環境科学計算機実験

担当：高須 夫悟

takasu@es.nara-wu.ac.jp

- 擬似乱数の生成と応用
- 4月14, 21, 28の 3回
- レポートで成績を評価する

現象のモデル化

- 多くの自然現象は、数理モデル、として記述できる
- 物体の自由落下、電気回路、放射性物質の崩壊、化学反応、生物集団の増減などは微分方程式で記述できる（初期値を決めれば振る舞いが一意に決まる決定論的モデル

質量 m の物体に力 F が作用するときの位置 x　　$$m\frac{d^2x}{dt^2} = F$$

放射能 radioactivity A と崩壊する原子の数 N　　$$A = \frac{dN}{dt} = -\lambda N$$

個体あたり出生率 b 死亡率 d の個体群動態　　$$\frac{dN}{dt} = bN - dN$$

現象のモデル化

- 多くの自然現象は、数理モデル、として記述できる
- 自然界には確率的に起こる現象がある（確率論的モデル）
- 確率論的モデルを実装するために乱数の生成が必要
- 乱数にもいろいろある。最も基本的なものが一様乱数

ブラウン運動（ランダムウォークの一つ）

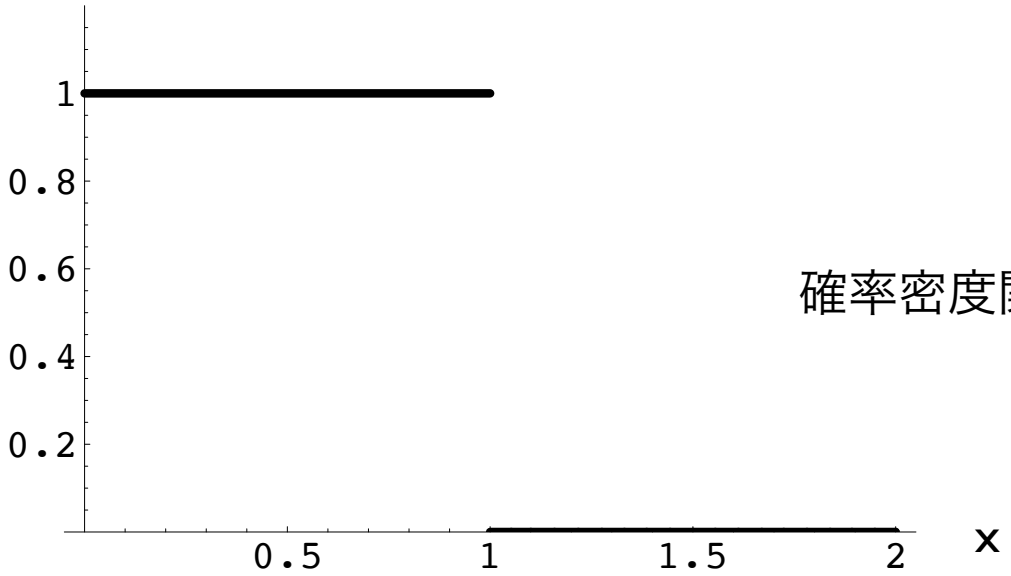
出生死亡過程、など

一様乱数

- 区間 [a, b) で一様な乱数 X を考える。X は a から b の値を同じ確からしさで取る連続確率変数

$$\mathrm{Prob}[x < X < x + dx] = \begin{cases} \frac{1}{b-a} dx & a \leq x < b \\ 0 & \mathrm{Otherwise} \end{cases}$$

a = 0, b = 1 の場合



確率密度関数はステップ関数

一様乱数

- 区間 [0, 1) で一様な乱数 U[0, 1) から任意の一様乱数を生成可能

$$U[0,1) \times a = U[0,a)$$

$$U[0,1) + b = U[b,1+b)$$

- どうやって U[0, 1) を生成するか？

擬似乱数の生成

- 擬似乱数をソフトウェア的に生成するアルゴリズム

線形合同法

$$X_{i+1} = (A \times X_i + B) \mod M$$

A mod B は整数 A を整数 B で割った余り

適当な整数値定数 A, B, M ($M > A, M > B, A > 0, B > 0$) を用いて、
X_0 を乱数の種 seed として逐次乱数列 X_i を生成する方法

