

# ランダムウォークと樹氷

SHIMURA Masato  
<http://japla.sakura.ne.jp>

2015 年 1 月 23 日

## 目次

1	ランダムウォーク	1
2	樹氷を描く	4
3	樹氷はできたか	7

## 1 ランダムウォーク

ランダムウォークはこの研究会で何回か報告したが、よく整理された教科書の数式からアルゴリズムを得るのは難しく、また新しい報告もあるので概略を記述する。

J803QT 版の Help から NuVoc に入れる。NY J Users Group が最近作った *Vocaburaly* でよく整理されている。

### 1. 乱数

- いつも同じ乱数を打ち出したいときは ?. とピリオドを付ける
- ? 0  $0,1$  間の乱数 (一個のみ)
- ? 2  $(0,1)$  のどちらか。

? 10# 2

0 1 0 0 1 1 0 1 1 1

コインを 10 個投げたもの。長半同数にはならない

- ?~n n ? n に同じ。~はチルドの単項。両項の場合は反転する

```
?~10
```

```
4 8 7 9 2 5 6 0 3 1
```

## 2. ランダムウォークのアルゴリズム

(a) 10 個の乱数 (固定)

```
?~10
```

```
4 3 6 2 9 8 5 1 7 0
```

(b) 中央値 (5) で大小を比較

```
(-:10)> ?~10
```

```
1 1 0 1 0 0 0 1 0 1
```

(c) 上の指標を用いて (0,1) を (-1,1) に変換

```
((-:10)> ?~10){_1 1
```

```
1 1 _1 1 _1 _1 _1 1 _1 1
```

(d)  $J$  の逐次累計関数 *Suffix* \ の単項

```
<\i.5
```

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

```
|0|0 1|0 1 2|0 1 2 3|0 1 2 3 4|
```

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

(e) ランダムウォーク

```
+/\ ((-:10)> ?~10){_1 1
```

```
1 2 1 2 1 0 _1 0 _1 0
```

(f) ランダムウォークのスクリプト。rwt は強引にタシットで記述したもの。下はコイン型で \_1 1 は同数ではないタイプ。ランダムウォークには *itrw0* の ± 同数型と *rw1* のコイン投げの 2 タイプがある。

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

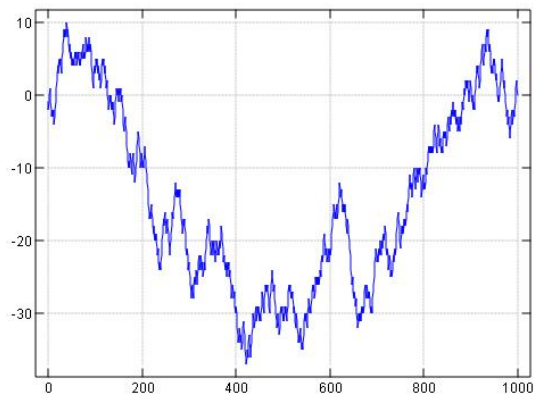
```
rw1=: [: +/\ _1 1{~(?~)]>~ -:&]
```

```
rw1=: 3 : '+/\ _1 1{~ ? y # 2' NB. another definitio
```

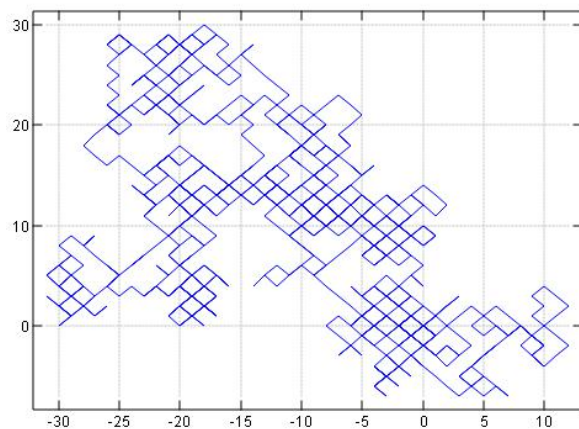
## 3. plot

plot (rw0 1000);rw0 1000 で 2D、plot (rw0 1000);(rw0 1000);rw0 1000 で 3D のランダムウォークを描くことができる。J803QT 版は *jpg* の取り扱いが容易になっており、プロパティーで *jpg* のサイズを求めて記述すれば *TeX* へ組込める

