２０２１年度コンピュータ演習
第１５回：１月１４日

1. 授業計画
2. 【復習】：擬似乱数の作成と確認
3. 【練習１】：モンテカルロ（MC）法
   球の体積（円周率）
4. 【練習２】：作図（gnuplot）
5. 小演習９：モンテカルロ（MC）法
   課題：回転体の体積と作図
6. 課題の提出（1/21 14:30 締切り）

２．【復習】擬似乱数の作成と確認

- 乱数列 $x_1, x_2, x_3,$ ・・・とは数が不規則に並んだ数列のこと．コンピュータで作る**擬似乱数**（pseudo-random numbers）は，一定の規則に従い数を生成するので，厳密には乱数ではないが，通常の応用には乱数と見なして差し支えない．

- C言語の標準ライブラリには，整数の一様乱数を発生する関数 rand() があり，0 以上 RAND_MAX（2,147,483,647）以下の整数の擬似乱数を生成する．（ #include <stdlib.h>）

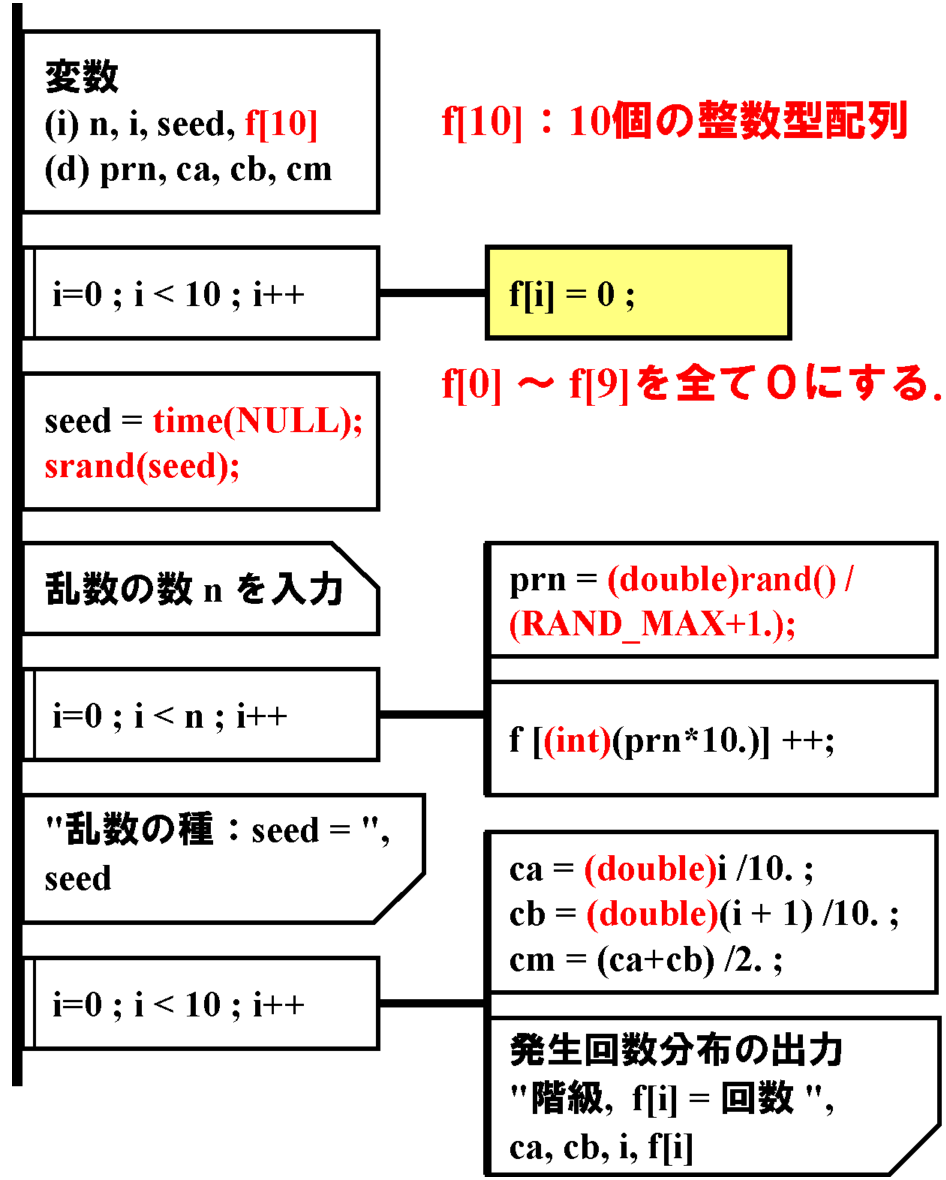
- 乱数生成のアルゴリズムは**シード（種）値**で開始する．srand(*seed*)の種値 *seed* には，コンピュータの現在時刻 time(NULL)がしばしば利用される．（毎回異なる値を種値とするため．#include <time.h> が必要です．）

