乱数とMonteCarlo法

モデリングとシミュレーション特論

2019年度

只木進一

様々な乱数生成 Random number generators

- 組み込みの一様乱数Math.random()から様々な分布の乱数を生成したい
- AbstractRandomクラス
  - 次の乱数を生成getNext()
- 変換法
  - Transformクラス
- 棄却法
  - Rejectionクラス

変換法
Transformation method

- 確率密度(probability density) $f(x)(a \le x < b)$に対する分布

$$F(x) = \int_a^x f(z)\mathrm{dz}$$

- 確率分布(probability distribution) $F(x)$の逆関数が得られる場合
  - 例：指数分布

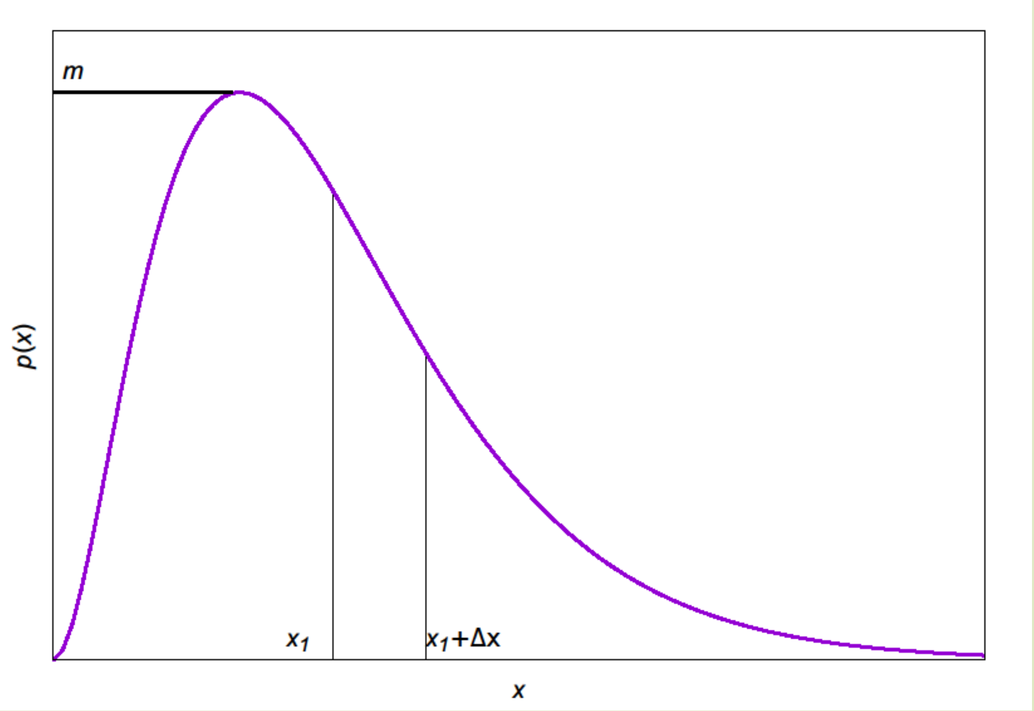
$$f(x) = Ae^{-x},\ 0 \le x < 1,\ A = \frac{e}{e-1}$$

