

関数のアクアリウム (ワイエルシュトラウス関数など)

Shimura Masato
JCD02773@nifty.ne.jp

2010年5月20日

目次

1	ワイエルシュトラウス関数	1
1.1	Script	2
1.2	J Grammar	3
2	3D ランダムウォーク	3
2.1	J Grammar	3
2.2	Script	4
3	References	5

概要

ワイエルシュトラウス関数と3D ランダムウォークを生成し描画する。

This article show Weierstraus function and 3D random walk graphics

1 ワイエルシュトラウス関数

Weierstraus の至るところ微分不可能な関数 (1872)

ワイエルシュトラウスの定理

奇数 $a \geq 3$ と、0 と 1 との間にある数 b が、 $ab > 1 + \frac{3\pi}{2}$ を満たしているとき

$$f(x) = \sum_{k=0}^{\infty} b^k \cos(\pi a^k x)$$

によって定義される関数 f は全ての点で連続であり、かつ全ての点で微分可能でない。

$$a = 21$$
$$b = \frac{12}{3} \text{ とする}$$

$$f(x) = \sum_{k=0}^{\infty} \frac{\cos(21^k \pi x)}{3^k} = \cos(\pi x) + \frac{\cos(21\pi x)}{3} + \frac{\cos(441\pi x)}{9} + \dots$$

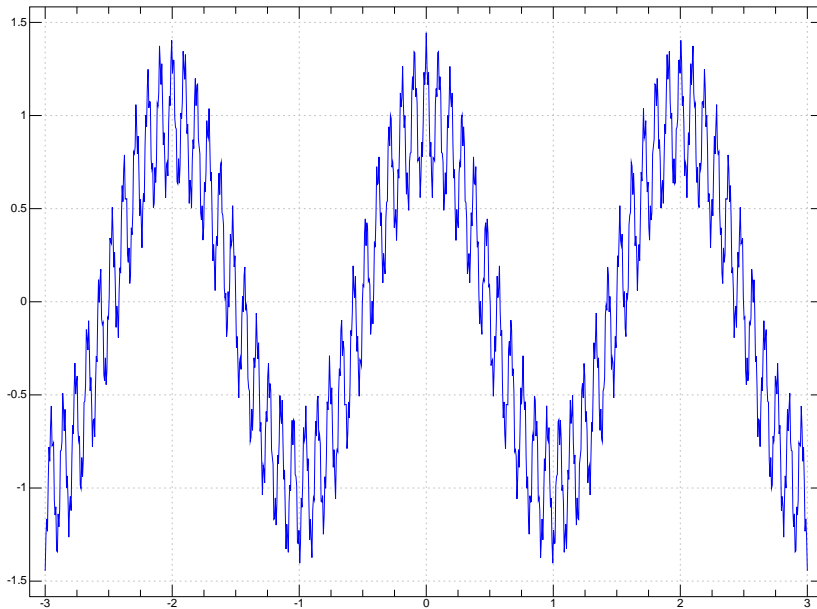


図 1 weierstras

3 項の 1000steps での作図。 *cos* を *sin* に換えてもよく似た図になる。

1.1 Script

```

a=: steps _2 2 1000
wsf=: 3 : 0 NB. 3 項限定
NB. Usage: wsf a // plot a;wsf a
(cos 1p1 * y)+(3%~ cos 21p1 * y)+ 9%~ cos 441p1 * y
)

wsf0=: 4 : 0 NB. 汎用
NB. x is kousuu
NB. 3 wsf0 steps _2 2 1000
+/"1 cos (3^>:i.x)%"1 ~ y*/ 1p1* 21^i.x
)

```

1.2 J Grammar

1p1 π
 21p1 は $21 \times \pi$
 steps numeric に入っている関数
 steps .2 2 1000 は steps from .2 to 2 with 1000
 ~ Refreixive $u y \rightarrow yuy$
 3%~ cos 21p1*y は (cos 21p1 * y)% 3

2 3D ランダムウォーク

- ランダムウォークのスク립ト

```
rw0=: : '+/\_1 1 {~(-: y)> ?~ y' NB. 明示型
rw_t=: [: +/\_1 1 {~ -: > ?~      NB. 関数型
rw_3d=: rw_t;rw_t;rw_t          NB. 3D の演算
```

2.1 J Grammar

乱数の生成 ?~20 は 20?20。
 -: $\frac{1}{2}$. 乱数の大小を生成した乱数の数の $\frac{1}{2}$ で比較して 0 1 の指標にする
 _1 1 {~ 0 1 0 1 の指標を-1 1 に変換する。0 1 { _1 1

