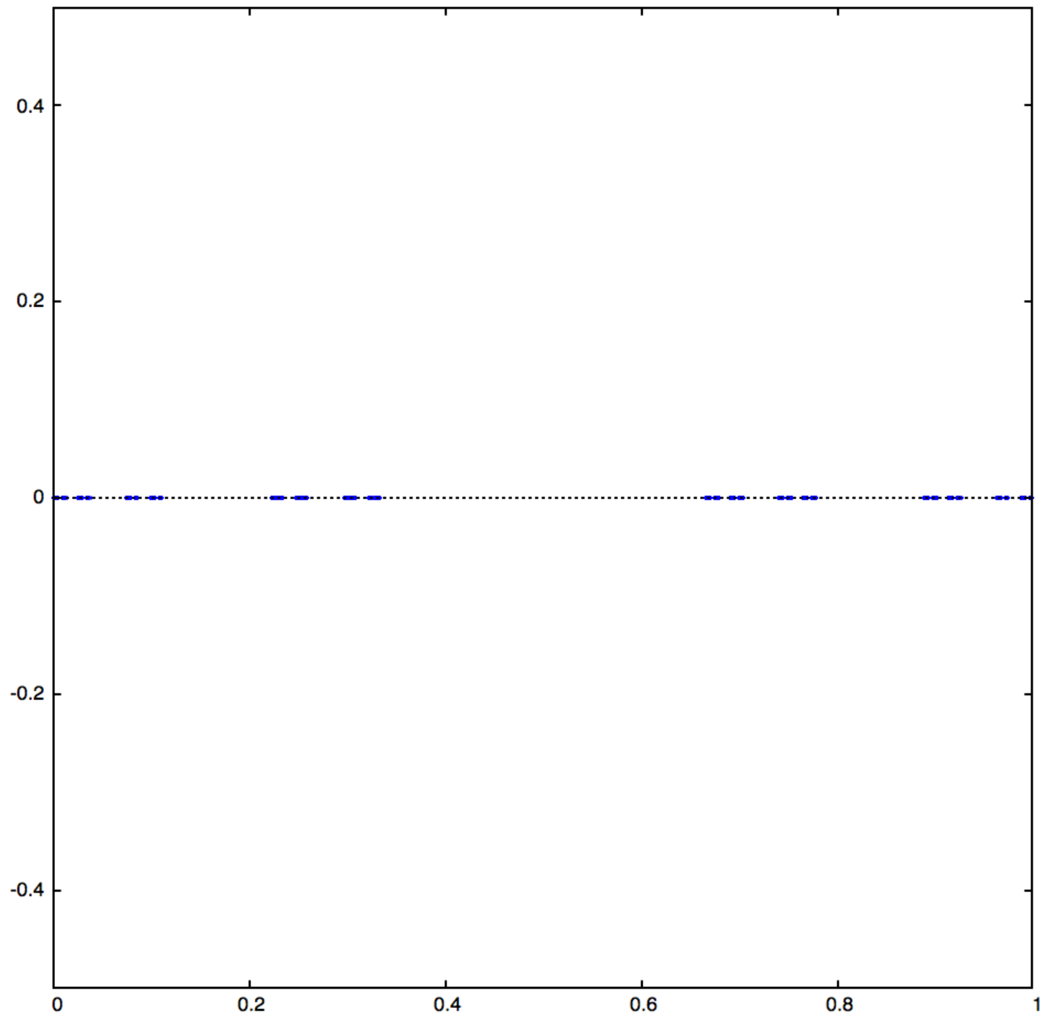


Cantor set

Maxima code

```
n:10000;
array(x,n);  array(rnd,n);
x[1]:random(1.0);
for i:1 thru n do rnd[i]:random(2);
a1:1/3;  a2:1/3;  p1:0;  p2:2/3;

cs(x,r):=block(if(r=0) then return(a1*x+p1) else return(a2*x+p2));
for i:1 thru n-1 do(x[i+1]:bfloat(cs(x[i],rnd[i])));

data:makelist([x[i],0],i,n/2,n)$
plot2d([discrete,data],[x,0,1],[y,-0.5,0.5],[style,[points,0.1]],same_xy)$
```