$$F(x) = \int_0^x f(z)\mathrm{dz} = A\left(1-e^{-x}\right)$$

$$F^{-1}(r) = -\ln\left(1-\frac{r}{A}\right)$$

棄却法
Rejection method

- bin$[x_1, x_1 + \Delta x)$に入る確率は、その面積に比例することを利用



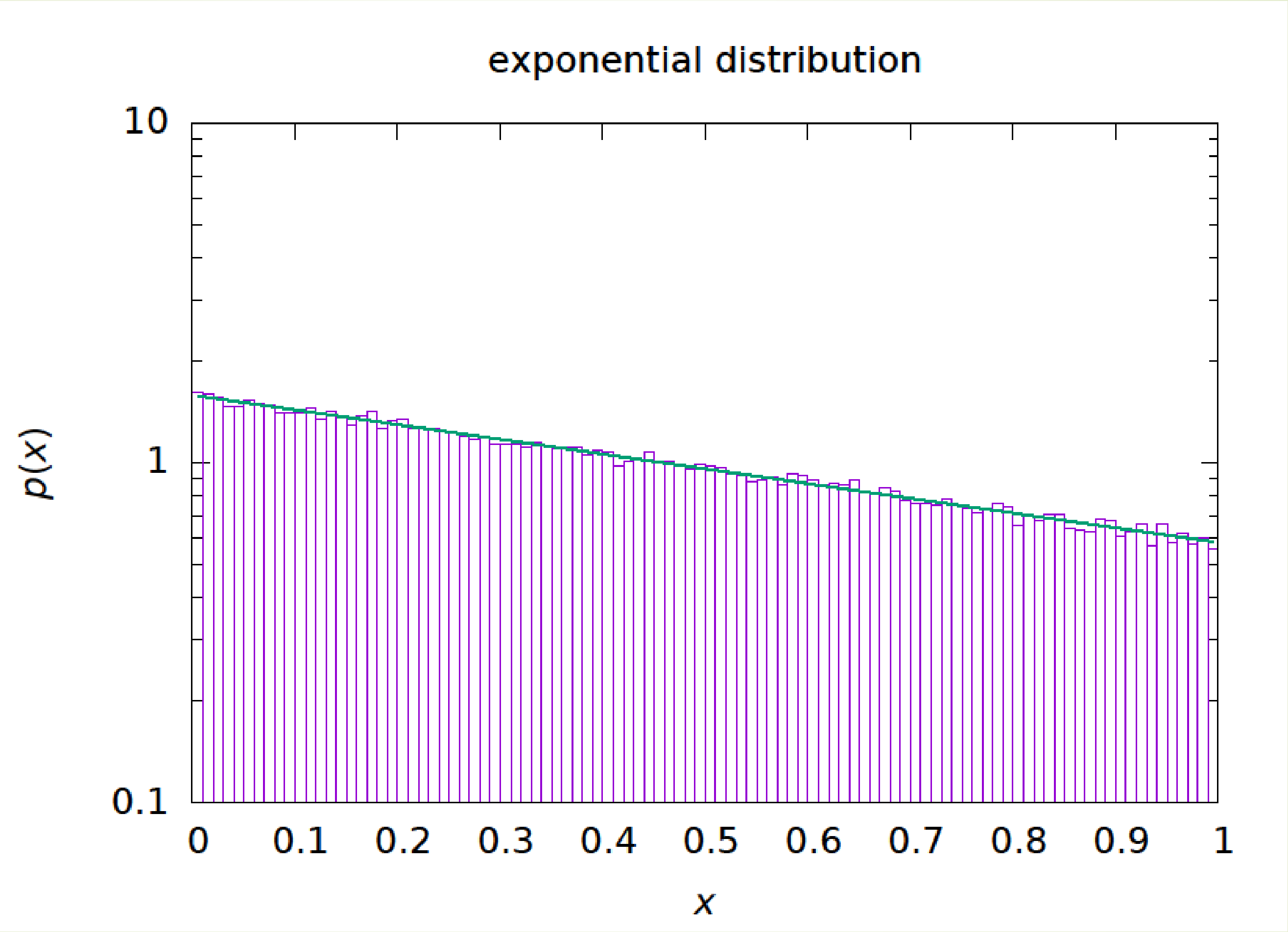
乱数生成クラスの関係

- randomNumbers/AbstractRandom.java
- randomNumbers/Transform.java
- randomNumbers/Rejection.java
- java.util.function.DoubleFunction<R>の利用

乱数生成の例：指数分布
Example : Exponential distribution

- randomNumbers/Exp.java

```
//指数分布に対応した分布関数の逆関数を定義
// A * exp (-x)
double A = Math.E / (Math.E - 1);
DoubleFunction<Double> invProDist = (x) -> {
   return -Math.log(1 - x / A);
};
//変換法による乱数生成のインスタンス
AbstractRandom aRandom = new Transform(invProDist);
```


exponential distribution
10
1
0.1
p(x)
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
x

確率と大数の法則

Probability and Law of Large Numbers

- 例えば、公正なサイコロを振った場合に、1の目がでる確率が1/6とはどういう意味か？
  - 十分な回数の試行を行うと、1が出る相対頻度が1/6に近づく
  - 大数の法則
  - もう少し詳しくみていく

- 平均μ、分散σ^2である確率変数X
- n個の標本平均

$$\bar{X} = \frac{1}{n}\sum_{k=0}^{n-1} X_k$$

- $\bar{X}$を元の分布で平均、分散を計算
  - 十分多数の標本を生成した場合に相当

- 平均は、元の分布の平均と等しい

$$E\left(\bar{X}\right)=\frac{1}{n}E\left(\sum X_i\right)=\frac{1}{n}\sum E\left(X_i\right)=\frac{1}{n}n\mu=\mu$$

- 分散はn^{-1}で小さくなる

$$V\left(\bar{X}\right)=E\left(\left(\bar{X}-\mu\right)^2\right)=E\left(\frac{1}{n^2}\left(\sum_i (X_i-\mu)\right)^2\right)$$

$$=\frac{1}{n^2}E\left(\sum_i (X_i-\mu)^2+\sum_{i\neq j}(X_i-\mu)\left(X_j-\mu\right)\right)$$

$$=\frac{1}{n^2}E\left(\sum_i (X_i-\mu)^2\right)+\frac{1}{n^2}E\left(\sum_{i\neq j}(X_i-\mu)\left(X_j-\mu\right)\right)=\frac{1}{n^2}n\sigma^2=\frac{\sigma^2}{n}$$