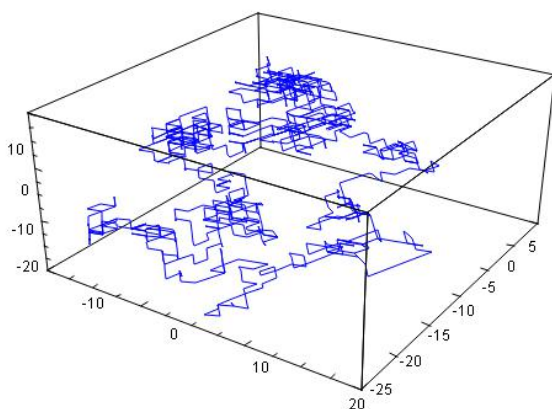
```
require 'plot'  
plot rw0 1000  
pd 'save jpg /temp/rw0.jpg'
```



```
plot (rw0 1000);rw0 1000  
pd 'save jpg /temp/rw1.jpg'
```



```
plot (rw0 1000);(rw0 1000);rw0 1000
```



## 2 樹氷を描く

大きめのスクリーンを用意し、中に 1 点もしくは何がしかの形状の「種」を置き、ランダムウォークで移動する分子がぶつかると種にくっつく。これを繰り返せばどのような図形に成長するかシミュレーションしてみる。

大変時間を要するので、高速型のスクリプトを作成する

### 1. スクリーンの作成

```
mk_screen 10
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
```

```
mk_screen=: 3 : '1 (<2#-:y) } (y,y)$0'
```

大海に一点では遭遇のチャンスが少ないので 10×10 ビットのクロスも用意した

```
mk_screen1=: 3 : ' 1 ({~.(2# 5~--: y)+"1((i.10),.5),(5,.i.10))} (y,y) $ 0'
```

## 2. ランダムウォークの作成

- コイン型、左右対称型のどちらも利用できる。
- スクリーンのビット数作成し、スクリーンをはみ出すものは捨てる

## 3. 座標と照査

### (a) 座標の種類

- $x, y$  座標を用いる
- 複素座標を用いる。`plot` と `viewmat` は複素座標を表示できる

### (b) $x, y$ 座標での照査

- *e. (members of)*
- リストの場合、左の各アイテムが右のリストに含まれているか照査する。  
含まれているかの照査のみでリストの順序は問わない

```
(<1 2) e.(L:0) 0 1;0 2;1 2;3 4;2 1
```

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

```
|1 0|0 1|1 1|0 0|1 1|
```

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

- `=` イコールはリストの順序関係も照査する

```
(<1 2) =(L:0) 0 1;0 2;1 2;3 4;1 1;2 2
```

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

```
|0 0|0 1|1 1|0 0|1 0|0 1|
```

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

### (c) 複素座標での照査

- 複素座標の作成

```
z=. i. 5 2
```

```
0 1
```

```
2 3
```

```
4 5
```

```
6 7
```

```
8 9
```

```
z=. i. 5 2
```

```

({."1 z) j. {"1 z
0j1 2j3 4j5 6j7 8j9

```

- 複素座標の照査

```

0j1 2j3 4j5 6j7 8j9 e. 2j3 6j7
0 1 0 1 0

```

- 複素座標を  $x,y$  座標に戻す

```

+. 0j1 2j3 4j5 6j7 8j9
0 1
2 3
4 5
6 7
8 9

```

(d) 座標は最終的に照査が簡単で計算時間も早い複素座標にした

#### 4. アmend マトリクスデータの書換え

アmendによるマトリクスの書き換えは多彩である。(appendix 参照)

いずれにしても書き換えは「行、列」の順であり、 $(y, x)$  である。ランダムウォークは  $x, y$  とも乱数で打出しており、このままでも差しさわりはないので特に  $x, y$  の入れ替えはしない。

```

100 (<3 3 ) } i. 5 5
0 1 2 3 4
5 6 7 8 9
10 11 12 13 14
15 16 17 100 19
20 21 22 23 24

```

#### 5. シミュレーションに関するメモ

- (a) 氷粒の発生位置は  $x, y$  を乱数で打ち出し、そこからランダムウォークする
- (b) 一回のランダムウォークに数分子入っており、一度種にぶつかってもまだ氷滴が浮遊する
- (c) 種にぶつかったポイントの直前ポイントを取り出し、ビットを立てることでくっつきを表す
- (d) ~. チルド 重複排除

接触の判定は図形の外枠を取り出さず、種に着いた数全体対象に数的に行い、重複は落とす

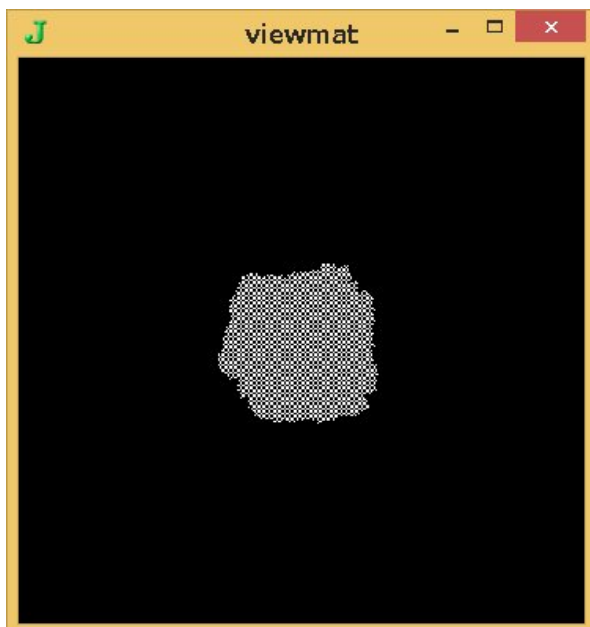
```
~. Nub 重複排除
~. 1 2 3 4 3 4 5
1 2 3 4 5
```