Cantor set



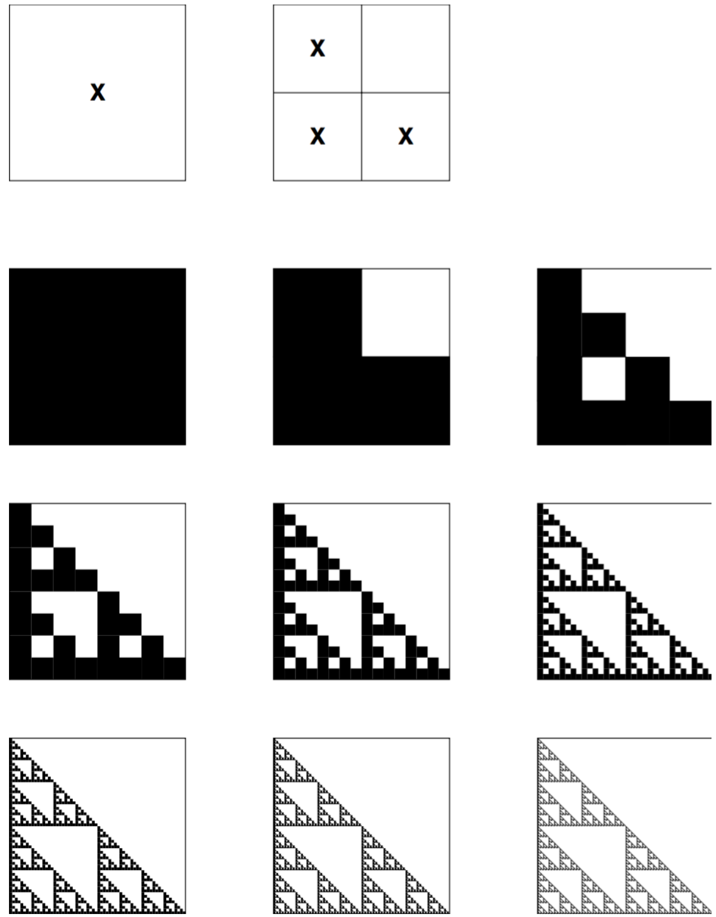
Sierpinski gasket

$$f_1(x, y) = (x/2, y/2)$$

$$f_2(x, y) = ((1 + x)/2, (y/2))$$

$$f_3(x, y) = (x/2, (1 + y)/2)$$

$$p_1 = p_2 = p_3 = 1/3$$

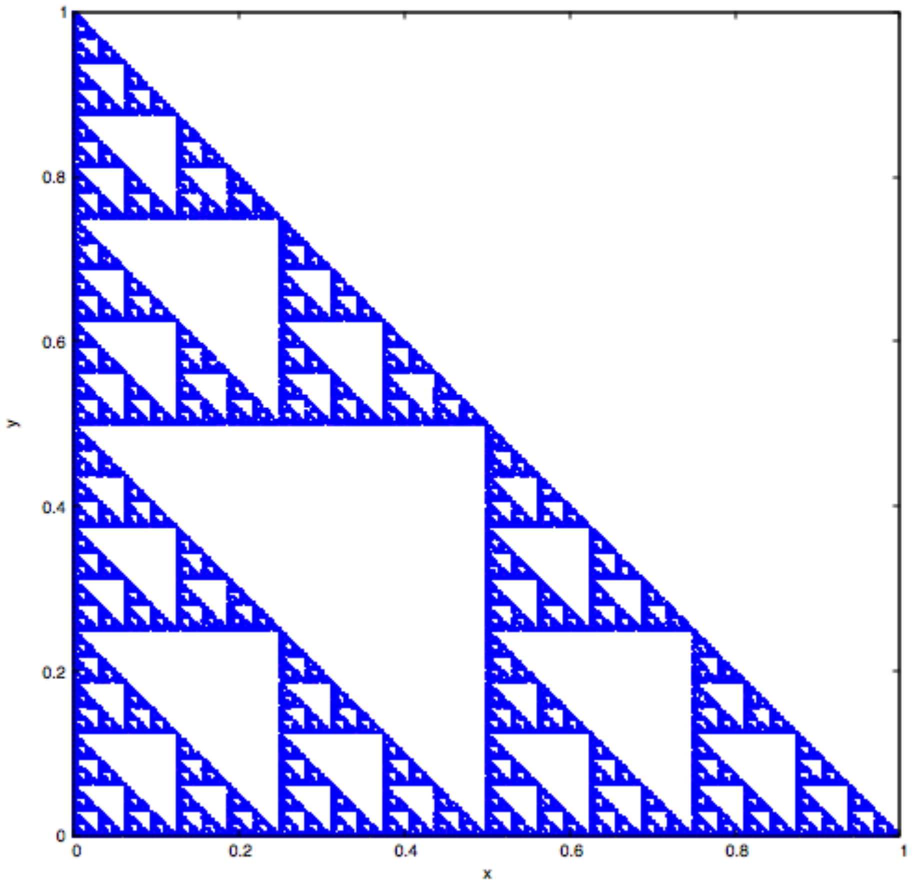


Sierpinski gasket

Maxima code

```
n:100000;
array(x,n);  array(y,n);  array(rnd,n);
x[1]:random(1.0)$  y[1]:random(1.0);
for i:1 thru n do rnd[i]:random(3);
a1: matrix([1/2,0],[0,1/2]); a2: matrix([1/2,0],[0,1/2]); a3: matrix([1/2,0],[0,1/2]);
p1: [0,0];  p2: [1/2,0];  p3: [0,1/2];

sgx(x,y,r):=block(if(r=0)  then return([1,0].(a1.[x,y]+p1))
else if(r=1) then return([1,0].(a2.[x,y]+p2))
else return([1,0].(a3.[x,y]+p3)));
sgy(x,y,r):=block(if(r=0)  then return([0,1].(a1.[x,y]+p1))
else if(r=1) then return([0,1].(a2.[x,y]+p2))
else return([0,1].(a3.[x,y]+p3)));

for i:1 thru n-1 do (x[i+1]:bfloat(sgx(x[i],y[i],rnd[i])), y[i+1]:bfloat(sgy(x[i],y[i],rnd[i])));
data:makelist([x[i],y[i]],i,n/2,n)$
plot2d([discrete,data],[x,0,1],[y,0,1],[style,[points,0.1]],same_xy)$
```

Sierpinski gasket



Barnsley fern

$$f_1(x,y) = \begin{bmatrix} 0.00 & 0.00 \\ 0.00 & 0.16 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$f_2(x,y) = \begin{bmatrix} 0.85 & 0.04 \\ -0.04 & 0.85 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.00 \\ 1.60 \end{bmatrix}$$

$$f_3(x,y) = \begin{bmatrix} 0.20 & -0.26 \\ 0.23 & 0.22 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.00 \\ 1.60 \end{bmatrix}$$

$$f_4(x,y) = \begin{bmatrix} -0.15 & 0.28 \\ 0.26 & 0.24 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 0.00 \\ 0.44 \end{bmatrix}$$

