

# ランダムウォークと離散型擬似乱数

樋口さぶろお

龍谷大学工学部数理情報学科

計算科学☆実習 B L01(2016-04-11 Mon)

最終更新: Time-stamp: "2016-05-23 Mon 10:25 JST hig"

## 今日の目標

- ランダムウォークとは何か説明できる
- C の `srand(unsigned)`, `rand()` 関数の働きを説明できる
- C で離散型擬似乱数を生成できる



<http://hig3.net>

## ここまで来たよ

### 3 はじめに

- この授業どんなのり?

### 4 ランダムウォークと離散型擬似乱数

- ランダムウォーク
- 擬似乱数
- 離散型擬似乱数の正しい/間違ったプログラム

## 科目の目標

もう少し正確にはシラバスを見てね.

- 現象の確率モデルとは何か, 確率過程とは何か, 例をあげて説明できる.
- 確率モデルをオイラー表示とラグランジュ表示で表現し, 量を計算することができる.
- 確率モデルのシミュレーションのプログラムを作成し, その実行結果から, 表計算ソフトウェア・統計ソフトウェアを用いて統計的推定・検定を行うことができる
- ∴
- チームで協力して問題を解決できる, 効率よく質問できる, 自分の学習方法を改善できる

## どんな人のための科目?

### 計算科学☆実習 B を履修した方がいい人

- 確率過程 (=時間に依存する確率的現象) を知りたい人
- 微分方程式 (決定論的モデル) の見ていない, 残り半分の世界を確率論的モデルで見たい人
- モデル駆動の研究の見ていない, データ駆動の研究の世界を見たい人
- 偶然性のあるゲームを仕組みからわかって作りたい人
- 確率を, プログラム作成の中で実感したい人
- ランダムアルゴリズムが使えるようになりたい人
- コンピュータでデータの解析ができるようになりたい人

### 計算科学☆実習 B を履修しない方がいい人

- (単位をとっているかどうかに関わらず) 確率統計☆演習 I, 数値計算法及び実習がぜんぜんわかってない感がある人, この機会にわかろうという決意のない人

## 科目ののり

注文が多くめんどくさい科目です…

成績計算 科目の成績 100 ピーナッツは

- 25 ピーナッツ:平常点. 毎回授業での quiz, 授業時間外の予習復習.
  - ▶ だいたい 10 講義の Quiz ほか
  - ▶ だいたい 15 実習時間内の課題提出 TA の現場チェックでなく教員の提出プログラムチェック. TA は間違いの発見に努めますが, 「それで OK」とは言いません.
- 50 ピーナッツ:プチテスト群
  - ▶ 15 紙の非参照プチテスト
  - ▶ **35=7+14+14** プログラミング実技の非参照プチテスト
- 25 ピーナッツ:定期試験期間内の紙のファイナルトライアル (外部記憶あり)
- その他追加ピーナッツ. その時に説明.

ファイナルトライアル時点で 35 ピーナッツ未満の人は, 本試験は (平均点を上げるために) 参加をおすすめしますが, 追試験は実施しません.

欠席届ピーナッツ的に考慮されたい場合は、専用用紙に事情を説明する書類を貼って、授業前後各5分に提出(事前事後とも可。ファイナルトライアルが締切)。欠席に事前連絡は不要。何回欠席しても期末試験受験資格を失うことはありません。

資料授業で配布。授業後に欲しい人は<http://hig3.net> から各自ダウンロード。1-503 前のレターボックスに残ってることも。

## 担当者ののり

- なまえ: 樋口さぶろお [hig-compsci@math.ryukoku.ac.jp](mailto:hig-compsci@math.ryukoku.ac.jp)
- へや: 1-502
- オフィスアワー: 月昼(1-502), 木6(1-502/1-539)。訪問歓迎な時間: 月木金昼(1-502, Math ラウンジに行ってることも)。お弁当持参歓迎。お湯あげます。
- Web ページ: <http://hig3.net> 実習の指示や、スケジュールもここから。

## 科目の1週間のタイムライン

- ① 月昼 樋口オフィスアワー (1-502)
- ② 月 15:20 予習復習問題 (e ラーニング) 解答 1 回のみ
- ③ 月 4 講義 (7-002), quiz(参照あり)
- ④ このころ実習のタスク公開
- ⑤ 火 23:55 先週の課題の一部の提出締切
- ⑥ 水 13:35 予習復習問題 (e ラーニング) 解答何回でも
- ⑦ 水 3 実習 (1-609), quiz 返却
- ⑧ 水 23:55 今週の課題の一部の提出締切

実習室に行ったら, <http://hig3.net> → 計算科学☆実習 B へ.