大数の法則をシミュレーションで確かめる

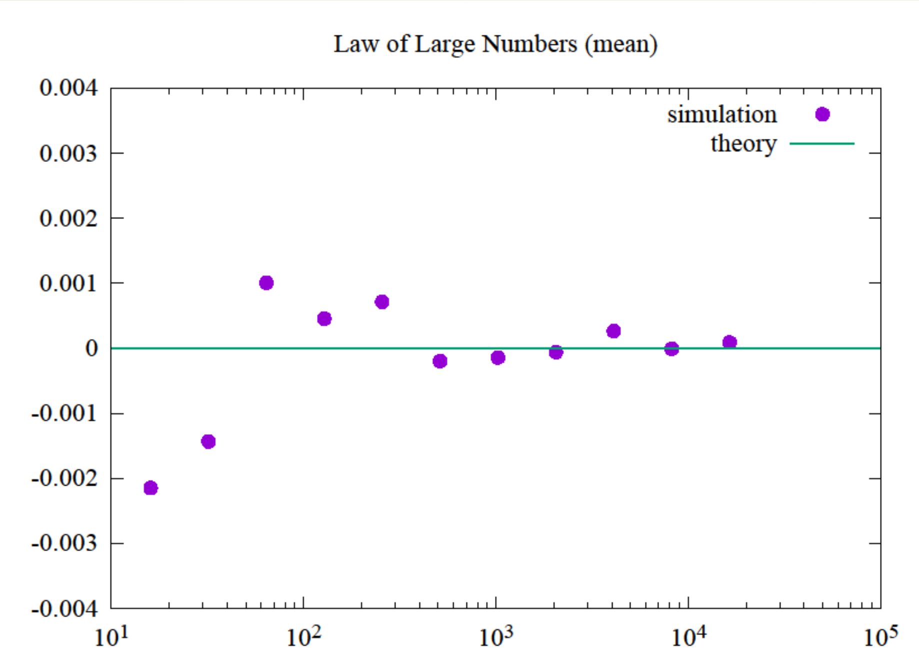
- 大きさnの標本を生成する
- 元の分布での平均はできないため、同じ大きさの標本を多数(m 個)生成して、平均
- nを変化させて、平均と分散を計測

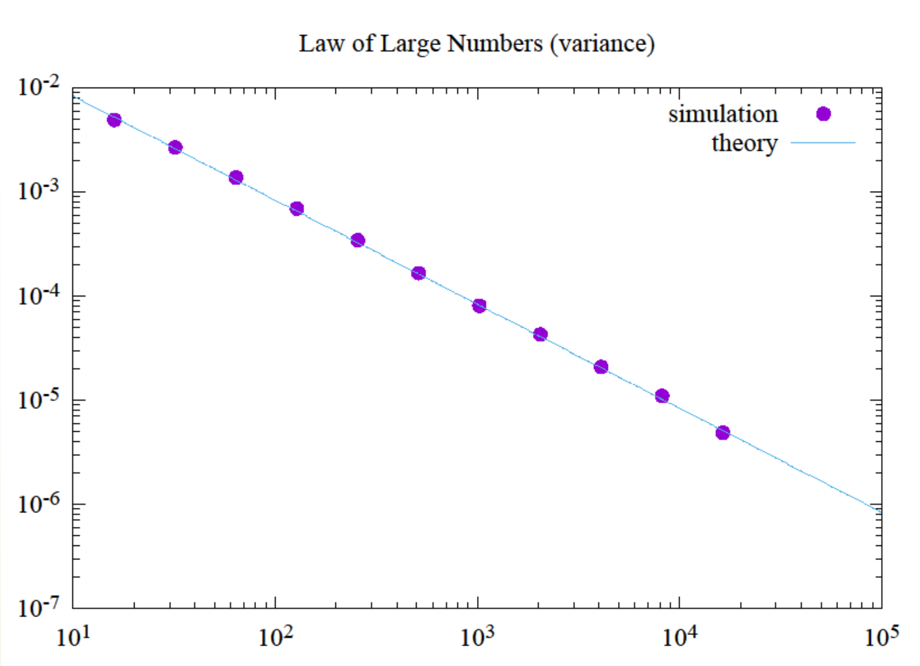
簡単な例：一様分布
Example： uniform

$$f(x)=\begin{cases}1 & -\frac{1}{2}\le x<\frac{1}{2}\\ 0 & \text{otherwise}\end{cases}$$

$$\langle x\rangle=\int_{-1/2}^{1/2}xf(x)\mathrm{d}x=\int_{-1/2}^{1/2}x\mathrm{d}x=\left[\frac{1}{2}x^2\right]_{-1/2}^{1/2}=0$$

$$\langle x^2\rangle=\int_{-1/2}^{1/2}x^2f(x)\mathrm{d}x=\int_{-1/2}^{1/2}x^2\mathrm{d}x=\left[\frac{1}{3}x^3\right]_{-1/2}^{1/2}=\frac{1}{3}\left(\frac{1}{8}+\frac{1}{8}\right)=\frac{1}{12}$$


Law of Large Numbers (mean)
simulation
theory
0.004
0.003
0.002
0.001
0
-0.001
-0.002
-0.003
-0.004
10^1
10^2
10^3
10^4
10^5