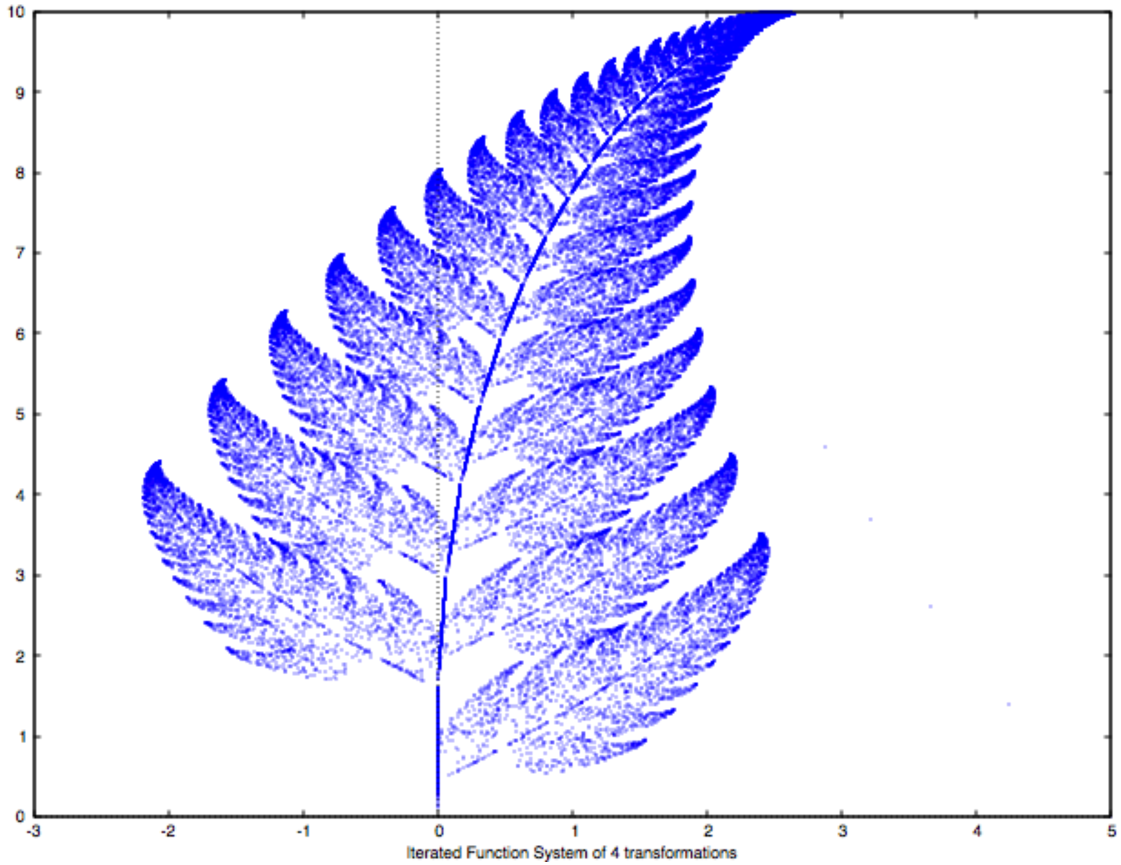
$$p_1 = 0.01,\ p_2 = 0.85,\ p_3 = p_4 = 0.07$$

Barnsley fern

Maxima code

```
a1: matrix([0.85,0.04],[-0.04,0.85])$
a2: matrix([0.2,-0.26],[0.23,0.22])$
a3: matrix([-0.15,0.28],[0.26,0.24])$
a4: matrix([0,0],[0,0.16])$p1: [0,1.6]$
p2: [0,1.6]$
p3: [0,0.44]$
p4: [0,0]$
w: [85,92,99,100]$

ifs(w, [a1,a2,a3,a4], [p1,p2,p3,p4], [5,0], 50000, [style,dots]);
```

Barnsley fern

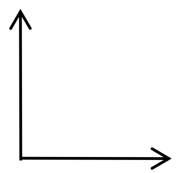


Filled Julia set

- $Z_{n+1} = Z_n^2 + C, \quad Z_n, C$ は複素数

  Cを固定したとき、すべてのnについて軌道が有界にとどまる相空間内の点集合。

Im(z)



Re(z)

c = −0.75

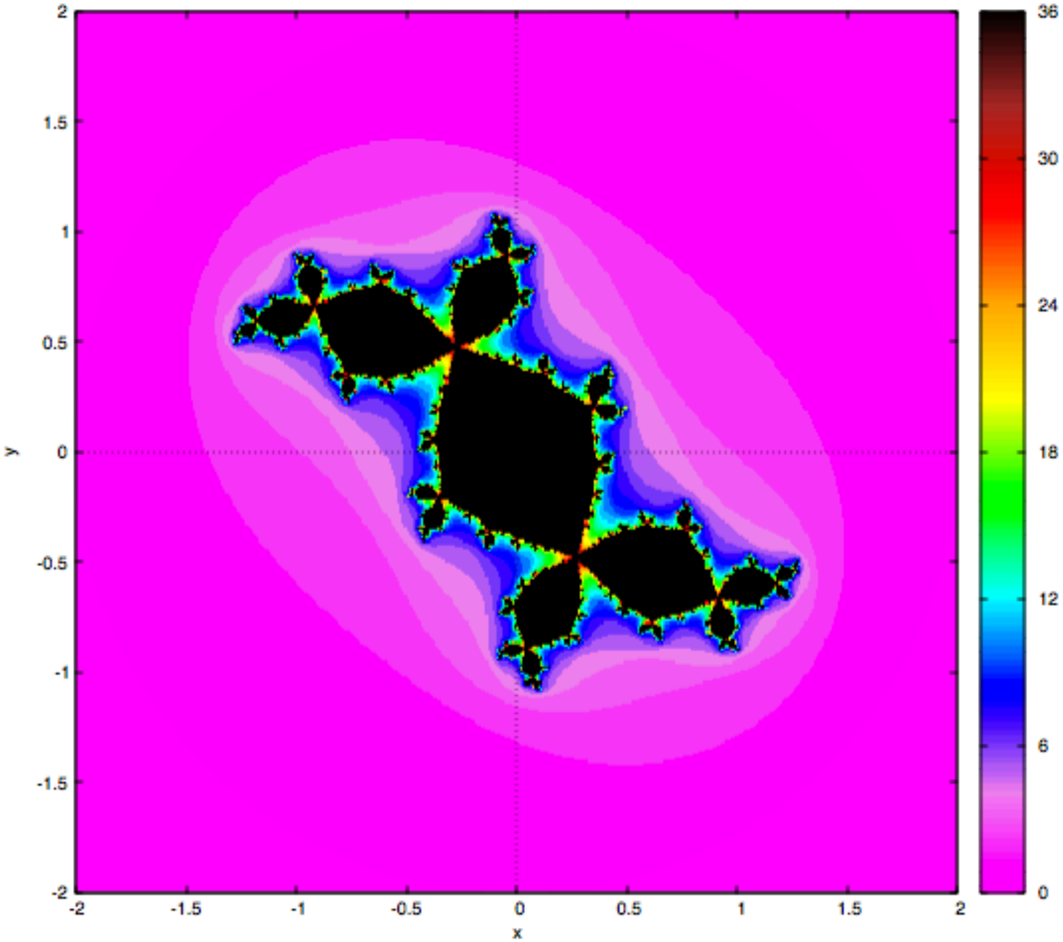
c = −0.75 + 0.1i

Filled Julia set

Maxima code

```
julia (-0.122, 0.745, [iterations, 36],[x, -2, 2],[y, -2, 2], [grid, 400, 400],
[color_bar_tics, 0, 6, 36])$
```