２．【復習】擬似乱数の作成と確認

【練習１】：

右のPADを参考にして，
０以上１未満の擬似乱数
prn（$0 \leqq prn < 1$）を
発生させるプログラムを
作成する．

作成した擬似乱数の0.1
間隔の発生回数分布を
確認する．



2．【復習】擬似乱数の作成と確認

```
// 擬似乱数の作成

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

int main(){
  int n,i,seed,f[10];
  double prn,ca,cb,cm;

  for(i=0 ; i<10 ; i++) f[i]=0;
    // 発生回数のリセット
  seed = time(NULL); // 擬似乱数の種
  printf("¥n seed= %d¥n¥n",seed);
    // 種値の確認
  srand((unsigned int)seed);
    // 擬似乱数の初期化

  printf("¥n擬似乱数の作成数：n を
   入力して下さい。¥n");
  scanf("%d", &n);

  for(i=0 ; i<n ; i++){    // 擬似乱数をn個作成.
    prn = (double)rand()/(RAND_MAX+1.);
    f[(int)(prn*10.)]++;   // 乱数の回数を数える.
  }

  for(i=0 ; i<10 ; i++){     // 擬似乱数の分布
    ca = (double)i/10.;      // 階級下限
    cb = (double)(i+1)/10.; // 階級上限
    cm = (ca+cb)/2.; // 階級中間値
    printf("%3.1lf - %3.1lf, f[%d]=%d¥n",
    ca,cb,i,f[i]);
  }
}

// 解答例では，gnuplotにより棒グラフを作成します.
```

２．【復習】擬似乱数の作成と確認

※ 擬似乱数 prn を作る数値の範囲を変える.

(1) prn = rand();：0 ≦ prn ≦ RAND_MAX の範囲の整数.

(2) 0 ～ 1 の範囲の浮動小数点数を作りたい場合：

prn = (double) rand() / (RAND_MAX + 1.)：（0 ≦ prn ＜ 1）

prn = (double) rand() / RAND_MAX：（0 ≦ prn ≦ 1）

(3) 0 ～ Nmax の範囲の浮動小数点数を作りたい場合

prn = Nmax*((double) rand() / RAND_MAX)

：（0≦*r*≦Nmax）

(4) -Nmax～Nmaxの範囲の浮動小数点数を作りたい場合

r = 2.*Nmax*((double)rand() / RAND_MAX) - Nmax

：（-Nmax ≦ *r* ≦ Nmax）

２．【復習】擬似乱数の作成と確認

- 配列の使い方

```
int n,i,seed,f[10]; // 宣言
```

整数型の変数１０個分の配列

f[0], f[1], ・・・, f[9]

プログラムでは，乱数の個数

```
f[(int)(prn*10.)]
```

f[0]：$0 \leqq prn < 0.1$

f[1]：$0.1 \leqq prn < 0.2$

...

f[9]：$0.9 \leqq prn < 1$

３．【練習１】モンテカルロ（MC）法

- **モンテカルロ法（Monte Carlo methods）**

　乱数を使って数学や物理などの問題を解く手法.

1949年に，MetropolisとUramがギャンブル場で有名なモナコの町名にちなんで命名した.

