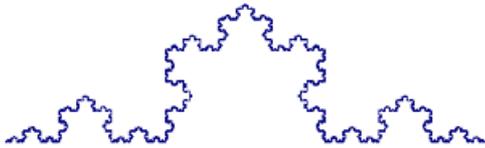


# フラクタル図形

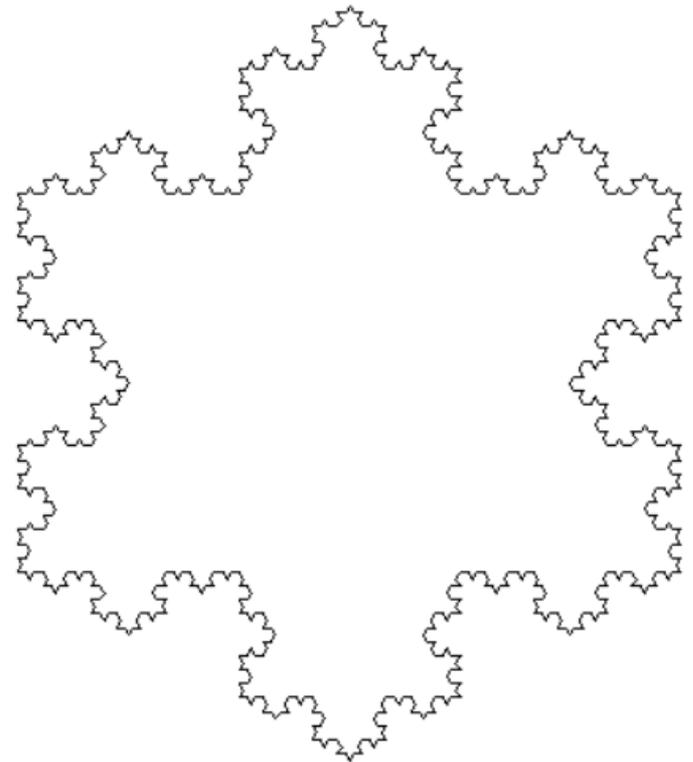
## 統計ソフト「R」使用

```
> koch = function(m=3, x=0, y=0, t=0, L=1) {  
+ if(m == 0) {  
+ lines(c(x, x+L*cos(t)), c(y, y+L*sin(t)))  
+ } else {  
+ koch(m-1, x, y, t, L/3)  
+ koch(m-1, x+(L/3)*cos(t), y+(L/3)*sin(t), t+pi/3, L/3)  
+ koch(m-1, x+(L/3)*(cos(t)+cos(t+pi/3)), y+(L/3)*(sin(t)+sin(t+pi/3)), t-pi/3, L/3)  
+ koch(m-1, x+(2*L/3)*cos(t), y+(2*L/3)*sin(t), t, L/3)  
+ }  
+ }  
> plot(c(0,1), c(-0.3,0.9), type="n", asp=1, axes=FALSE)  
> koch(4,x=0,y=0,t=pi/3)  
> koch(4,x=0.5,y=sqrt(3)/2,t=-pi/3)  
> koch(4,x=1,y=0,t=-pi)  
> |
```

## コッホ曲線



3等分の真中を2等辺三角形にし、  
それを繰り返していく



出典:

<http://www.f.waseda.jp/sakas/R/Rgraphics17.html#fractal>

```
from turtle import *
```

```
def koch(n, length):
```

```
    if n <= 0:
```

```
        fd(length)
```

```
        return
```

```
    koch(n-1, length/3)
```

```
    lt(60)
```

```
    koch(n-1, length/3)
```

```
    rt(120)
```

```
    koch(n-1, length/3)
```

```
    lt(60)
```

```
    koch(n-1, length/3)
```

```
if __name__ == '__main__':
```

```
    n = 4
```

```
    length = 400
```

```
    pu()
```

```
    setpos(-200, 200)
```

```
    pd()
```

```
    speed(0)
```

```
    hideturtle()
```