Law of Large Numbers (variance)
simulation
theory
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^1
10^2
10^3
10^4
10^5

疑似乱数 Pseudo random numbers

- コンピュータ内で乱数を発生させる
- なんらかのアルゴリズム(algorithm)が必要
  - つまり、相関(出現した数値と次の数値の関係)がある
- 同じ乱数を何度も発生できる

線形合同法
Linear Congruential Method

- 漸化式:$i_n = (a i_{n-1} + c) \bmod m$
  - 定数の選び方で性質が大きく異なる。
  - 良い定数は経験的に知られている。
- C/C++は符号なし整数が使える
  - Overflow制御が不要
- FORTRANのような符号なし整数がない言語
  - Overflowが起きないような工夫が必要

- 32ビット符号なし演算の場合のパラメタ例

$$a = 2416,\ c = 374441,\ m = 1771875$$

- 32ビット符号あり演算の場合のパラメタ例

$$a = 9301,\ c = 49297,\ m = 233280$$

Schrageの方法

- 32ビット符号あり演算
  - オーバーフローを避けて$m = 2^{31} - 1$とする
- 与えられたmに対して

$$(a = 17807,\ c = 0,\ m = 2^{31} - 1)$$

$$q = \lfloor m/a \rfloor,\ r = m \bmod a$$

$$m = aq + r$$

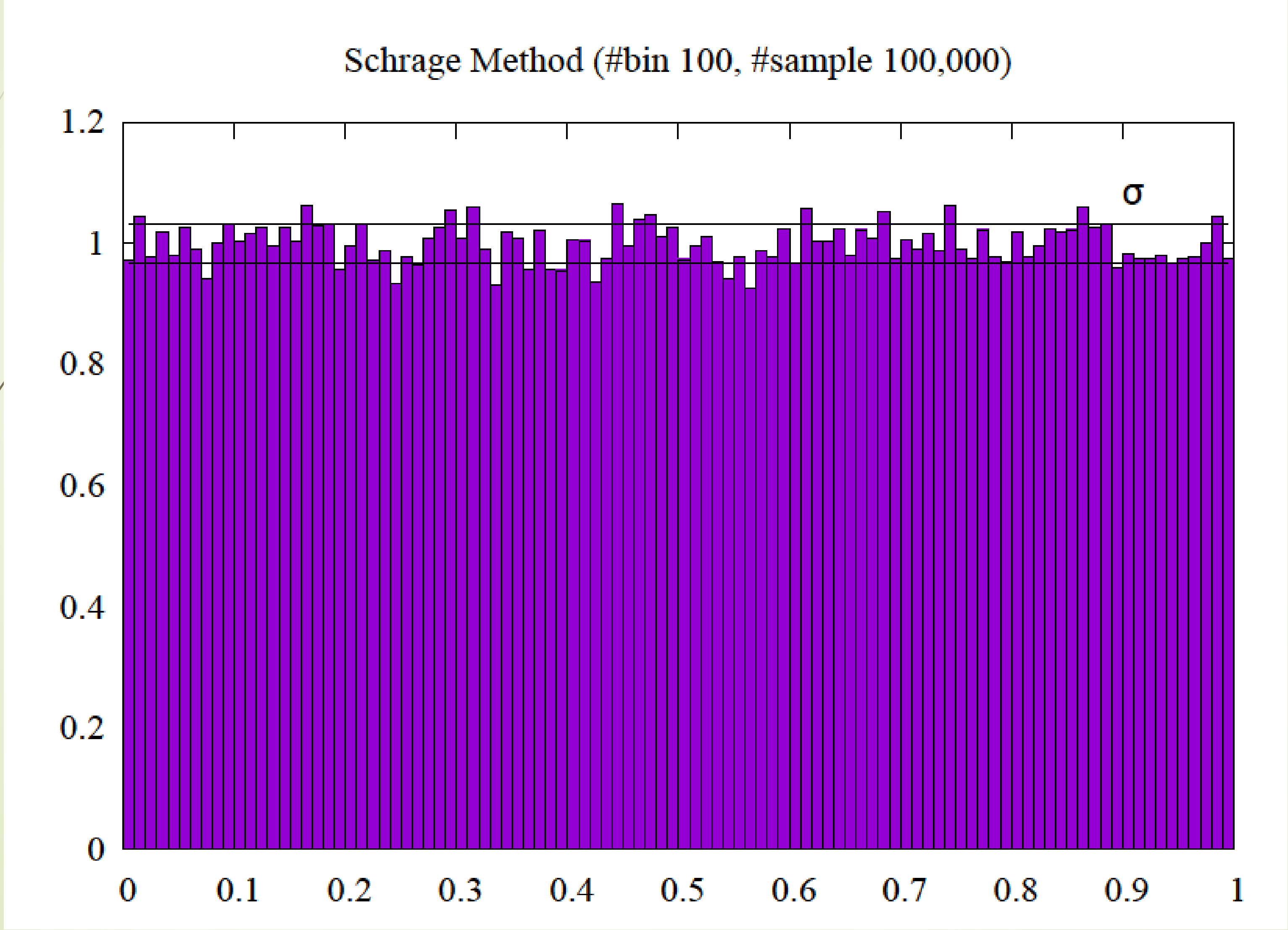
- $r < q$の条件が必要

$$ai_n \bmod m = \begin{cases} a(i_n \bmod q) - \lfloor i_n / q \rfloor r & \text{if not negative} \\ a(i_n \bmod q) - \lfloor i_n / q \rfloor r + m & \text{otherwise} \end{cases}$$