（ギャンブル好きのS. Uramの叔父が毎年モンテ

　カルロに通っていた？）

　材料科学の分野では，「拡散」，

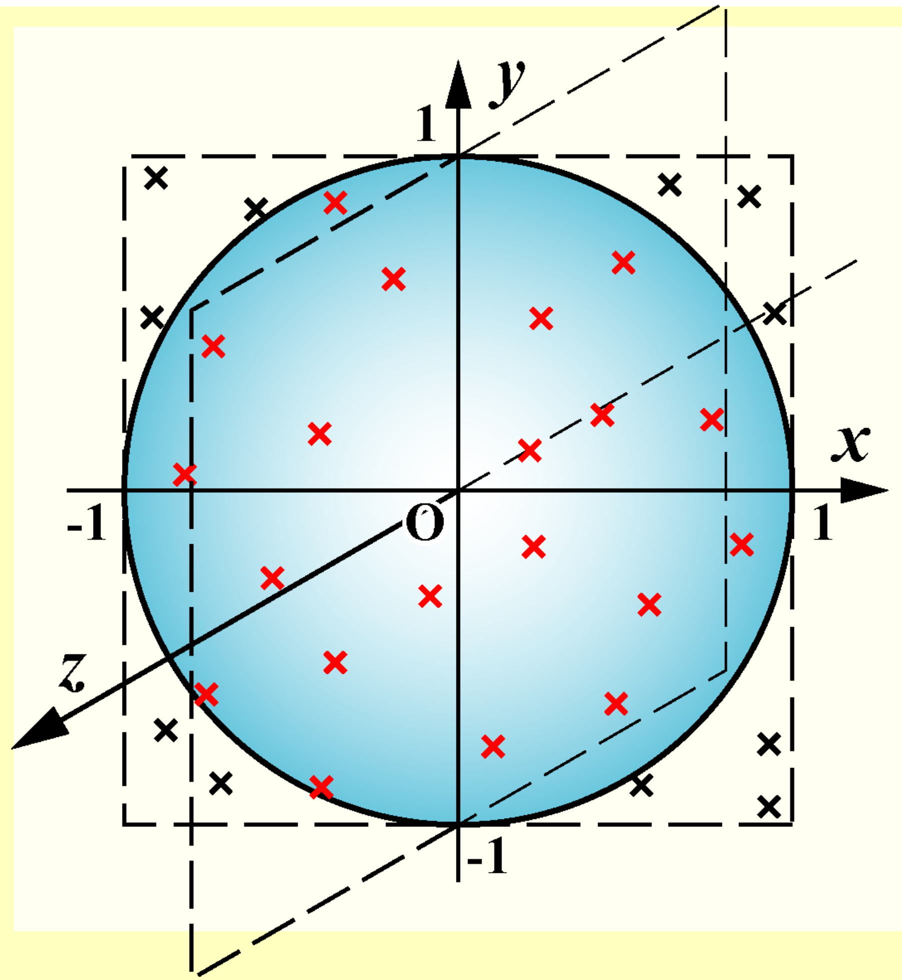
「エピタキシャル成長」，「イジングモデル」，

「結晶粒成長」，「状態図」，「相転移」などの

シミュレーションに応用されている.

３．【練習１】MC法による球の体積

・いわゆる“あたりはずれ”
（hit-or-miss）法.
右図のような，立方体領域
（$-1 \leqq x \leqq 1$，$-1 \leqq y \leqq 1$，
$-1 \leqq z \leqq 1$，体積８）内に，
擬似乱数で点 (x, y, z) を打つ.
この点が，原点Oを中心と
する半径１の球内に入る
（すなわち，$x^2 + y^2 + z^2 < 1$）
確率を８倍（立方体の体積）
すれば，球の体積 v（＝$4\pi/3$）
を求めることができる.



図：モンテカルロ法による球の体積

３．【練習１】MC法による球の体積

【練習１】

右のPAD を参考にして，モンテカルロ法のプログラムを作成し，半径１の球の体積 v を計算する．

（14_pr1.c）

【練習２】試行回数 n が n=50*2^m（m = 0, 1, 2, ... n < 1E7）の時の v を計算し，n と π の関係を確認・図示する．

（14_pr2.c）

変数
(i) n, seed, i, c
(d) x, y, z, rr, v, pi

seed = time(NULL) ;
srand(seed) ;

c = 0 ;

n 入力

i=0 ; i < n ; i++

x = 2.*rand() / RAND_MAX - 1. ;
y = 2.*rand() / RAND_MAX - 1. ;
z = 2.*rand() / RAND_MAX - 1. ;
rr = x*x + y*y + z*z ;

rr < 1.

c++ ;

v = 8. * c / n ;
pi = 3.*v / 4. ;

計算結果の出力
"試行回数：n = ", n
"円の面積：v = ", v
"円周率：pi = ", pi

n： 乱数の発生回数．
c： 乱数の点(x,y,z)が球の中に入った回数．
v： 球の体積
v0：立方体の体積
c : n = v : v0
(v0 = 2.*2.*2.)

３．【練習１】MC法による球の体積

【練習１】

$-1 \leqq x, y, z \leqq 1$のとき，

点Ｐ（x, y, z）は一辺が２の立方体（体積８）の中．

これが，半径$r = 1$の球の内部に入る条件は，

◆ $r^2 = x^2 + y^2 + z^2 < 1$

このとき，球の体積vは

◆ v（$= 4\pi / 3$）

＝（立方体の体積）×（Ｐが球内に入る確率）

で計算できる．円周率πは，

◆ $\pi = 3v / 4$　で計算できる．

３．【練習１】MC法による球の体積

```
// Monte-Carlo methods
#include <stdlib.h>
#include <stdio.h>
#include <time.h>
#include <math.h>

int main(){
  int n,seed,i,c;
  double x,y,z,rr,v,pi;

  seed = time(NULL);  // 擬似乱数の種
  srand((unsigned int)seed);
      // 擬似乱数初期化

  printf("¥n試行回数：n は？¥n  n= ");
  scanf("%d", &n);

  c = 0;  // 球内に入った回数の初期化
  for(i=0 ; i<n ; i++){
    x = 2.*rand()/RAND_MAX -1.;
    y = 2.*rand()/RAND_MAX -1.;
    z = 2.*rand()/RAND_MAX -1.;
    rr = x*x + y*y + z*z;
    if(1. > rr) c++; // 球の中（当り）
  }

  // 球の体積を計算する.
  v = 8.*(double)c/(double)n;
      // （立方体の体積）×（球内の確率）
  pi = 3.*v/4.;  // 円周率

  printf("¥n試行回数：n = %d¥n",n);
  printf("球の体積：v =%9.7f¥n¥n",v);
  printf("円周率  ：pi=%9.7f¥n¥n",pi);
}
```