## ここまで来たよ

### 3 はじめに

- この授業どんなのり?

### 4 ランダムウォークと離散型擬似乱数

- ランダムウォーク
- 擬似乱数
- 離散型擬似乱数の正しい/間違ったプログラム



## C 言語で数列の計算

数値計算法

数列  $\{X(t)\}$ , 時刻  $t = 0, 1, 2, \dots$

初項  $X(0) = a$ , 漸化式  $X(t+1) = X(t) + R(t+1)$ .

階差数列  $R(t+1) = \text{定数}$  なら  $X(t)$  は等差数列.

C 言語で数列を書く?

```
int x;
int r;
int t;

t=0;
x=初項;

printf("%d\n", t, x);
for (t=0; t < 100; t++){
    r=(階差数列の一般項 R(t+1));
    x=x+r; /* X(t+1) を求めた */
    printf("%d,%d\n", t+1, x);
}
```

## ランダムウォーク (確率過程の例)

ランダムウォーク  $\Leftrightarrow$  階差数列  $R(t+1)$  が

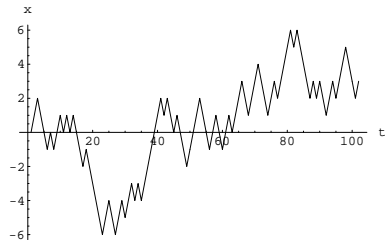
現象の数学 A, 確率統計 II

つまり  $R(t+1)$  がランダム. 例えば, こんな場合.

$R(t+1)$	確率
+1	$p$
-1	$q(=1-p)$

ランダムウォークってどんなところに出てくる?

等差数列 vs ランダムウォーク



- 株価変動

- 

- 

- モンテカルロ数値積分＝ランダムアルゴリズムの典型

## ここまで来たよ

### 3 はじめに

- この授業どんなのり?

### 4 ランダムウォークと離散型擬似乱数

- ランダムウォーク
- 擬似乱数
- 離散型擬似乱数の正しい/間違ったプログラム

## 離散型擬似乱数列の生成

### モンテカル口法

確率的/決定的な量を計算するのに、確率変数の標本抽出を実際にコンピュータで乱数 (random number) を使って行う方法

乱数列 = ある確率変数の標本になってる数列 = ランダムな数列  
 $R(t+1)$  を C 言語でどう書く?

```
#include <stdlib.h>
```

```
/* 0以上 RAND_MAX 以下の正の整数をランダムに選んで返す関数 */  
int rand();  
/* その初期化 */  
void srand(unsigned seed);
```

RAND\_MAX は M\_PI みたいな定数. 値は処理系依存. 例えば  $2^{31} - 1$ .  
今の目的としては、得られる値は、+1, -1 だけでいいんだけどな～

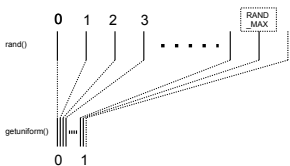
～→

この授業の約束 (+世の中の習慣). `rand()` を生で使わず, いったん  $[0, 1)$  一様乱数にして使う.  $\rightarrow$  `double getuniform()`

```
/* [0,1) 一様乱数 */
double getuniform(){
    return rand()/(1.0+RAND_MAX);
}
```

## `getuniform()` の性質

- ‘値域’ は  $[0, 1)$ .  $0 \leq \text{getuniform}() < 1$ .
- $(\text{getuniform}() < r \text{ となる確率}) = r$ . ( $0 \leq r \leq 1$ )



連続型確率変数みたいなもの.

$$\text{確率密度関数 } f(x) = \begin{cases} 1 & (0 \leq x < 1) \\ 0 & (\text{他}) \end{cases}.$$