- なぜなら$i_n = xq + y$とすると
  - 右辺：$ay - xr$
  - 一方

$$ai_n = a(xq + y) = xaq + ay = x(m - r) + ay$$

$$= xm + ay - xr$$


Schrage Method (#bin 100, #sample 100,000)
σ
1.2
1
0.8
0.6
0.4
0.2
0
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

線形合同法の問題点 problems in LCM

- ある数aが発生すると次が一意に決まっている。
- 周期がm
  - ある種のシミュレーションでは不足
- 多次元疎結晶構造
  - 連続するn個の乱数を一つのn次元空間の座標とすると、パラメタによっては、結晶構造が見える

Monte Carlo法

- 狭義
  - 乱数を用いた積分（和）計算
- 広義
  - 乱数を用いたアルゴリズム・シミュレーション技法

Monte Carlo法の例：πの計算 Example : estimation π

- 一辺の長さ1の正方形内に2次元乱数を生成：(x, y) $(0 \le x, y < 1)$
- 乱数が半径1の扇形に入る $\left(0 \le (x^2 + y^2)^{\frac{1}{2}} < 1\right)$ 確率
  - 正方形に対する扇形の面積の比$\pi/4$

- N個の2次元乱数のうち、m個が扇形に入る確率$P_N(m)$は二項分布

$$P_N(m) = \binom{N}{m} p^m (1-p)^{N-m}, \ p = \frac{\pi}{4}$$

- 確率母関数を使うと平均等が計算できる

$$G(z) = \sum_{m=0}^{N} P_N(m) z^m = (1-p+pz)^N$$

$$G'(z) = \sum_{m=0}^{N} m P_N(m) z^{m-1} = Np(1 - p + pz)^{N-1}$$

$$\langle m \rangle = G'(1) = Np$$

$$G''(z) = \sum_{m=0}^{N} m(m-1) P_N(m) z^{m-2}$$

$$= N(N-1) p^2 (1 - p + pz)^{N-2}$$

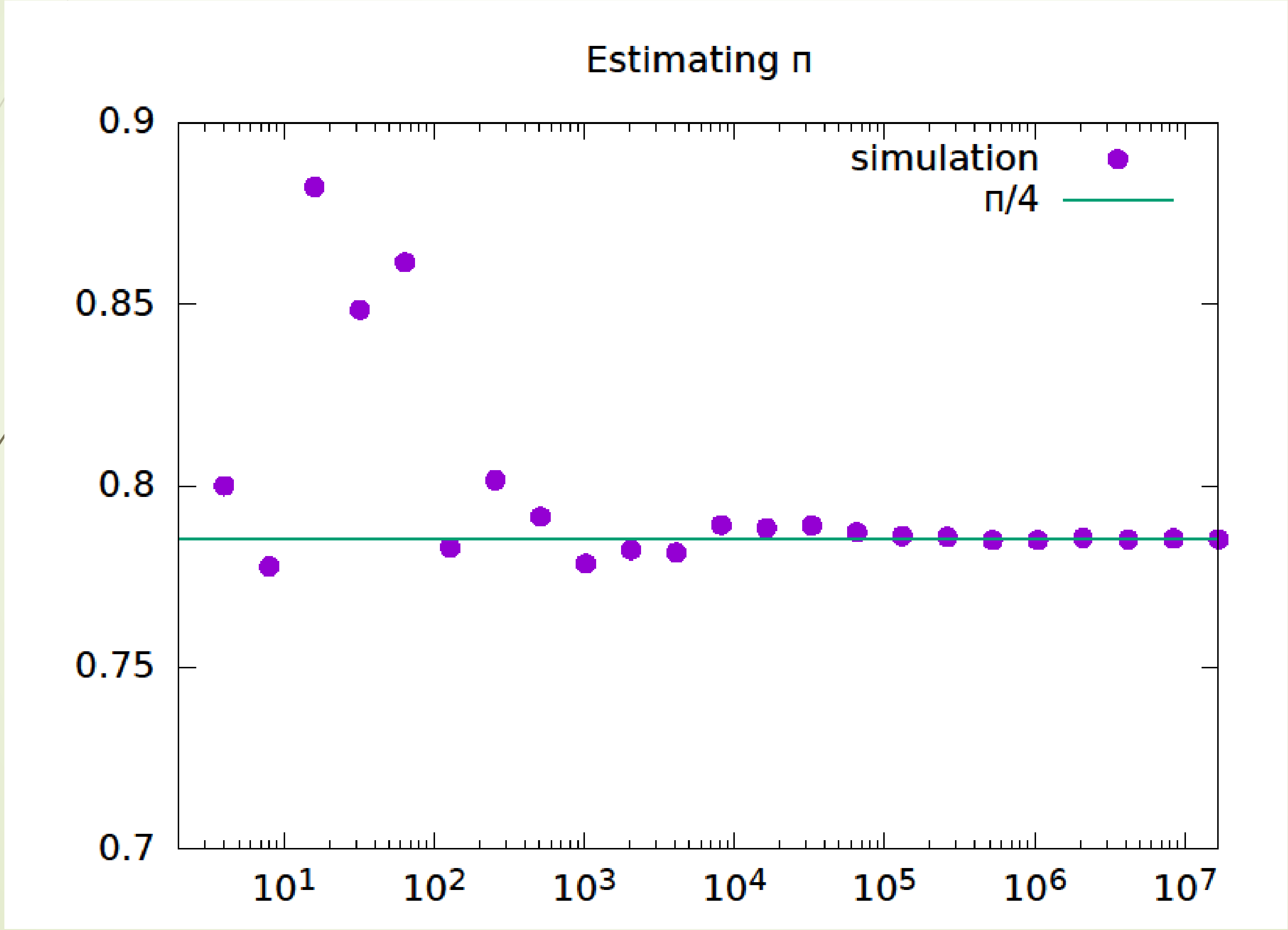
$$\langle m^2 \rangle - \langle m \rangle = G''(1) = N(N-1) p^2$$