※ 試行回数 *n* を増やすと，円周率の計算値 *pi* が次第に π（= 3.1415926・・・）に近づきます．

⇒ 【練習②】*n* と *pi* の関係をグラフにする．

４．【練習２】試行回数 n と円周率 π

```
// Monte-Carlo methods

#include <stdlib.h>
#include <stdio.h>
#include <math.h>
#include <time.h>

#define NMAX 1E7
#define GNUPLOT "/usr/bin/gnuplot"  // gnuplot 用

int main(){
  int seed,i,j,c,step;
  double x,y,z,rr,v,pi;
  FILE *fp,*pipe;

  fp = fopen("pi.dat","w");

  seed = time(NULL); // 擬似乱数の種
  srand((unsigned int)seed); // 擬似乱数の初期化
```

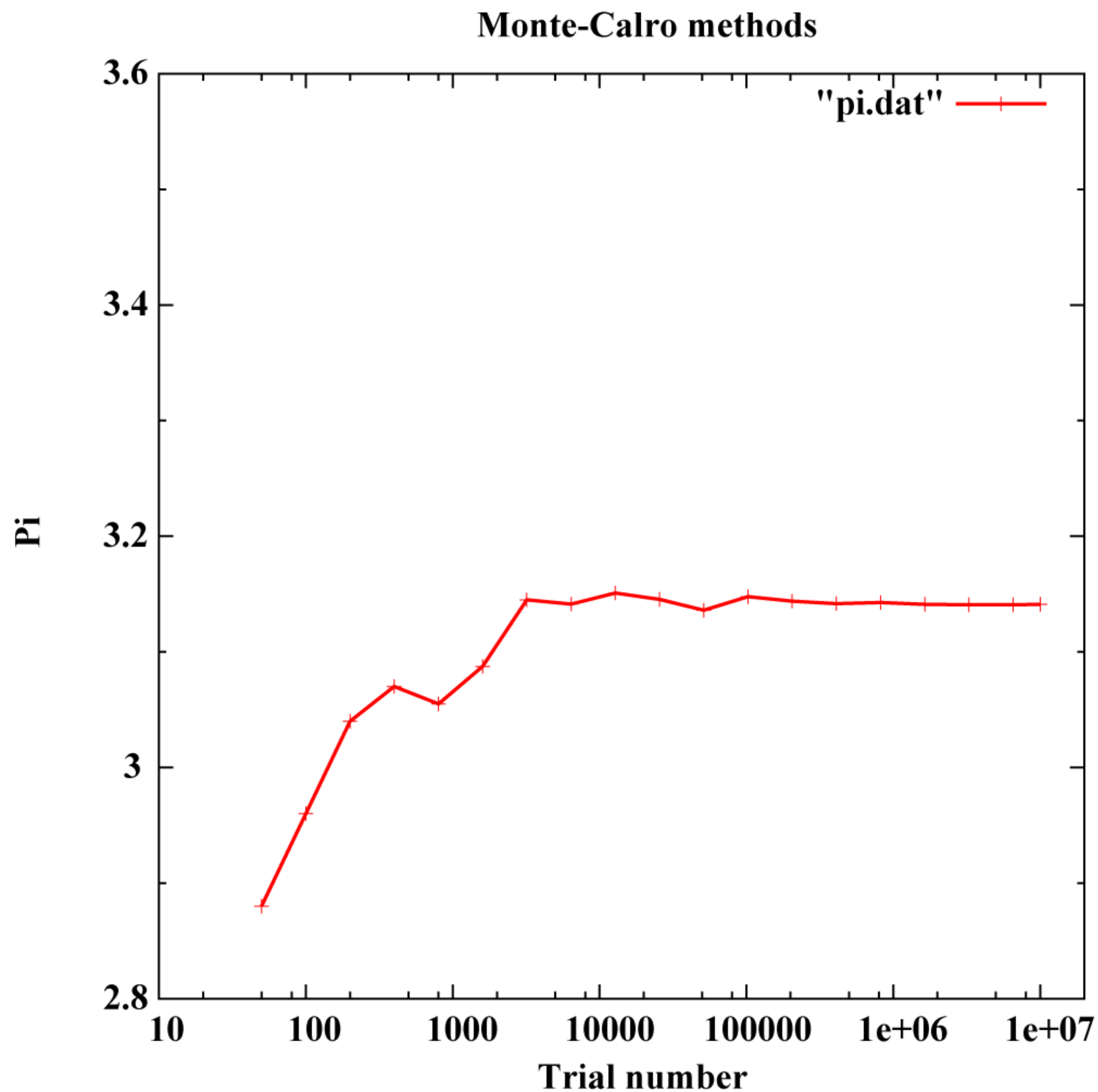
```
c = 0;  // 球内に入った回数の初期化
step = 50;  // データ保存回の初期値
for(i=0 ; i<NMAX ; i++){  // NMAX = 1E7
  x = 2.*(double)rand()/RAND_MAX -1.;
  y = 2.*(double)rand()/RAND_MAX -1.;
  z = 2.*(double)rand()/RAND_MAX -1.;
  rr = x*x + y*y + z*z;
  if(1. > rr) c++; // 球の中（当り）
  if(i+1 == step){
    v = 8.*(double)c/(double)(i+1.);  // 球の体積
    pi = 3.*v/4.;  // 円周率
    printf("¥n試行回数：n= %d  /  球の体積：v= %9.7f  /  ",i+1,v);
    printf("円周率  ：pi= %9.7f¥n",pi);
    fprintf(fp,“%d  %9.7f ¥n”,i+1,pi);
    step *= 2; // データ保存回を２倍にする
  }
}

printf("¥n試行回数：n= %8.0f  /  球の体積：v= %9.7f  /  ",NMAX,v);
printf("円周率  ：pi= %9.7f¥n¥n",pi);
fprintf(fp,"%8.0f  %9.7f ¥n",NMAX,pi);
fclose(fp);
```