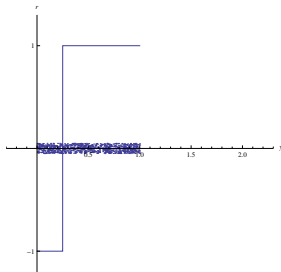
## 計算機の頭の中どうなってるの？

擬似乱数列 = 'ほぼ' ランダムな数列

ある確率で  $\pm 1$  を返したい!

```
/* 引数  $y$  が  $[0, 1)$  一様乱数なら, getrandom の返り値は  
   確率  $1/4$  で  $-1$ , 確率  $3/4$  で  $+1$ */
```

```
int getrandom(double y){  
    if( y < 0.25){  
        return -1;  
    } else {  
        return +1;  
    }  
}
```



```
r=getrandom(getuniform());
```

## ここまで来たよ

### 3 はじめに

- この授業どんなのり?

### 4 ランダムウォークと離散型擬似乱数

- ランダムウォーク
- 擬似乱数
- 離散型擬似乱数の正しい/間違っただプログラム



## ソースコード 1: 乱数

```
1 /*
2 randl.c --- -1 or +1 を確率1/4, 3/4で選ぶ乱数
3 Time-stamp: "2013-04-09 Tue 18:57 JST hig"
4 */
5 #define _CRT_SECURE_NO_WARNINGS // VC++2008用おまじない
6 #include <stdio.h>
7 #include <stdlib.h> /* srand(), rand() を使うのに必要 */
8
9 /* 関数プロトタイプ宣言 */
10 double getuniform();
11 int getrandom(double y);
12
13 int main(){
14     int seed; /* 疑似乱数のシード */
15     int t; /* カウンタ */
16     int tmax=100; /* 疑似乱数を得る回数 */
17
18     scanf("%d",&seed);
19     srand(seed); /* シードの設定 */
20     for(t=0;t<tmax;t++){
21         /* srand(seed); */ /*ここに置くと? */
22         printf("%f\n",getrandom(getuniform()));
23     }
24     return 0;
25 }
26
27 /** [0,1) 一様疑似乱数を返す */
28 double getuniform(){
29     return rand()/(RAND_MAX+1.0);
30 }
31
32 /** -1 or +1 を確率1/4, 3/4 で返す乱数 */
33 int getrandom(double y){
34     if( y < 0.25 ){
35         return -1;
36     } else {
37         return +1;
38     }
39 }
```

## L01-Q1

## Quiz(擬似乱数の使いかた)

引数  $y$  として  $[0, 1)$  一様乱数が与えられたとき, 下の確率で値を返す `double getrandom(double y)` を, サンプルプログラムを参考に書こう.

返り値	確率
0.6	0.7
0.4	0.3

## L01-Q2

## Quiz(rand() の振る舞い)

次のプログラムで、A が出力される確率は?

```
if( getuniform()  
    ==getuniform() ){  
    printf("A\n");  
}
```

- ① 0
- ② 0 に近い
- ③ 1/2
- ④ 1/2 くらい
- ⑤ 1 に近い
- ⑥ 1

## L01-Q3

## Quiz(rand() の振る舞い)

次のプログラムで、A が出力される確率は?

```
1 if( getuniform() < 0.1 ){  
2     if( getuniform() < 0.2 ){  
3         printf("A\n");  
4     }  
5 }
```

- ① 0
- ② 0.02
- ③ 0.1
- ④ 0.2
- ⑤ 0.3
- ⑥ 1

## L01-Q4

## Quiz(疑似乱数の使いかた)

引数  $y$  として  $[0, 1)$  一様乱数が与えられたとき, 下の確率で値を返す `int getrandom(double y)` を, サンプルプログラムを参考に書こう.

返り値	確率
-1	1/3
0	1/2
+1	1/6

Hint: `getrandom()` のグラフは, なぜ, どのような形になるべき?

予習復習問題 次回の実習までの間には, まだ予習復習問題はありません.

<https://manaba.ryukoku.ac.jp>

マイページの下の方に manaba 出席カード  
提出

今日は「匿名で提出する」にチェック. 自由  
記述に学籍番号書いてね.

今週水の実習にはイヤフォン持参 (必須)

