# 数値計算法
2011/6/29

林田　清
大阪大学大学院理学研究科

# プログラム例

- ここではseedは任意の数(できれば素数)を入力する。
  - srandom(time(NULL));として時間情報を種とすることもできる。この場合はtime.hをincludeする必要あり。
- 右の例でpi1は整数の割り算が最初に行われるので値がゼロになる。pi2, pi3のような記法にせよ。
- RAND_MAXは/usr/include/stdlib.hの中で定義(#define)されている。

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

int main()
{
  int seed, n, i, m;
  double x,y,pi1,pi2,pi3;

  printf("input seed number¥n");
  scanf("%d",&seed);

  printf("input number of points ¥n");
  scanf("%d",&n);

  srandom(seed);

  m=0;
  for(i=0;i<n;i++) {
    x=((double)random())/(RAND_MAX+1.0);
    y=((double)random())/(RAND_MAX+1.0);
    if((x*x+y*y)<1.0) m++;
    printf("%d x=%lf y=%lf¥n",i,x,y);
  }
  pi1 = m/n * 4.0;
  pi2 = (m*4.0)/n;
  pi3 = ((double) m)/((double) n )*4.0;
  printf("pi=%lf %lf %lf¥n",pi1,pi2,pi3);
}
```

# プリプロセッサ

- #include <*fname.h*>
  - コンパイルされる際に、ソースコードのこの位置に*fname.h*(通常は/usr/includeの下にある;具体的にはstdio.h,stdlib.h,math.h等)の内容が挿入される。
- #define *ABC* *abc*
  - ソースコードのこの位置以降にあるABCという文字列があれば、コンパイル時に自動的にabcにおきかえられて解釈される。
  - RAND_MAXは stdlib.hの中でdefineされている。

# 与えられた分布関数に従う乱数1:逆変換法

分布関数$f(x)$に従う乱数$x$を発生させたい

累積分布関数$F(x)$

$$F(x)=\int_{x_{\min}}^{x} f(x')dx'$$

は区間(0, 1)に一様にランダムに分布する。

もし$F$の逆関数$F^{-1}$をみつけることができれば$x=F^{-1}(r)$により

擬似一様乱数$r$から分布関数$f(x)$に従う乱数$x$を求めることができる

例：$f(x)=\lambda\exp(-\lambda x)$

$$F(x)=\int_0^x f(x')dx'=\left[-\exp(-\lambda x')\right]_0^x=1-\exp(-\lambda x)$$

$$x=-\frac{1}{\lambda}\ln(1-F(x))=-\frac{1}{\lambda}\ln(r)$$

# 逆変換法の考え方

$r$が0<r<1で定義される一様乱数のとき

確率分布は　$p_{\mathrm{r}}(r)dr = dr \quad (0 < r < 1)$