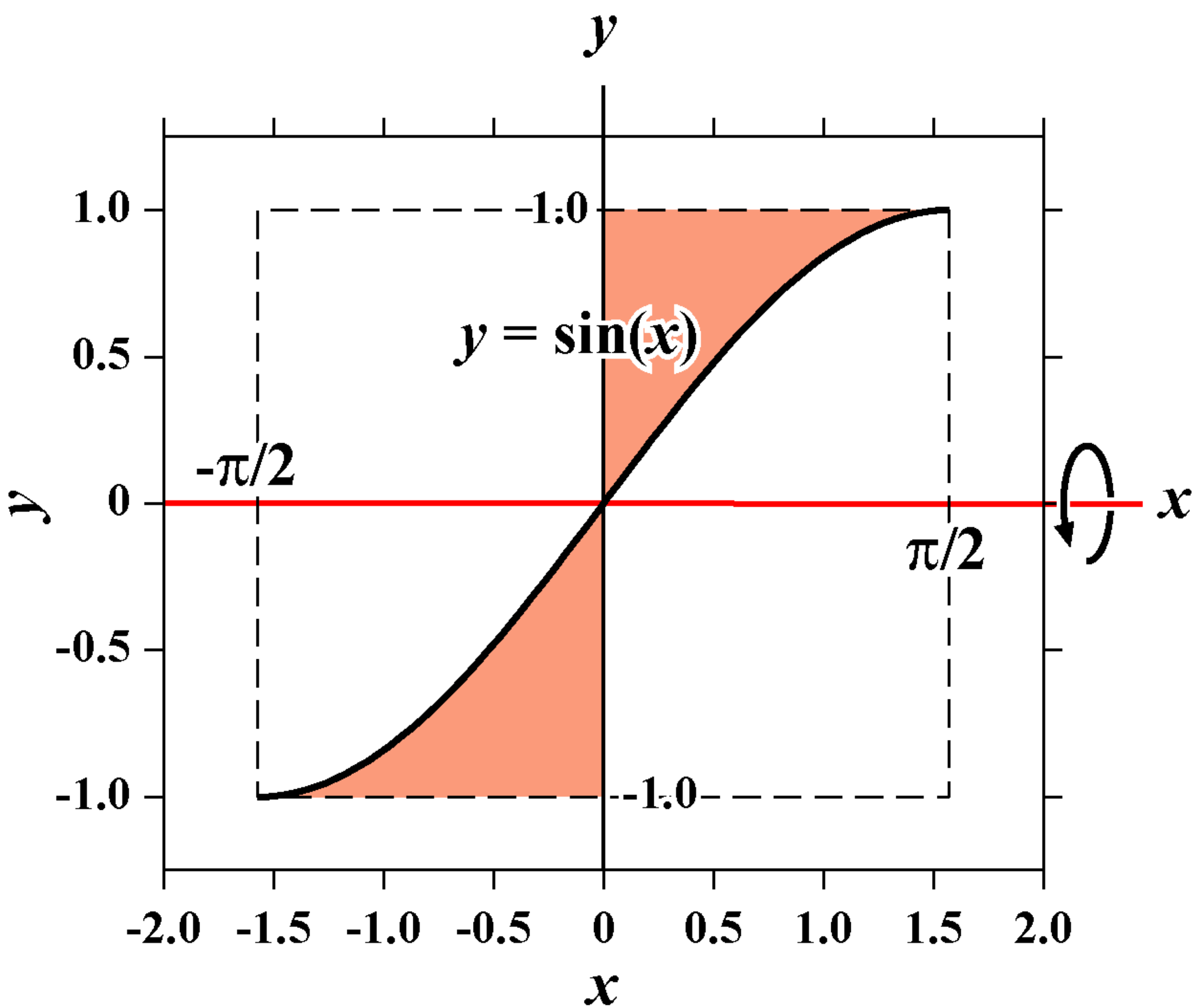
```
// 計算結果をgnuplotで画面に表示し，pngファイルに出力する．
  if(NULL == (pipe=popen(GNUPLOT " -persist","w"))){
    fprintf(stderr,"パイプが開けません¥n");
    exit(1);
  }
  fprintf(pipe,"set title ¥"Monte-Calro methods¥"¥n"); // タイトル
  fprintf(pipe,"set xlabel ¥"Trial number¥"¥n");  // x軸のラベル
  fprintf(pipe,"set ylabel ¥"Pi¥"¥n");  // y軸のラベル
  fprintf(pipe,"set logscale x¥n");  // x軸を対数表示に
  fprintf(pipe,"set xrange [10:2E7]¥n");  // x軸の範囲
  fprintf(pipe,"set yrange [2.7:3.6]¥n");  // y軸の範囲
  fprintf(pipe,"set ytics 0.2¥n");  // y軸メモリの間隔
  fprintf(pipe,"set mytics 2¥n");  // y軸メモリ間の分割数
  fprintf(pipe,"set size square¥n");  // グラフの縦横比：正方形
  fprintf(pipe,"plot ¥"pi.dat¥" with linespoints ¥n"); // 点と線
  fflush(pipe);

  fprintf(pipe,"set term png ¥n"); // 出力先⇒png
  fprintf(pipe,"set output ¥"pi.png¥"¥n");  // pngファイル名
  fprintf(pipe,"replot ¥n");  // 再描画
  fflush(pipe);
  pclose(pipe);
}
```