最大値は $M - 1$、周期は最大で M

線形合同法

- 初期値 X_0 を与える漸化式に他ならない
- A, B, M をうまく選ぶとそれらしい擬似乱数が生成出来るか？

sample-1.c

```
#include <stdio.h>
#define IA    3877
#define IB    29573
#define IM    139968 // 周期！
int main (void) {

   int i, randomInt;
   double randomDouble;

   randomInt = 10;

   for(i=0; i<3000; i++){
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f\n", randomDouble);
   }

   return 0;
}
```

疑似乱数の質

- 線形合同法で生成された乱数 [0, 1) はどの程度良質か？
- 擬似乱数列 $\{X_0, X_1, X_2, \ldots\}$ を生成してファイルに書き出す
- 生成した疑似乱数列を gnuplot/Mathematica 等を用いて可視化する

```
% cc sample-1.c          線形合同法によるプログラムをコンパイルして実行

% ./a.out > data-1d      リダイレクションを用いて出力をファイルに保存
```

データの可視化

gnuplot でデータを読み込み視覚化する

```
% gnuplot
```

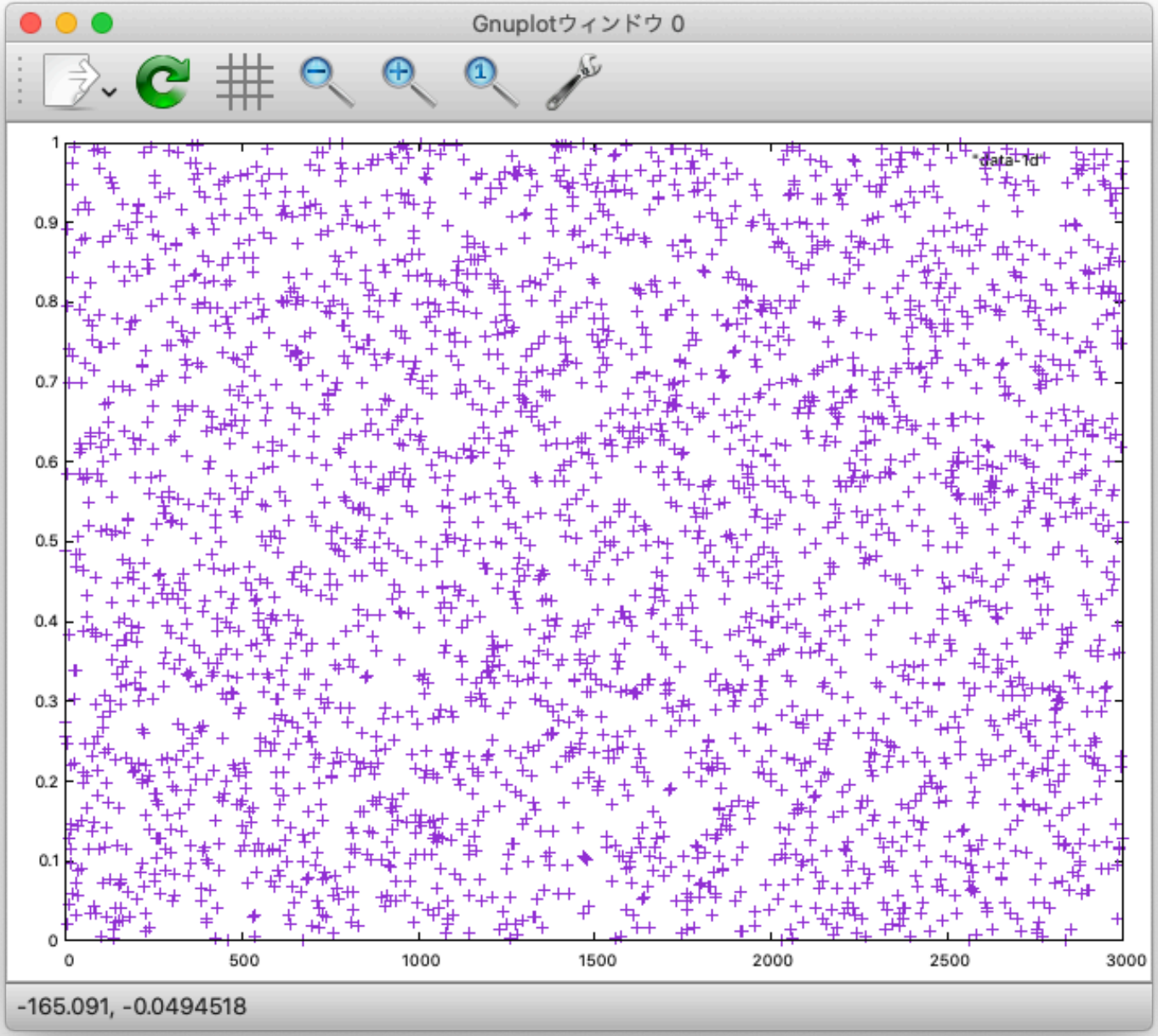
<— ターミナルからgnuplotを起動

```
        G N U P L O T
        Version 5.2 patchlevel 8    last modified 2019-12-01

        Copyright (C) 1986-1993, 1998, 2004, 2007-2019
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type is now 'qt'
gnuplot> plot "data-1d"
```