この$r$がある関数$x(r)$で$x$に変換されるとき

$x$の確率分布$p_{\mathrm{x}}(x)$は$p_{\mathrm{x}}(x)dx = p_{\mathrm{r}}(r)dr$から

算出できる。

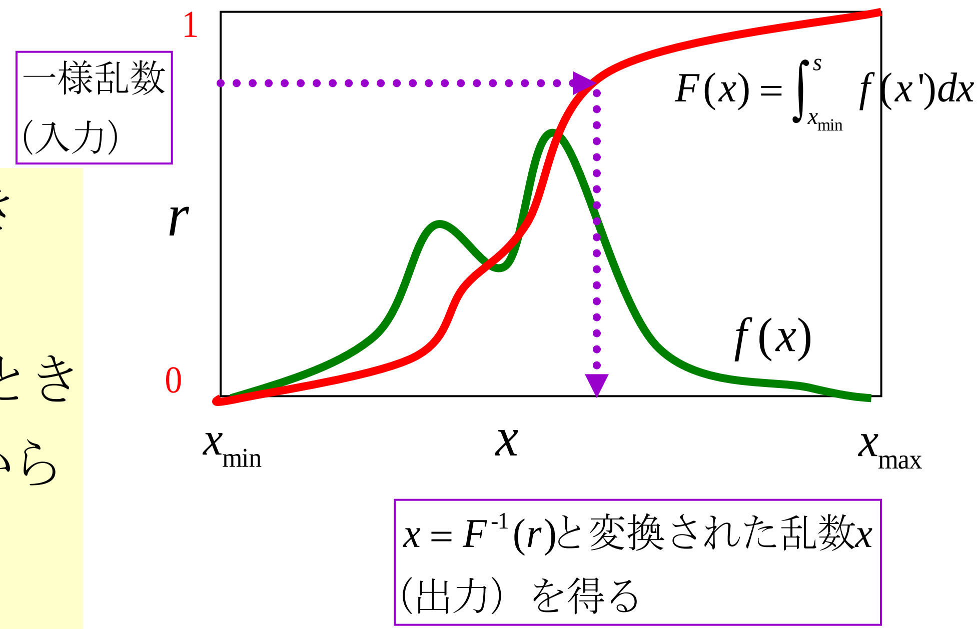
$$p_{\mathrm{x}}(x) = \frac{dr}{dx} \quad (x(r=0) < x < x(r=1))$$

逆変換法では$r = F(x)$従って$x = F^{-1}(r)$と

とる。すると

$$p_{\mathrm{x}}(x) = \frac{dF(x)}{dx} = f(x) \quad (x_{\min} < x < x_{\max})$$

$x$の確率分布は$f(x)$になる。




$F^{-1}(r)$が容易に計算できることが必要

# 正規乱数の発生

- 積分は誤差関数、解析的に記述できない。
  - Box-Muller法か中心極限定理

$$x=\sum_{i=1}^{12} r_i - 6$$

ここで$r_i$は(0,1)の一様乱数

一般に確率密度関数$q(x_1,x_2)$に従う変数$x_1,x_2$の関数$y_1,y_2$の確率密度関数は

$$p(y_1,y_2)dy_1dy_2 = q(x_1,x_2)\left|\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)}\right|dy_1dy_2$$

$$p(y_1,y_2)dy_1dy_2 = \left[\frac{1}{\sqrt{2\pi}}\exp(-y_1^{\,2}/2)\right]\left[\frac{1}{\sqrt{2\pi}}\exp(-y_2^{\,2}/2)\right]dy_1dy_2$$

となるような変数$y_1,y_2$を生成したい$\Rightarrow$ Box-Muller法

$$y_1=\sqrt{-2\ln x_1}\cos 2\pi x_2,\ y_2=\sqrt{-2\ln x_1}\sin 2\pi x_2$$

$x_1,x_2$が区間(0, 1)の独立な一様乱数であれば$q(x_1,x_2)dx_1dx_2 = dx_1dx_2$

$$x_1=\exp\left[-\frac{1}{2}(y_1^{\,2}+y_2^{\,2})\right],\ x_2=\frac{1}{2\pi}\arctan\frac{y_2}{y_1}$$

$$\frac{\partial(x_1,x_2)}{\partial(y_1,y_2)} = -\left[\frac{1}{\sqrt{2\pi}}\exp(-y_1^{\,2}/2)\right]\left[\frac{1}{\sqrt{2\pi}}\exp(-y_2^{\,2}/2)\right]$$

# 与えられた分布関数に従う乱数2:棄却法

発生が容易な分布関数（例えば一様分布）$g(x)$の定数倍$Cg(x)$によって$f(x)$を完全に包み込む。

1. $g(x)$の確率分布に従う乱数$x_0$を発生させる

２． 点$x_0$で$f(x_0)$と$Cg(x_0)$を計算する。(0,1)の乱数$r$を発生させ、

　(a) $f(x_0) < rCg(x_0)$ならば、$x_0$を棄ててステップ１からやり直す

　(b) $f(x_0) \geq rCg(x_0)$ならば、$x_0$が求める乱数