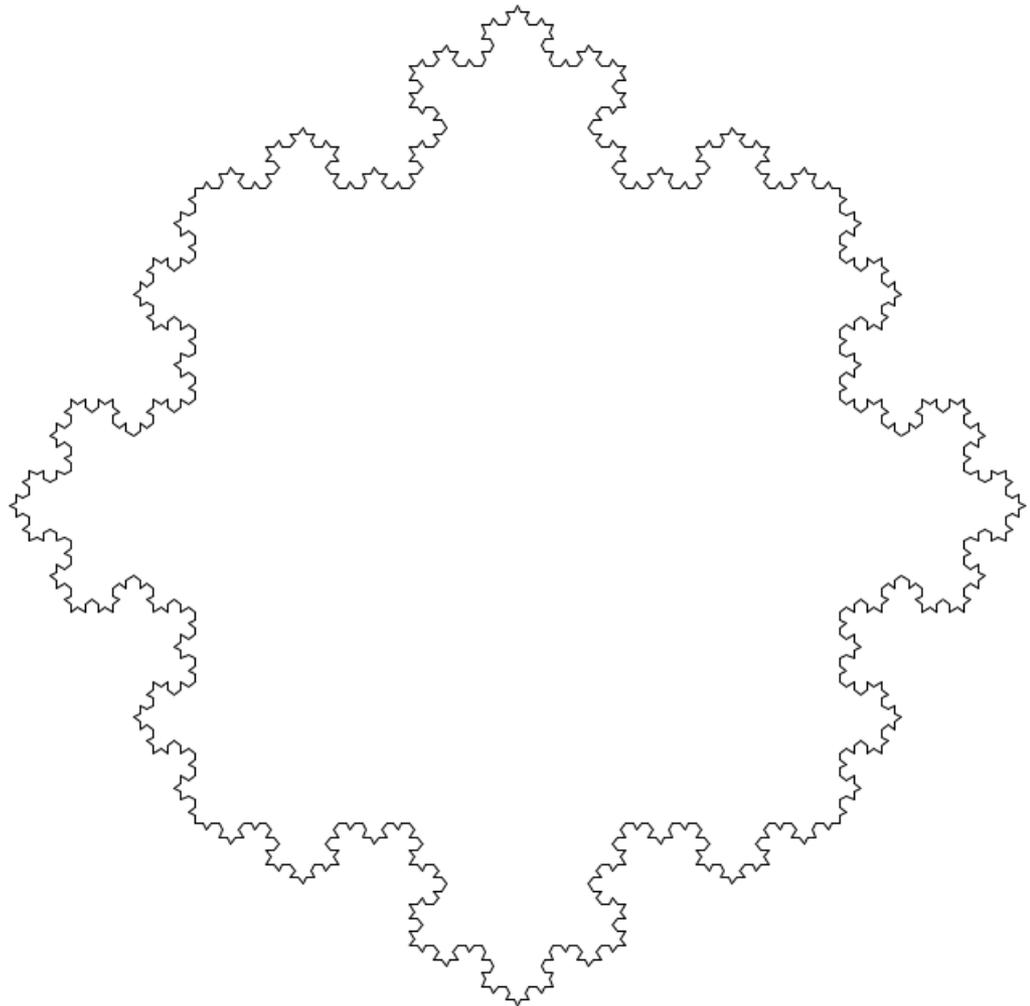
```
    for _ in range(4):
```

```
        koch(n, length)
```

```
        rt(90)
```

```
    done()
```

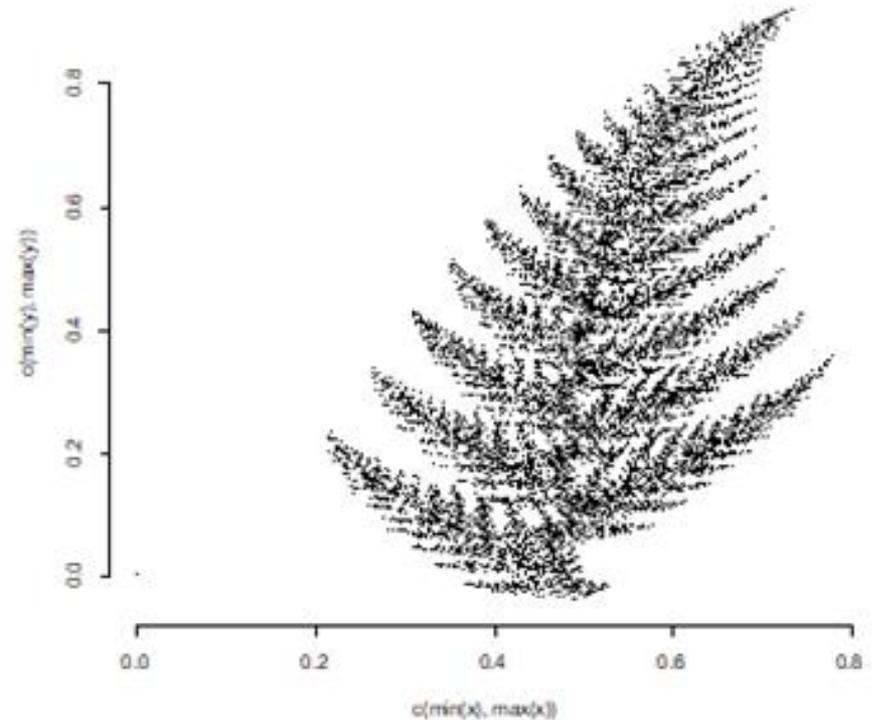
## pythonのturtleで描く



出典:

<https://codom.hatenablog.com/entry/2017/01/20/225952>

```
> fractal = function(n, B, pr) {  
+   x = numeric(n)  
+   y = numeric(n)  
+   m = length(pr)  
+   k = sample(1:m, n, prob=pr, replace=TRUE)  
+   for(i in 2:n) {  
+     x[i] = x[i-1] * B[k[i],1] + y[i-1] * B[k[i],2] + B[k[i],3]  
+     y[i] = x[i-1] * B[k[i],4] + y[i-1] * B[k[i],5] + B[k[i],6]  
+   }  
+   plot(c(min(x),max(x)), c(min(y),max(y)), type="n", bty="n")  
+   points(x, y, pch=".")  
+ }  
> B = matrix(c(0.856, 0.0414, 0.07, -0.0205, 0.858, 0.147,  
+ 0.244, -0.385, 0.393, 0.176, 0.224, -0.102,  
+ -0.144, 0.39, 0.527, 0.181, 0.259, -0.014,  
+ 0, 0, 0.486, 0.355, 0.216, 0.05), 4, 4)  
> pr = c(0.73, 0.13, 0.13, 0.01)  
> n = 50000  
> fractal(n, B, pr)  
> |
```



出典:

<http://www.f.waseda.jp/sakas/R/Rgraphics17.html#fractal>

コマンドプロンプトでpip install Pillow を入れておく

```
import random

N = 20
xm = 0
ym = 0.5
h = 0.6

width = 500
height = 500

W1x = lambda x, y: 0.836 * x + 0.044 * y
W1y = lambda x, y: -0.044 * x + 0.836 * y + 0.169
W2x = lambda x, y: -0.141 * x + 0.302 * y
W2y = lambda x, y: 0.302 * x + 0.141 * y + 0.127
W3x = lambda x, y: 0.141 * x - 0.302 * y
W3y = lambda x, y: 0.302 * x + 0.141 * y + 0.169
W4x = lambda x, y: 0
W4y = lambda x, y: 0.175337 * y

def f(im, k, x, y):
    if 0 < k:
        f(im, k - 1, W1x(x, y), W1y(x, y))
        if random.random() < 0.3:
            f(im, k - 1, W2x(x, y), W2y(x, y))
        if random.random() < 0.3:
            f(im, k - 1, W3x(x, y), W3y(x, y))
        if random.random() < 0.3:
            f(im, k - 1, W4x(x, y), W4y(x, y))
    else:
        s = 490
        im.putpixel((int(x * s + width * 0.5), int(height - y * s)), (0, 128, 0))

if __name__ == '__main__':
    from PIL import Image
    im = Image.new("RGB", (width, height), (255, 255, 255))
    f(im, N, 0, 0)
    im.show()
```

pythonで描く



出典:

<https://qiita.com/noc06140728/items/8b8f06cfc312b8492df4>

# バタフライエフェクト

## pythonで描く

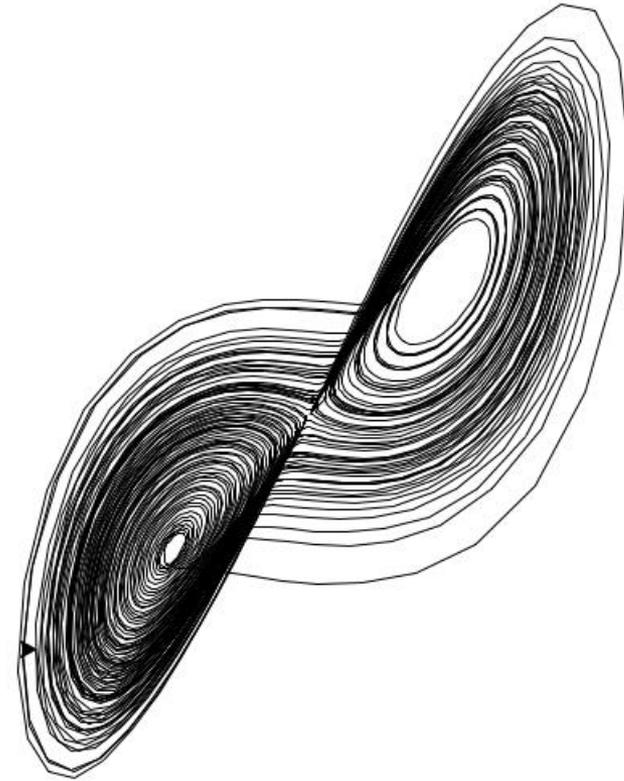
```
from turtle import *
from math import atan2

def lorenz(sigma, beta, rho):
    dt = 0.01
    scale = 10
    x, y, z = (1, 1, 1)
    dx, dy = (0, 0)
    while True:
        setpos(x*scale, y*scale)
        setheading(atan2(dy, dx))

        dx = (sigma*(y-x)) * dt
        dy = (x*(rho-z)-y) * dt

        x += dx
        y += dy
        z += (x*y - beta*z) * dt

if __name__ == '__main__':
    speed(0)
    lorenz(10, 8/3, 28)
```



出典:

<https://codom.hatenablog.com/entry/2017/01/20/225952>