### 3 樹氷はできたか

複素座標を用いて、スクリプトを簡素化したが、 $500 \times 500$  のスクリーンの中央に 1 ビットの種を置いた場合、わずか 500 回の反復でも 3 分を要する。(コア i5 2Mhz)

```
ts=. 6!:2,7!:2@] NB. time space
ts 'a=. icetree1 500;500'
182.436 3.37649e7 NB. 182sec. 33.7M
```

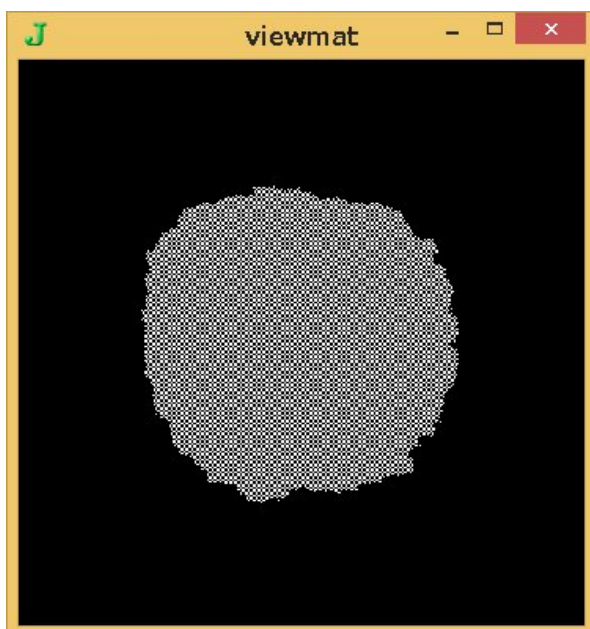
```
+./+ a
7885 NB. contact 7885 ice drops
viewmat a
```



\*1

ループを 1000 回に増やすと 20 分近くかかる。図は細胞増殖のように広がる

```
ts 'a=. icetree1 500;1000'
1092.73 3.25238e7
+ /+ / a
31092
viewmat a
```



処理に時間がかかり、細胞増殖状になる点に関して、西川利男氏から乱数を短くして、点を多くしては如何かとアドバイスをいただいた。スクリプトを改良した。

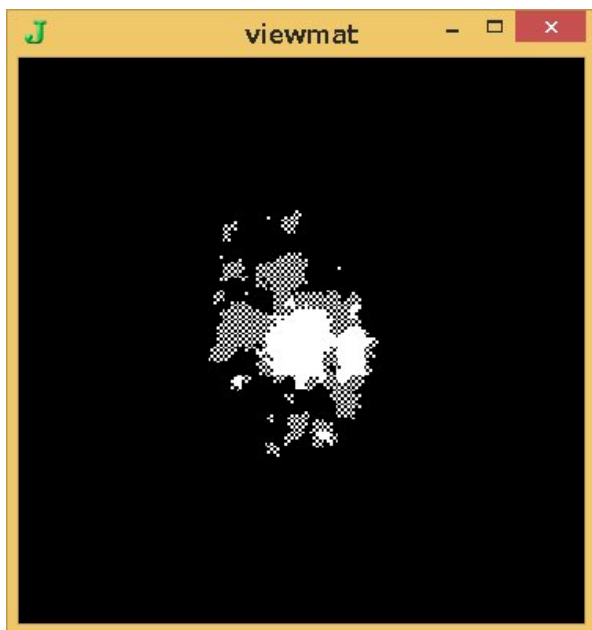
また、大海に一点では遭遇のチャンスが少ないので 10×10 ビットのクロスも用意した

```
mk_screen1=: 3 : ' 1 ({~.(2# 5~--: y)+"1((i.10),.5),(5,.i.10))} (y,y) $ 0'
+ /+ / a1=. icetree1 200;6000
1386
viewmat a1
```

---

\*1 take jpeg size, show propaty 詳細





1. マトリクスのアメンドの指標は  $x, y$  ではなく行; 列である。  $(y, x)$  となる。  $J6, J8$  共同

```

      1(5 5;6 5;7 6)} 10 10 $ 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```

2. 2重ボックス → 2を5列目、3を7列目に。行は2, 4行

```

      2 3 (<2 4 ;5 7)}10 10 $ 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 0 3 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 0 3 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0

```