Filled Julia set

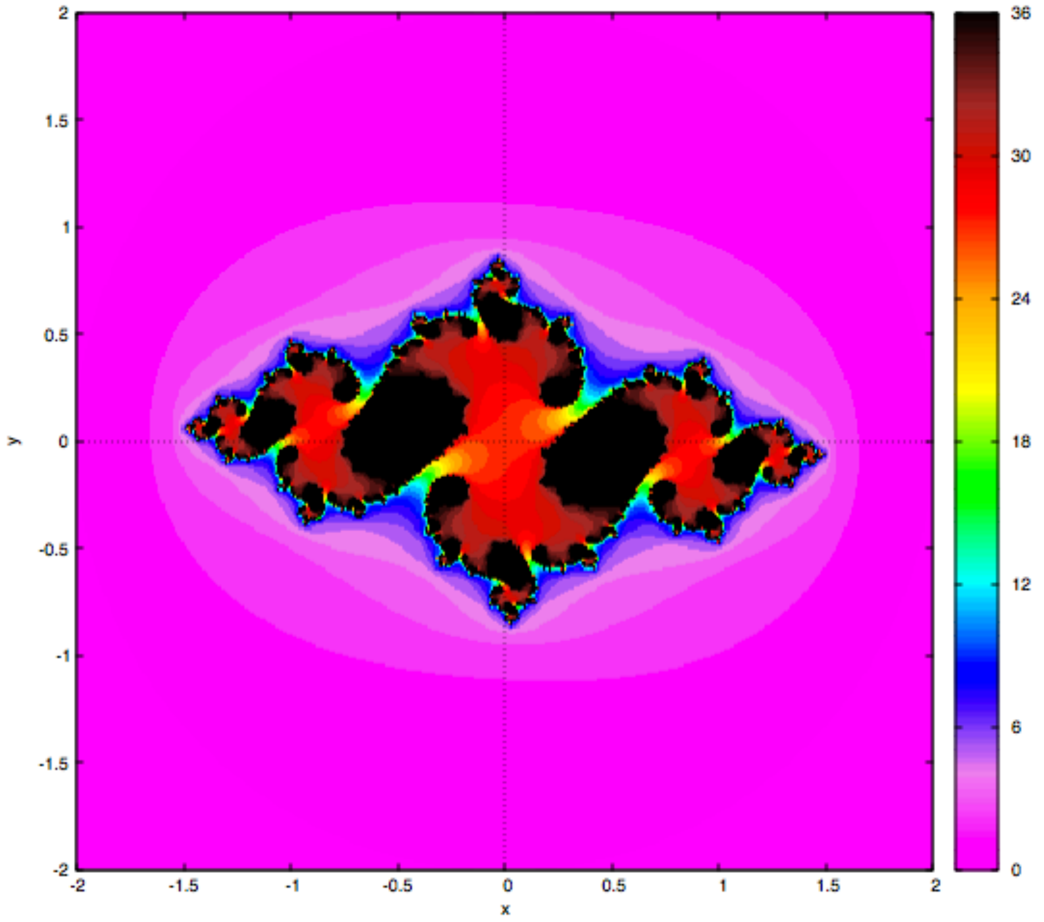


Filled Julia set

Maxima code

```
julia (-0.754, 0.113, [iterations, 36],[x, -2, 2],[y, -2, 2], [grid, 400, 400],
[color_bar_tics, 0, 6, 36])$
```

Filled Julia set

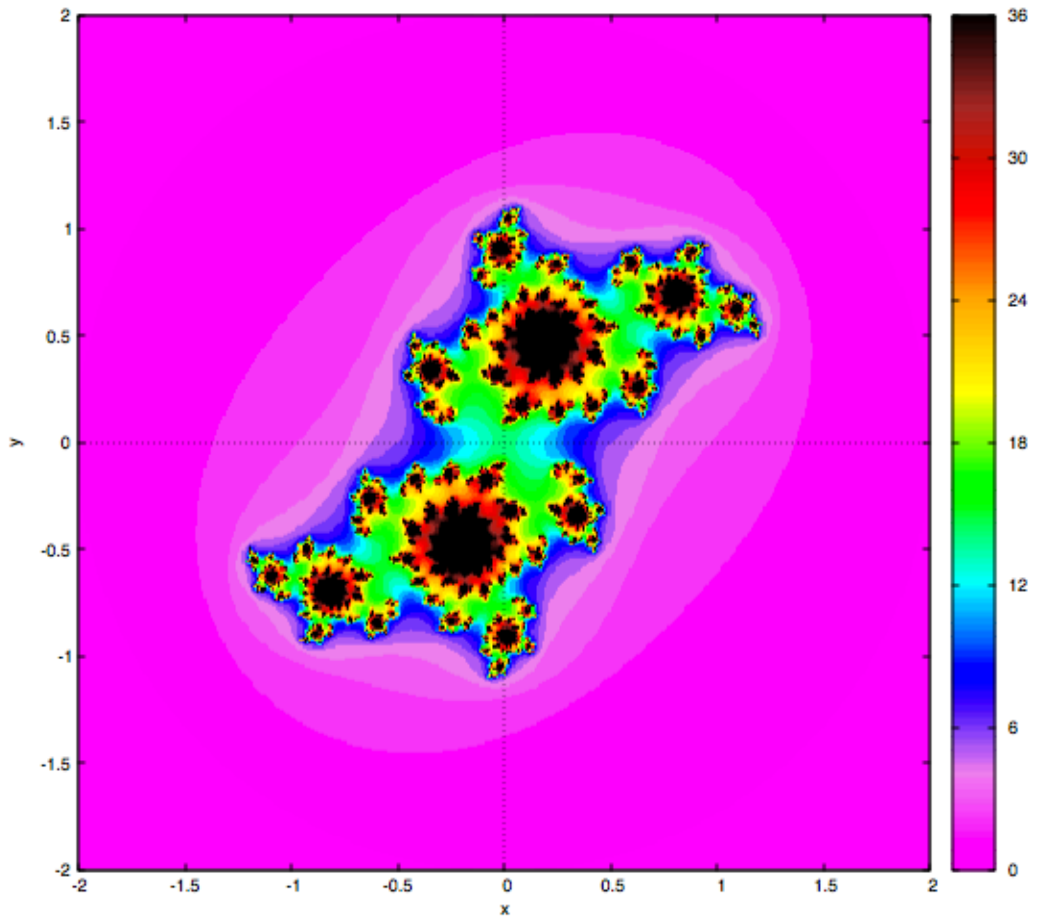


Filled Julia set

Maxima code

```
julia (0.011, -0.670, [iterations, 36],[x, -2, 2],[y, -2, 2], [grid, 400, 400],
[color_bar_tics, 0, 6, 36])$
```