$$\sigma^2 = \langle m^2 \rangle - \langle m \rangle^2 = N(N-1) p^2 + Np - N^2 p^2$$

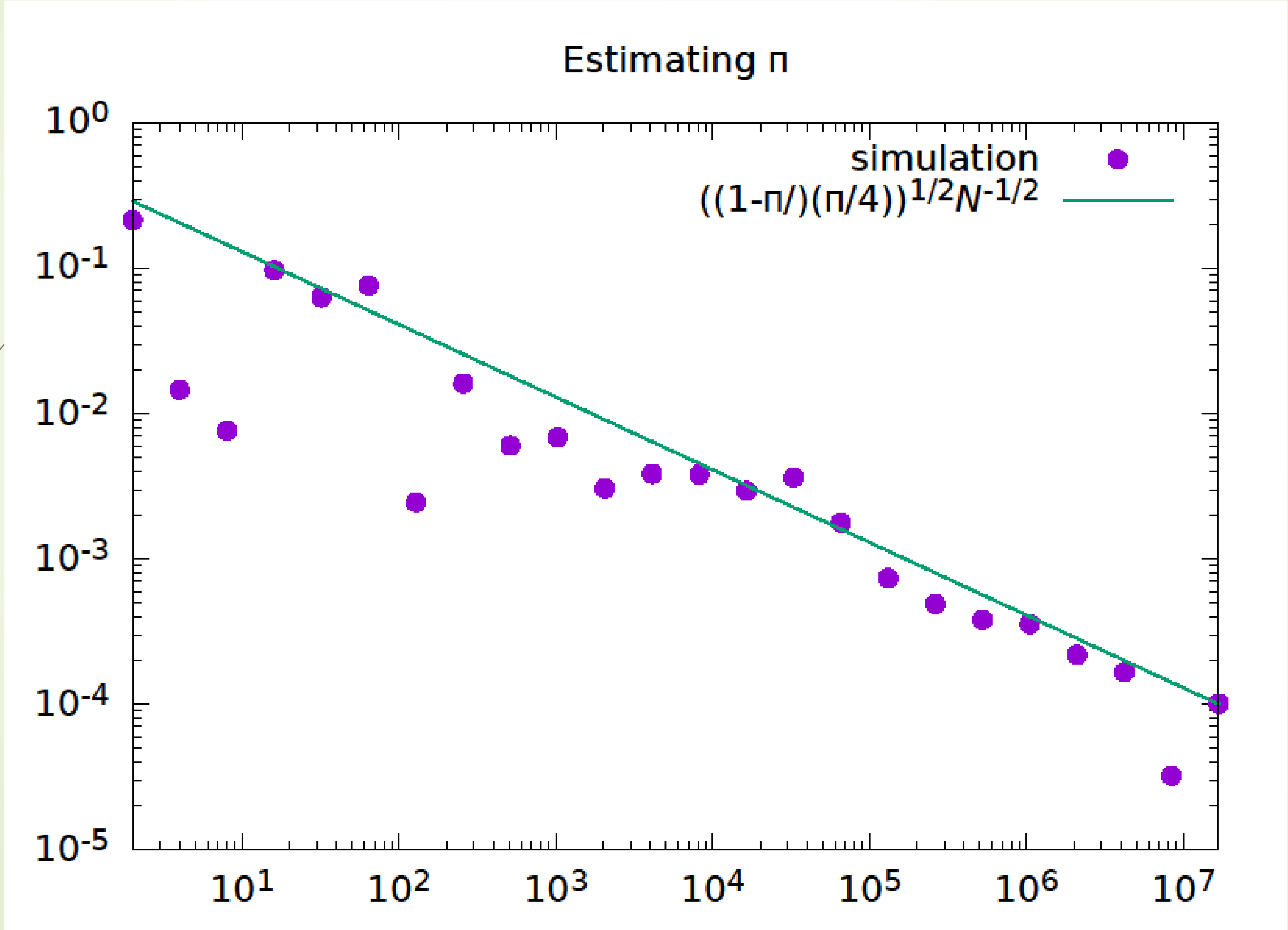
$$= Np(1 - p)$$

これが正しいことを確かめる ただし、簡単に

- $\langle m \rangle_{\exp}/N \sim p \cong \pi/4$
- $\frac{\sigma}{\langle m \rangle} = \left(\frac{1-p}{p}\right)^{1/2} N^{-1/2}$より