４．【練習２】試行回数 n と円周率 π



５．小演習９：回転体の体積（MC法）



５．小演習９：回転体の体積（MC法）

【小演習９】

(1)【練習１】を参考にして，モンテカルロ法により $y = \sin(x)$（$-\pi/2 \leqq x \leqq \pi/2$）と$y$軸に囲まれる面を，$x$軸周りに１回転した回転体の体積 v を計算するプログラムを作成せよ．（学籍番号-9-1.c）

(2)【練習２】を参考にして，$n = 50*2^m$（$m = 0, 1, 2, \ldots, n < 1E7$）の時の体積 v を計算し，n と v の関係を図示するプログラムを作成せよ．
（学籍番号-9-2.c, 学籍番号-9-2.png）．

◆ 上記３つのファイルを提出して下さい．

※ 円周率として定数 PI（= 3.1415926）を定義する．

5．小演習９：回転体の体積（MC法）

変数
(i) i, c, seed, step
(d) x, y, z, rr, v
(F) *fp, *pipe

PI	3.1415926
NMAX	1E7
GNUPLOT	"/usr/bin/gnuplot"

fp = fopen(...) ;

seed = time(NULL) ;
srand(seed) ;

c = 0 ;

step = 50 ;

i=0 ; i < NMAX ; i++

x =
y =
z =
rr =
考えてください

条件を考えて下さい.

c++ ;

i == step

v = 考えてください

計算結果の出力（画面）
"試行回数：n = ", i
"体積：v = ", v

計算結果の出力（fp）
i v

step = step*2 ;

v = 考えてください

計算結果の出力（画面）
"試行回数：n = ", i
"体積：v = ", v

計算結果の出力（fp）
i v

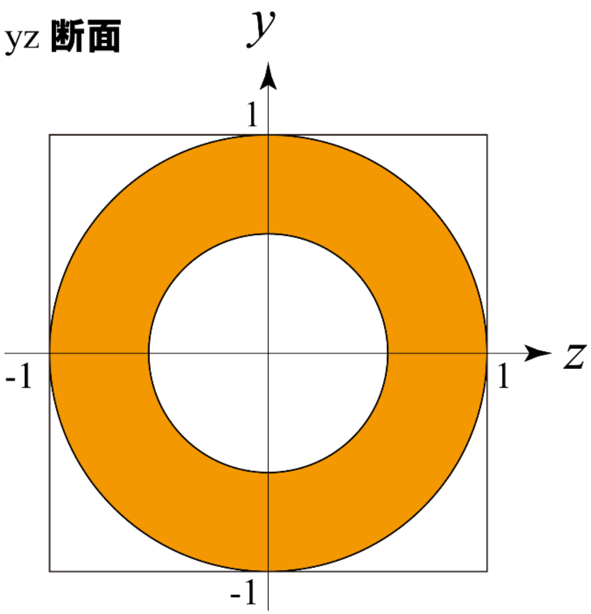
fclose(fp) ;

pipe = popen(...) ;
"gnuplotで作図"
pclose(pipe) ;