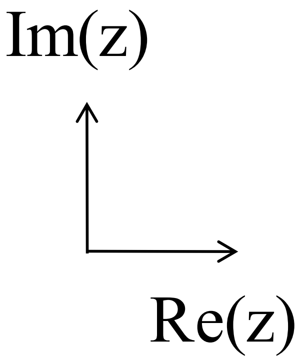
Filled Julia set

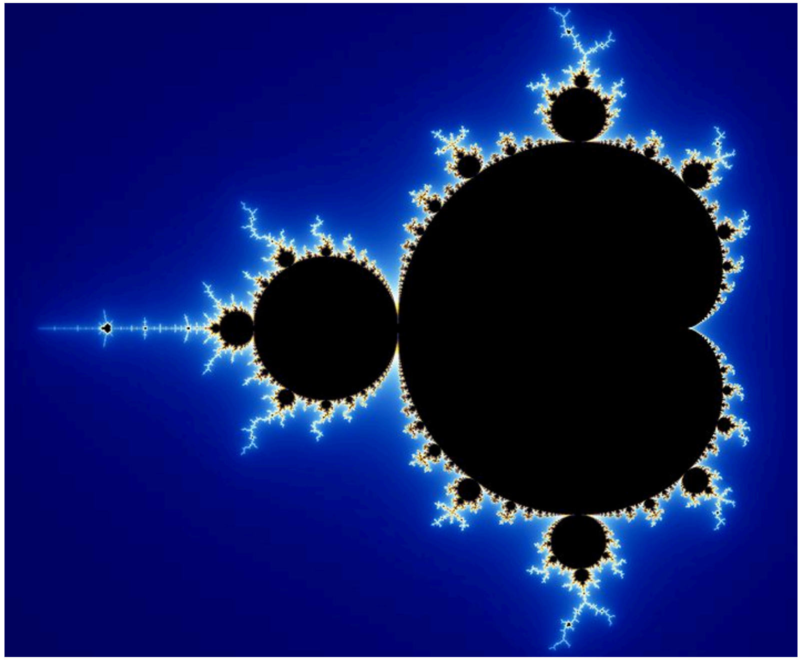


Mandelbrot set

- $Z_{n+1} = Z_n^2 + C, \quad Z_n, C$ は複素数

Z=0が有界数列を生成するようなCの集合。



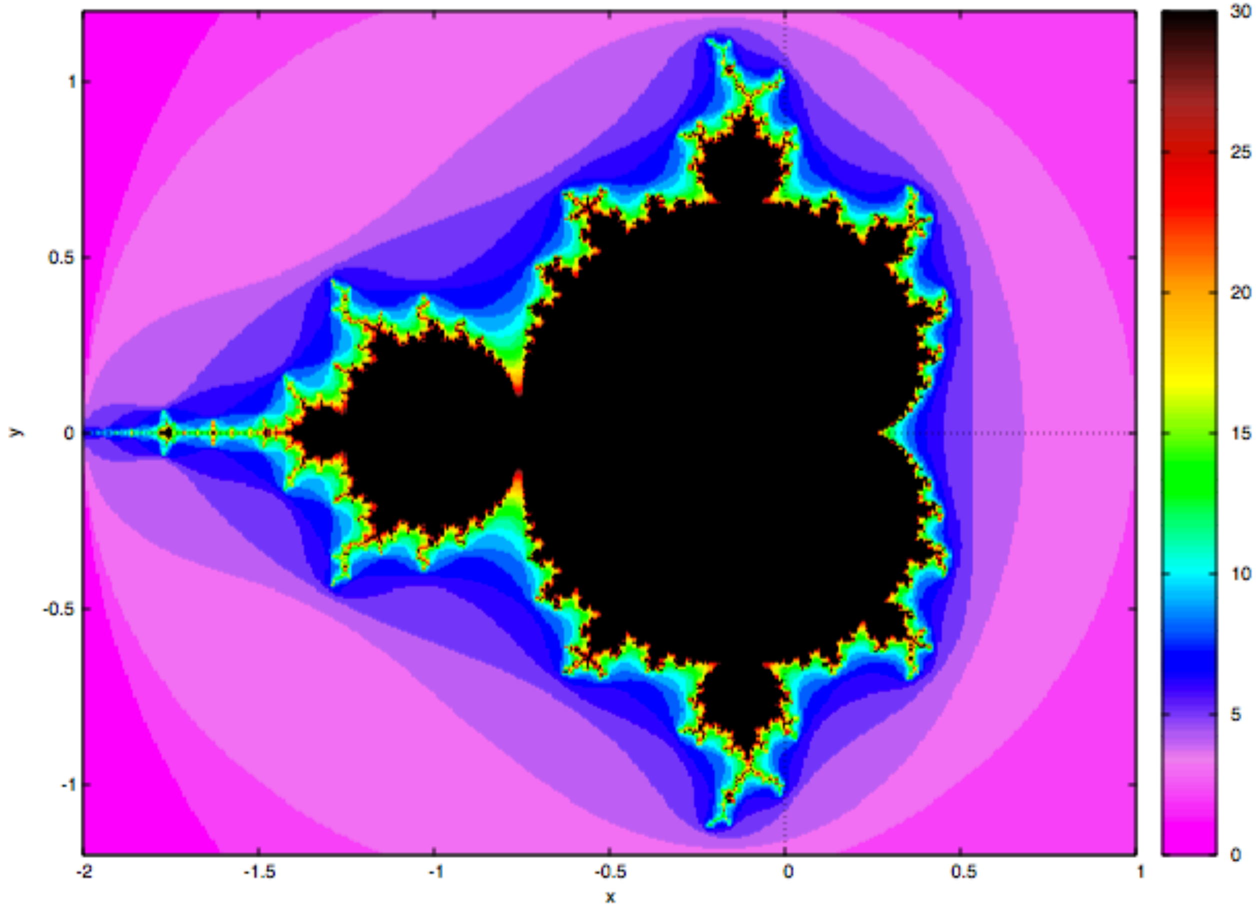


Mandelbrot set

Maxima code

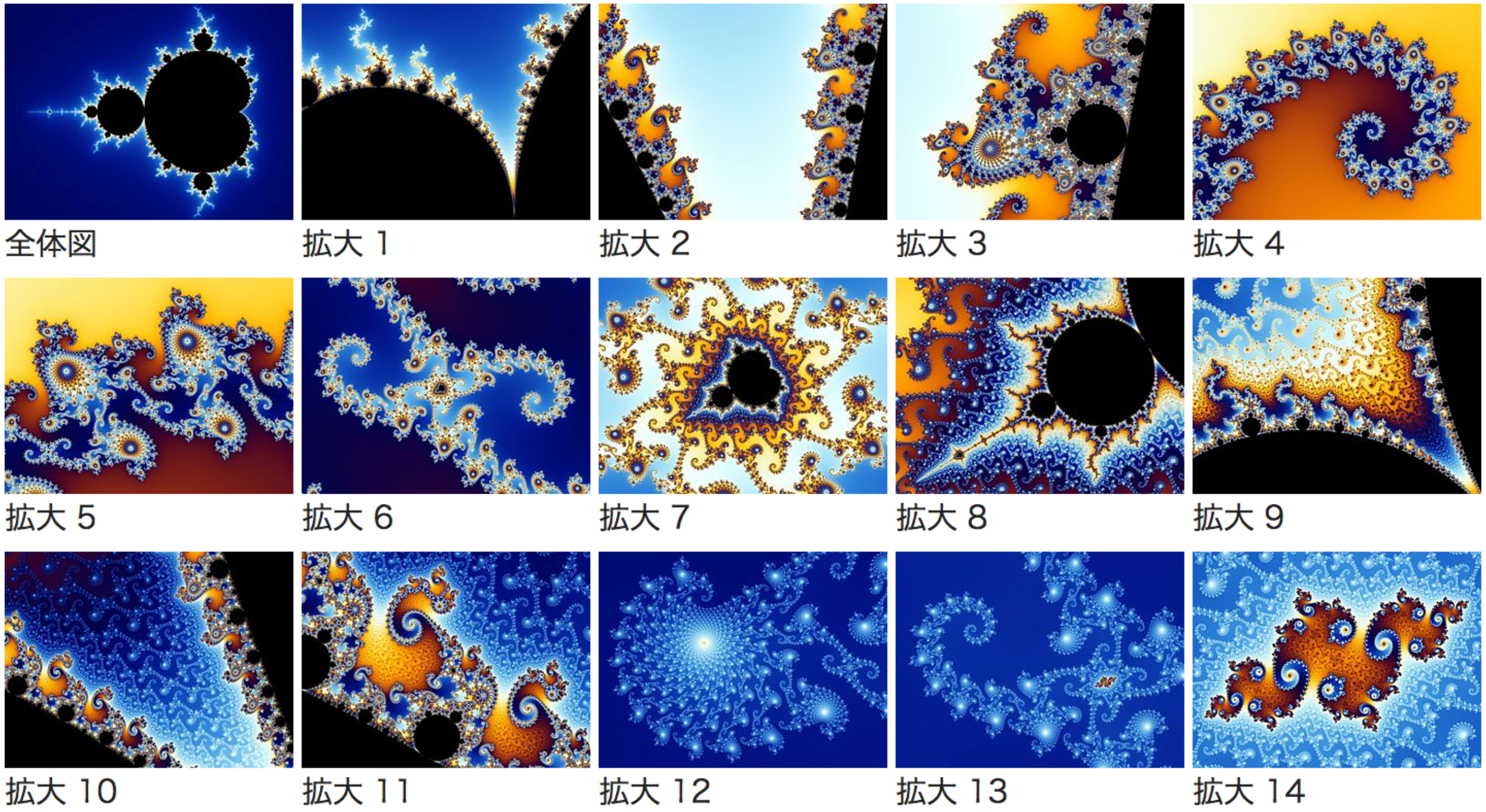
```
mandelbrot ([iterations, 30], [x, -2, 1], [y, -1.2, 1.2], [grid,400,400])$
```

Mandelbrot set



Mandelbrot set

- 連結かつ境界のHausdorff次元が2(=位相次元)である複雑なフラクタル。



Topological dimension

- 一点からなる集合{p}に対して

$$dim_T(\{p\}) = 0$$

- 単位線分 I^1 に対して

$$dim_T(I^1) = 1$$

- 単位正方形 I^2 に対して

$$dim_T(I^2) = 2$$

- 一般にm次元超立方体 I^m に対して