  - $\left| \langle m \rangle_{\exp}/N - \frac{\pi}{4} \right|$が$N^{-1/2}$でゼロに近づく

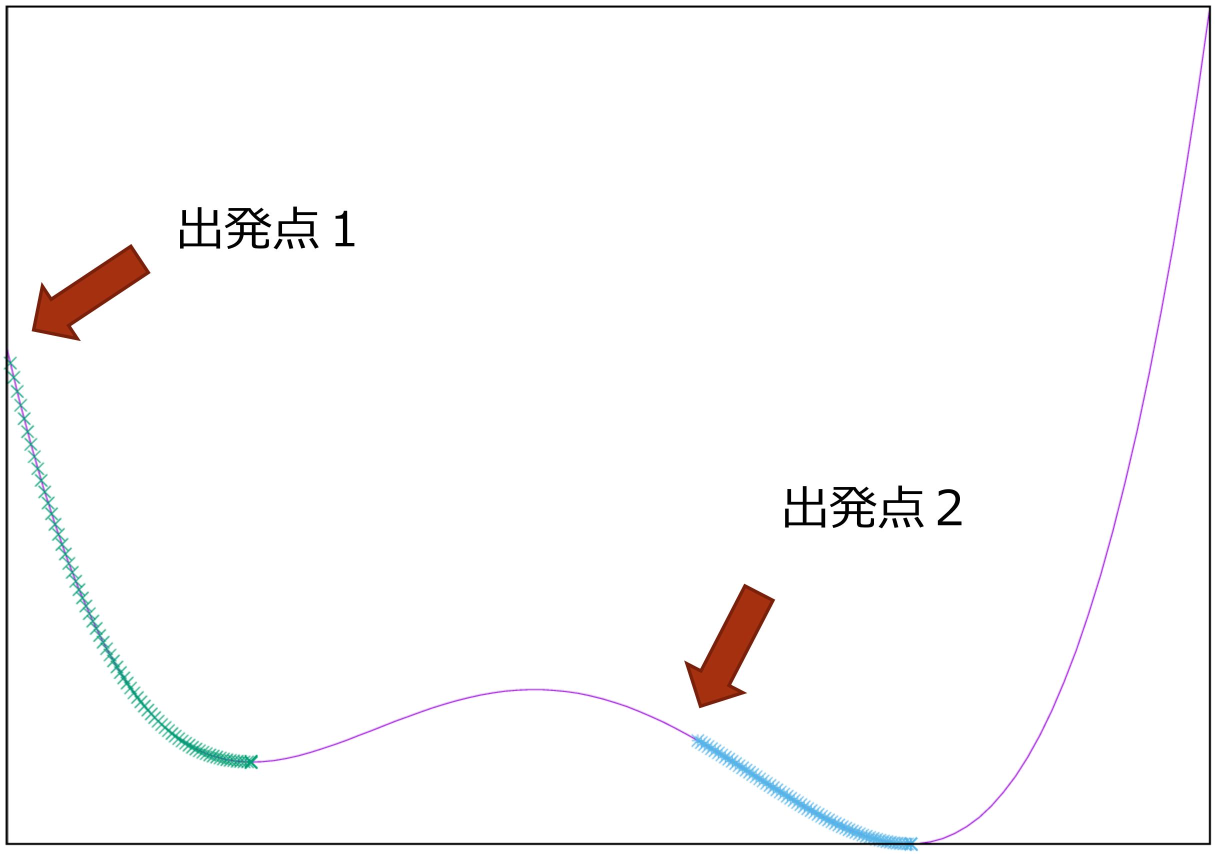


Estimating π
simulation
π/4
0.9
0.85
0.8
0.75
0.7
10^1
10^2
10^3
10^4
10^5
10^6
10^7


Estimating π
simulation
((1-π/)(π/4))^1/2N^-1/2
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^1
10^2
10^3
10^4
10^5
10^6
10^7