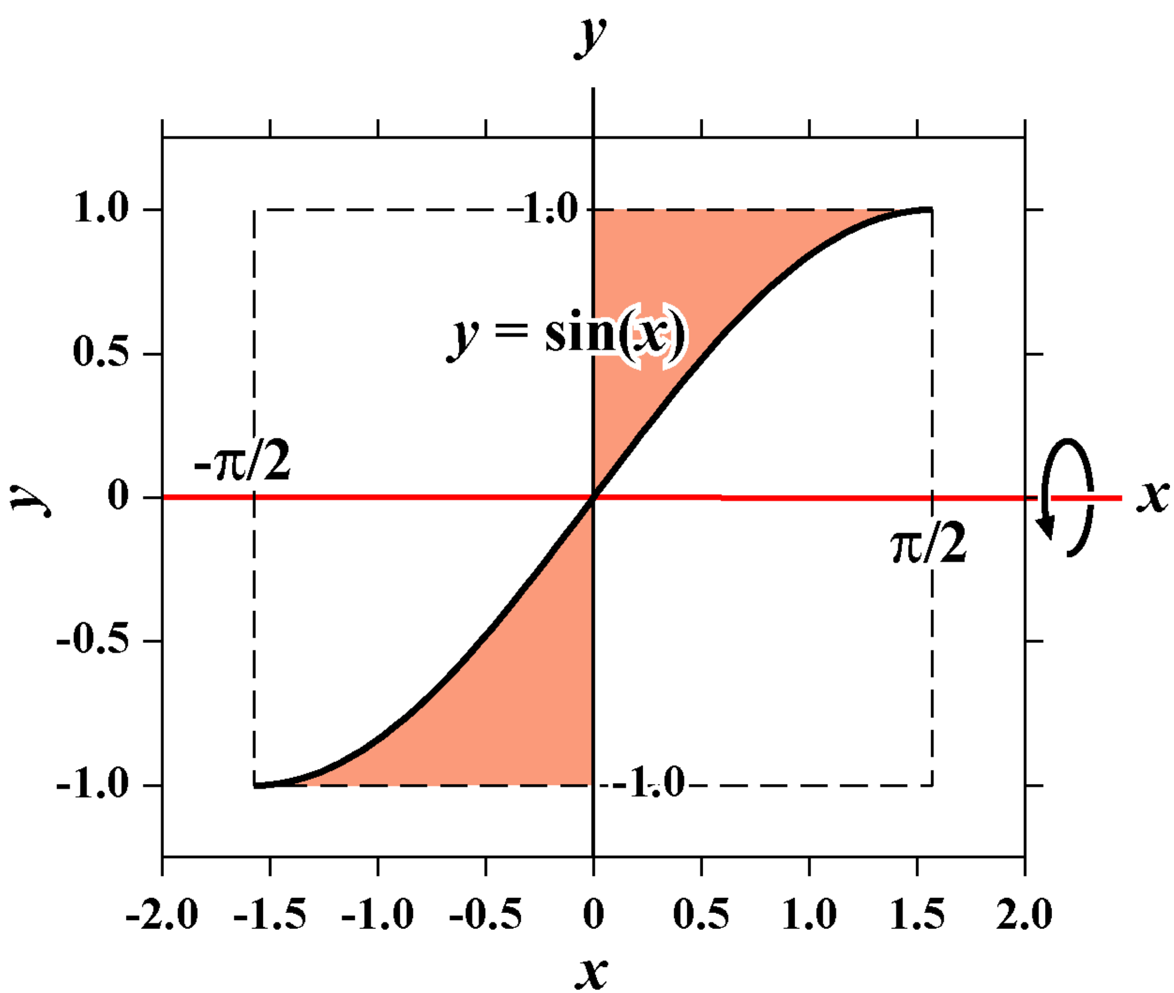
i： 乱数の発生回数.
c： 乱数の点(x,y,z)が
回転体の中に入った回数.
v： 回転体の体積
v0：立方体の体積
c : i = v : v0
(v0 = PI*2.*2.)

５．本日の作業

◆　ヒント：あるx座標に対するyz断面を考えると良い



＊円の方程式：$y^2 + z^2 = r^2$



５．本日の作業

【練習１・２】モンテカルロ法による球の体積と図示.

解答例“14_pr1.c”と“14_pr2.c”をダウンロードして，コンパイル・実行する.

【小演習９】モンテカルロ法による回転体の体積の計算

【練習１】を参考にして，$y = \sin(x)$（$-\pi/2 \leqq x \leqq \pi/2$ ）とy軸に囲まれる面をx軸周りに１回転した回転体の体積vを計算する．【練習２】を参考にして，計算結果vと試行回数nの関係を図示する.