$$dim_T(I^m) = m$$

Hausdorff dimension

- R^n内の集合Xを直径が ϵ 以下の閉集合で覆う。閉集合の直径を $\epsilon_1, \epsilon_2, \ldots, < \epsilon$ としてX の最小被覆 を構成し、$\Lambda_d(E) = \inf \sum_k \epsilon_k^d$ を考える。このとき

$$\Lambda_d(E) = \begin{cases} \infty & (d < d^*) \\ 0 & (d > d^*) \end{cases}$$

となるd^*をHausdorff次元 $dim_H(X)$ という。

Definition of fractals

- R^n上の集合Xがフラクタルであるとは

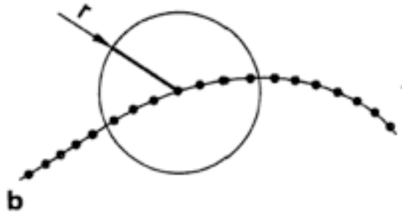
$$dim_T(X) < dim_H(X)$$

が成り立つときにいう。

[山口昌也, 1993]

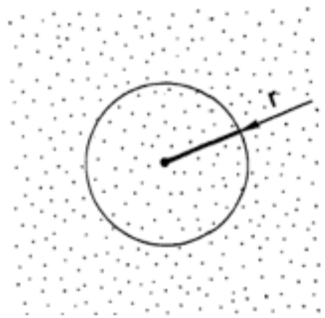
Scale invariance and dimension

- n次元集合に対して、その集合上の点を中心とする半径rの球の中に入っている点の数をN(r)とすると、次元νは $N(r) \sim r^{\nu}$ で与えられる。