 5 > ?~10
 0 0 1 1 1 1 1 0 0 0
 _1 1 {~ (-:10)> ?~10
 _1 _1 1 1 1 1 1 _1 _1 _1

```
<\                    prefix  

                           <\_1 _1 1 1 1 1 (部分)  

                           +---+-----+-----+-----+-----+-----+  

                           |_1|_1 _1|_1 _1 1|_1 _1 1 1|_1 _1 1 1 1|_1 _1 1 1 1 1|  

                           +---+-----+-----+-----+-----+-----+-----+
```

+/\ <\の<を +/に置き換え box の中を合計する
 明示型の関数型へ
 の変換 13 : '+/_1 1 {~(-:y)> y?y'

[: +/\ _1 1 {~ -: > ?~

rw_t=: [: +/\ _1 1 {~ -: > ?~

rw_t;rw_t;rw_t 関数を box で囲った並列演算 (ループ不要)

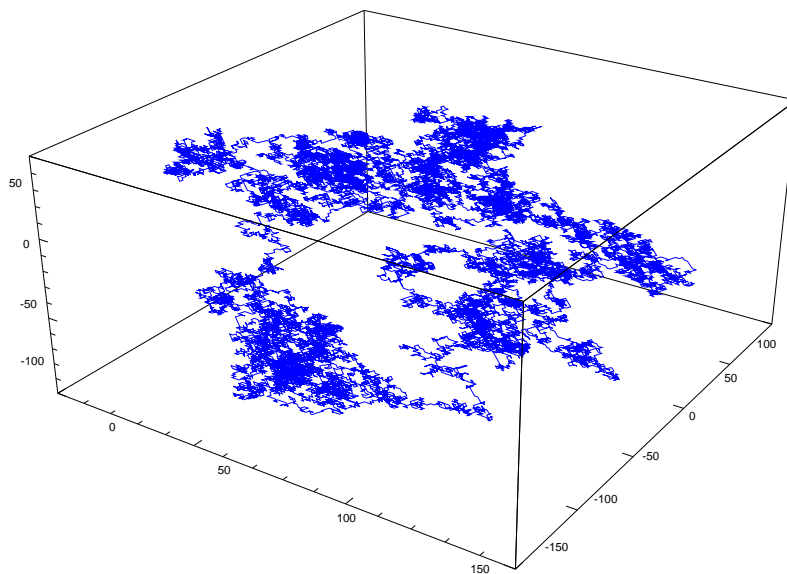


図2 randomwalk-3D

2.2 Script

NB. Usage: rw0 1000

NB. * plot rw_3d 10000

rw0=: 3 : '+/_1 1 {~(-:y)> ?~y'

rw_t=: [: +/\ _1 1 {~ -: > ?~ NB. Tacit

rw_3d=: rw_t;rw_t;rw_t

plot rw_3d 30000 位で何回かトライする。

plot (rw0;rw0) 20000 NB. 2D

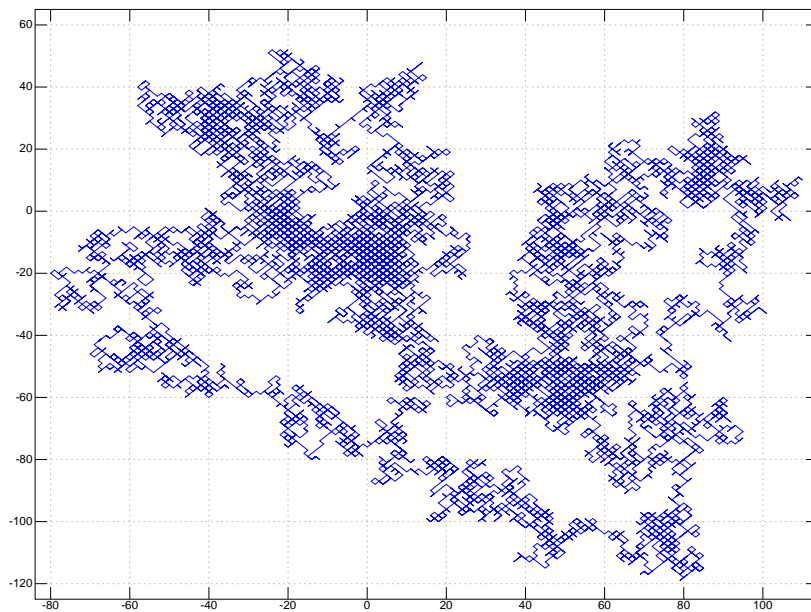


図 3 randomwalk-2D

3 References

William Dunham 一楽 実川訳「微積分名作ギャラリー」日本評論社 2009

Download

J Language <http://www.jsoftware.com> から DL (無料)

Papers and Script

http://homepage3.nifty.com/asagaya_avenue APL_J workshop

<http://www.japla.info> → Library