◆ 課題提出：学籍番号-9-1.c，

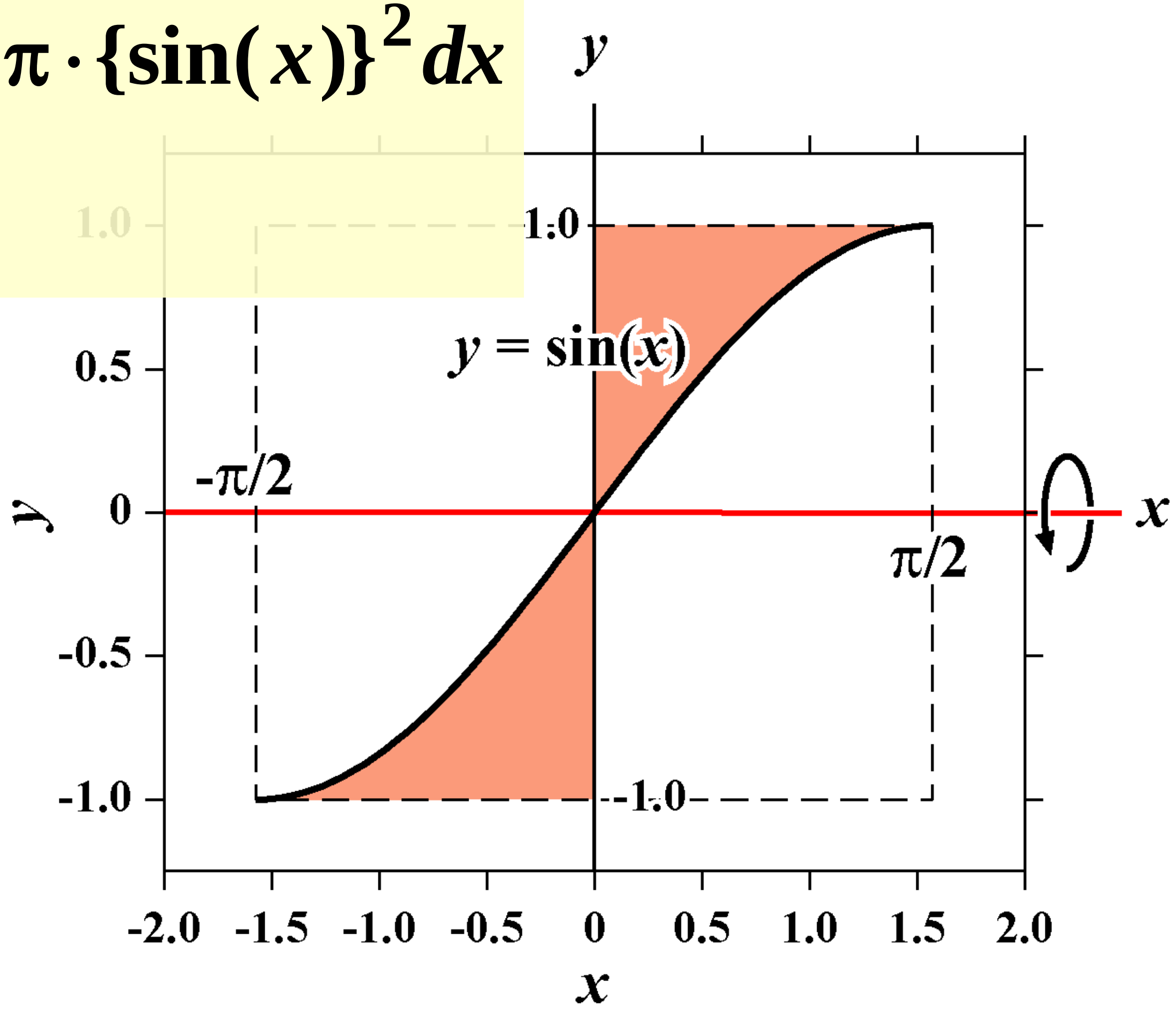
学籍番号-9-2.c，　学籍番号-9-2.png

５．本日の作業

◆ 回転体体積 v の計算値と比較して下さい．

$$v = \pi^2 - \int_{-\pi/2}^{\pi/2} \pi \cdot \{\sin(x)\}^2 dx$$

$$= 4.9348$$



今週のヒント

Q1. 『正しい条件文 if（～）の書き方？』

<u>*a* ≦ *x* ≦ *b*：*x* は *a* 以上 *b* 以下</u>

× `if(a <= x <= b)`：**この間違いが多い**

◎ `if((a<=x)&&(x<=b))`

×では，まず a<=x が判断され，これが正しければ１，正しくなければ０になる．その後，２つ目の不等号が判断される．

① １つ目の＜が真：　1 < b

② １つ目の＜が偽：　0 < b