線分

$$N(r) \sim r^1$$



面

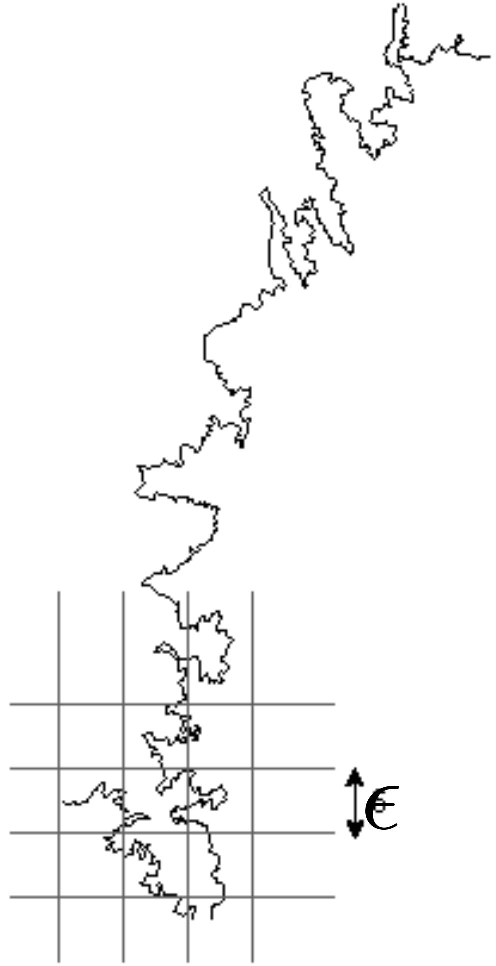
$$N(r) \sim r^2$$

Scale invariance and dimension

- 相空間を一辺 ϵ の区間で分割し、与えられた集合Xを覆うために区間がいくつ必要になるかを数える。

- 区間が $C\epsilon^{-D}$ 個必要となるとき、スケーリング則よりDを求めることができる。

$$N(\epsilon) \sim C\epsilon^{-D}$$

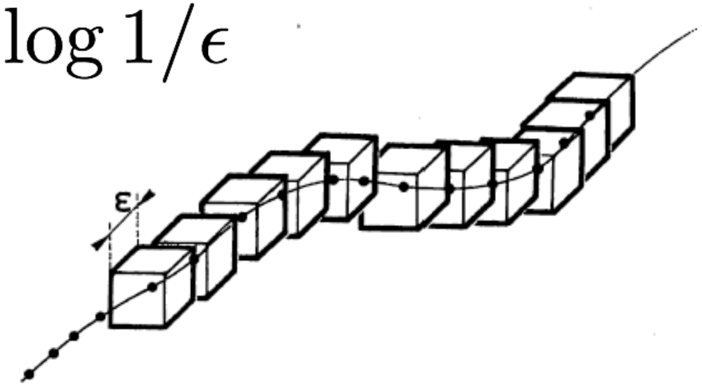


Box-counting dimension

- n次元空間内の点集合Xを一辺がεのn次元の立方体で覆う。この被覆に必要な最小の立方体の数をN(ε)とすると、Hausdorff次元の上限は、以下のBox-counting次元 $D(= dim_B(X))$ で与えられる。

$$N(\epsilon) \sim \epsilon^D \iff D \sim \frac{\log N(\epsilon)}{\log 1/\epsilon}$$

$$D = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$



$$dim_T(X) \leq dim_H(X) \leq dim_B(X)$$

Box-counting dimension

- 一点

$$N(\epsilon) = 1, D = 0$$

- 長さLの線分

$$N(\epsilon) = L \cdot \epsilon^{-1}$$

$$D = \lim_{\epsilon \to 0} \frac{\log(L \cdot \epsilon^{-1})}{\epsilon^{-1}} = 1$$

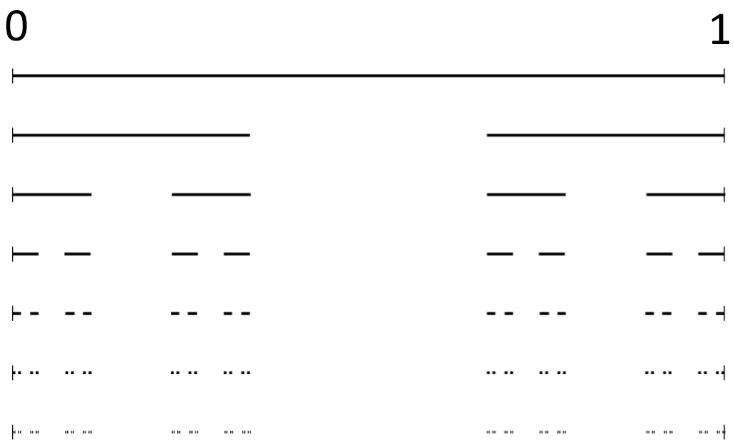
$$D = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$

- 面積Sの局面

$$N(\epsilon) = S \cdot \epsilon^{-2}$$

$$D = \lim_{\epsilon \to 0} \frac{\log(S \cdot \epsilon^{-2})}{\epsilon^{-1}} = 2$$