<— ファイル“data-1d” を読み込み視覚化

```
qt.qpa.fonts: Populating font family aliases took 271 ms. Replace uses of missing
font family "Sans" with one that exists to avoid this cost.
gnuplot>
```

データの可視化


Gnuplotウィンドウ 0
"data.fd"
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
500
1000
1500
2000
2500
3000
-165.091, -0.0494518

データの可視化

疑似乱数を2つ生成して2次元空間上の点として表現する（x, y座標を空白で区切る）

sample-2.c

```
#include <stdio.h>
#define IA    3877
#define IB    29573
#define IM    139968 // 周期！
int main (int argc, const char * argv[]) {

   int i, randomInt;
   double randomDouble;

   randomInt = 10;

   for(i=0; i<3000; i++){
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f ", randomDouble);          <— 出力
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f\n", randomDouble);         <— 出力
   }

   return 0;
}
```

疑似乱数の質

- 擬似乱数列を用いて2次元空間上の点として $\{X_0, X_1\}, \{X_2, X_3\}, \{X_4, X_5\}, \ldots$ をファイルに書き出す
- 生成した疑似乱数列を gnuplot/Mathematica 等を用いて可視化する

```
% cc sample-2.c          線形合同法によるプログラムをコンパイルして実行

% ./a.out > data-2d      リダイレクションを用いて出力をファイルに保存
```