例：複数の極値を有する場合 Example : finding minima

- 関数$f(x)$の極小値を求める
  - 極小値が一つならば、適当な出発点xから$f(x)$の値が小さくなるように、xを少しずつ変化させる
  - 極小値が複数ならば
    - 出発点をランダムに選んで、複数回試行する

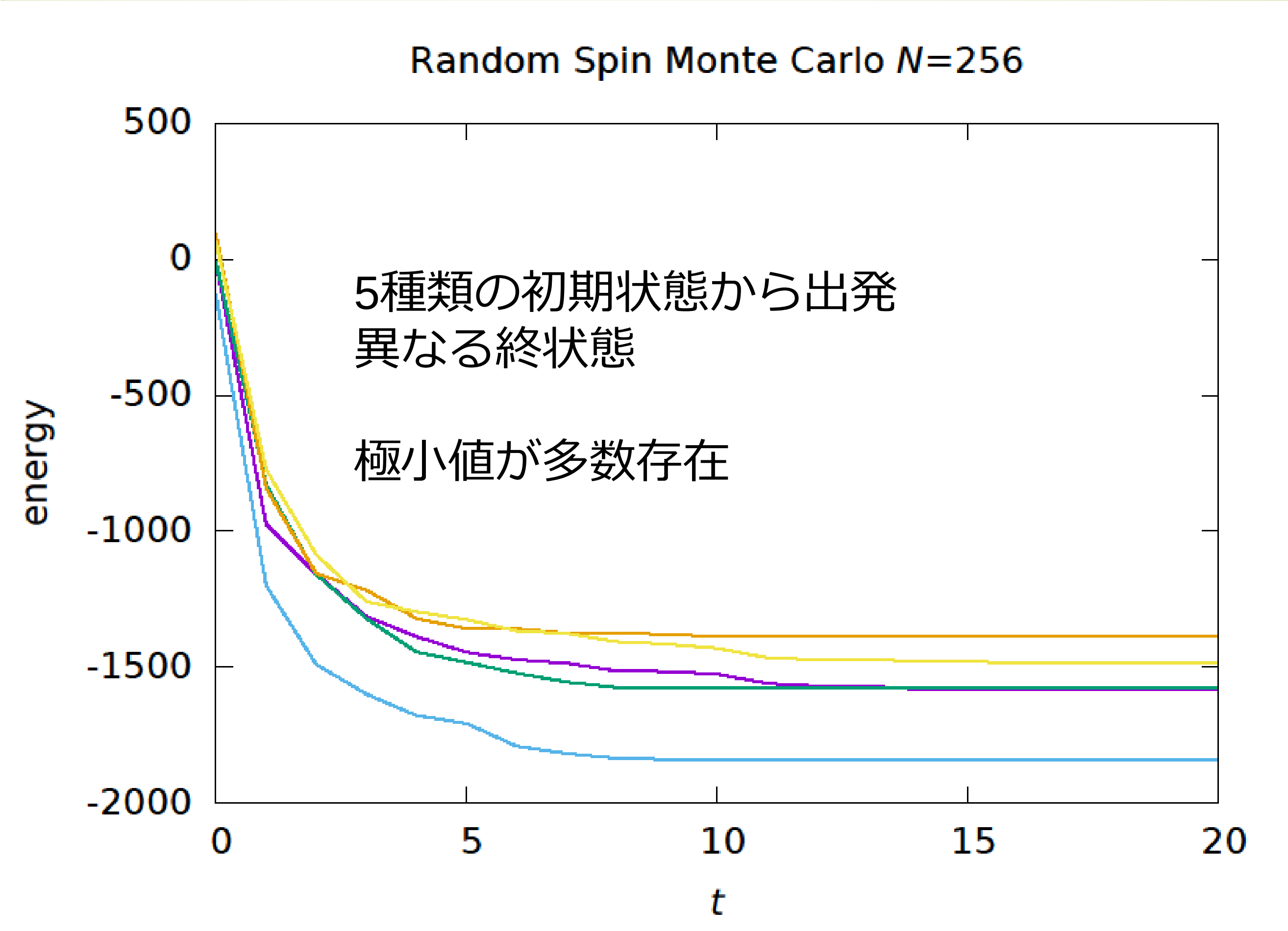

出発点 1
出発点 2

random spin系
Example : random spins

- n個の± 1を取る変数s_i
- 相互作用J_{ij} $(J_{ij} = J_{ji}, J_{ii} = 0)$
  - 正負の値がランダム
- $E = -\sum_{ij} J_{ij} s_i s_j$を最小にする
  - s_iを二つのグループに分ける
  - 二分割問題の一種

spin系のMonte Carlo法

- 毎回、ランダムにs_iを選び、変化
  - $s_i \to s_i + \Delta \mathrm{s_i}$
- $\Delta E = -2\sum_j J_{ij} s_j \Delta s_i < 0$ならば$s_i$を変更する（符号を反転する）
- 1 Monte Carlo step : n回の更新



Monte Carlo step