Box-counting dimension of Cantor set



$$D = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$

$$\epsilon = 1/3, N(\epsilon) = 2$$

$$\epsilon = (1/3)^2, N(\epsilon) = 2^2$$

$$\vdots$$

$$\epsilon = (1/3)^m, N(\epsilon) = 2^m$$

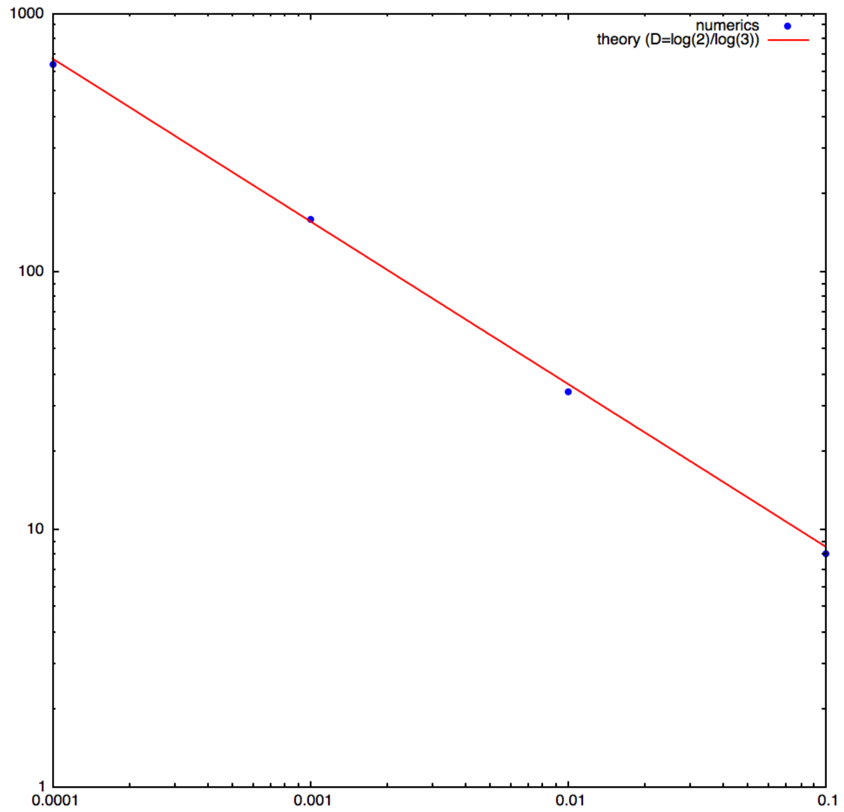
$$D = \lim_{\epsilon \to 0} \frac{\log(2^m)}{\log(1/3)^{-m}} = \frac{\log 2}{\log 3} \sim 0.63$$

Box-counting dimension of Cantor set

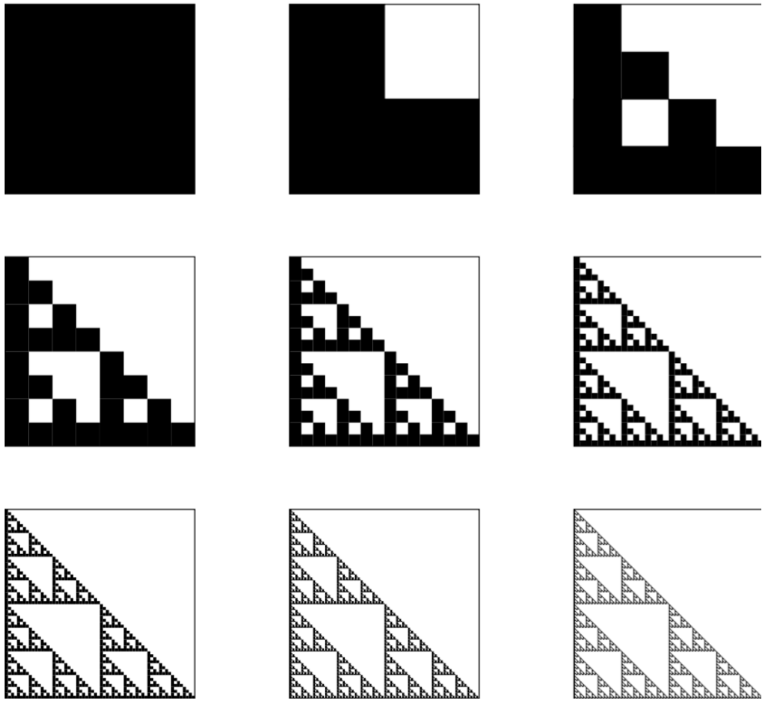
Maxima code

```
array(m,4); for i:1 thru 4 do m[i]:10^i;
array(box,4); for i:1 thru 4 do box[i]:0;
array(flag,m[4]); for i:1 thru m[4] do flag[i]:0;

for j:1 thru 4 do (block(
    for i:1 thru m[j] do flag[i]:0,
    for i:n/2 thru n do flag[entier(x[i]*m[j])+1]:flag[entier(x[i]*m[j])+1]+1,
    for i:1 thru m[j] do if(flag[i]>0) thenbox[j]:box[j]+1)
)$

db:makelist([1/m[i],box[i]],i,1,4)$
dh:log(2)/log(3);
plot2d([[discrete,db],2*x^(-dh)],[x,1/10000,1/10],
[style,[points,1],lines],[legend, "numerics", "theory (D=log(2)/log(3))"],
same_xy,[gnuplot_preamble, "set logscale xy"])$
```

Box-counting dimension of Cantor set



Box-counting dimension of Sierpinski gasket



$$D = \lim_{\epsilon \to 0} \frac{\log N(\epsilon)}{\log(1/\epsilon)}$$

$\epsilon = 1/2, N(\epsilon) = 3$

$\epsilon = (1/2)^2, N(\epsilon) = 3^2$

$\vdots$

$\epsilon = (1/2)^m, N(\epsilon) = 3^m$

$$D = \lim_{\epsilon \to 0} \frac{\log(3^m)}{\log(1/2)^{-m}} = \frac{\log 3}{\log 2} \sim 1.54$$

Box-counting dimension of Sierpinski gasket

Maxima code

```
array(m,3);
for i:1 thru 3 do m[i]:10^i;array(box,3);
for i:1 thru 3 do box[i]:0;
array(flag,m[3],m[3]);
for i:1 thru m[3] do (for j:1 thru m[3] do flag[i,j]:0);

for k:1 thru 3 do block(
    for i:1 thru m[k] do (for j:1 thru m[k] do flag[i,j]:0),
    for i:n/2 thru n do (for j:n/2 thru n do
    flag[entier(x[i]*m[k])+1,entier(y[j]*m[k])+1]:flag[entier(x[i]*m[k])+1,entier(y[j]*m[k])+1]+1),
    for i:1 thru m[k] do if(flag[i,j]>0) then box[k]:box[k]+1
)$

db:makelist([1/m[i],box[i]],i,1,3)$
dh:log(3)/log(2);
plot2d([[discrete,db],2*x^(-dh)],[x,1/1000,1/10],[style,[points,1],lines],[legend, "numerics",
"theory (D=log(3)/log(2))"],same_xy,[gnuplot_preamble, "set logscale xy"])$
```