データの可視化

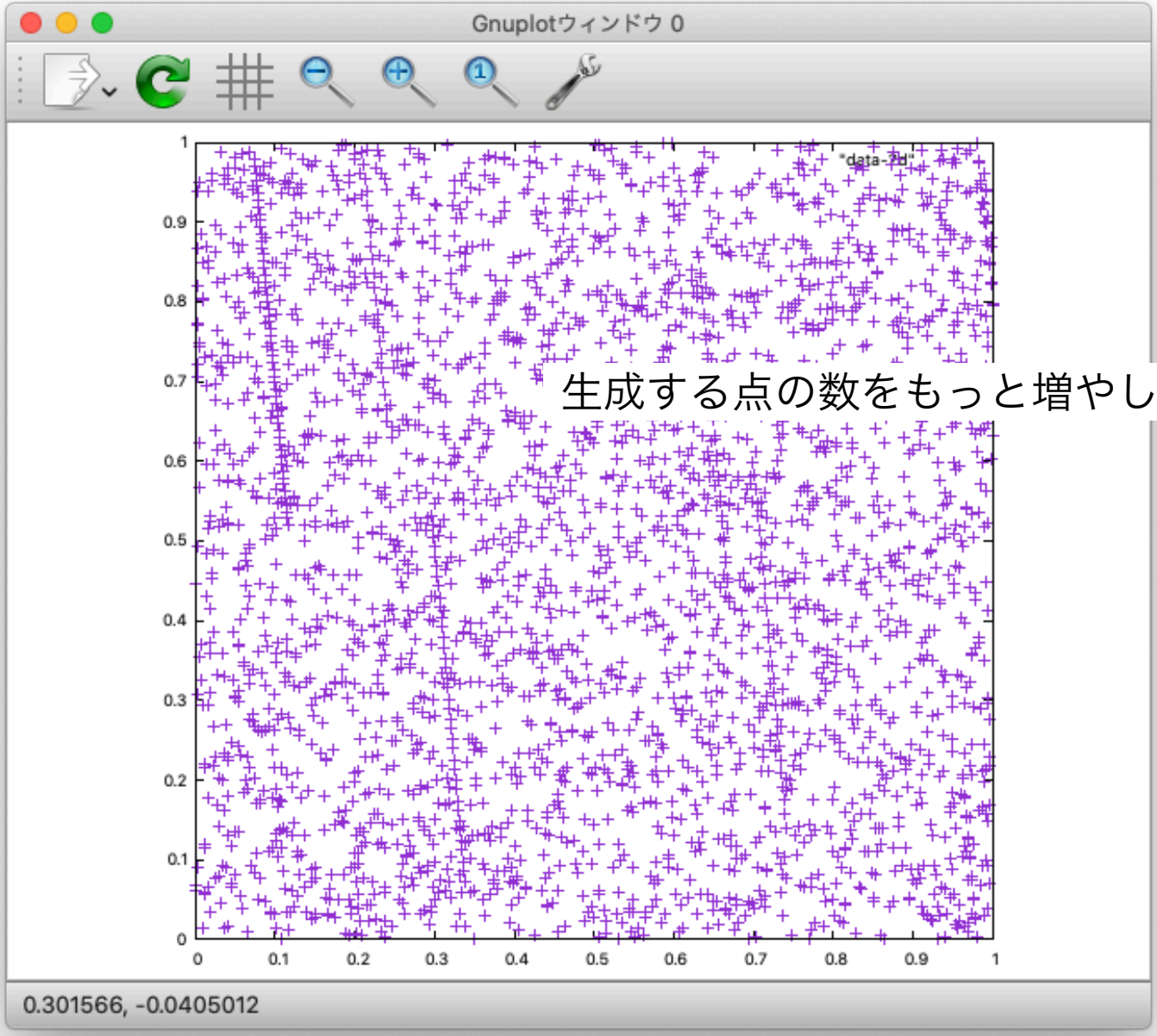
gnuplot でデータを読み込み視覚化する

```
% gnuplot
```
<— ターミナルからgnuplotを起動

```
        G N U P L O T
        Version 5.2 patchlevel 8    last modified 2019-12-01

        Copyright (C) 1986-1993, 1998, 2004, 2007-2019
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type is now 'qt'
gnuplot> set size square
gnuplot> plot "data-2d"
gnuplot>
```
<— 図の縦横比を1:1に設定

<— ファイル“data-2d” を読み込み視覚化

データの可視化



生成する点の数をもっと増やしたらどうなるか？

疑似乱数の質

- 擬似乱数列を用いて3次元空間上の点として $\{X_0, X_1, X_2\}, \{X_3, X_4, X_5\}, \{X_6, X_7, X_8\}, \ldots$ をファイルに書き出す
- 生成した疑似乱数列を gnuplot/Mathematica 等を用いて可視化する

```
% cc sample-3.c          線形合同法によるプログラムをコンパイルして実行

% ./a.out > data-3d      リダイレクションを用いて出力をファイルに保存
```

データの可視化

sample-3.c

```
#include <stdio.h>
#define IA    3877
#define IB    29573
#define IM    139968 // 周期！
int main (int argc, const char * argv[]) {

   int i, randomInt;
   double randomDouble;

   randomInt = 10;

   for(i=0; i<3000; i++){
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f ", randomDouble);          <— 出力
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f ", randomDouble);          <— 出力
     randomInt = (randomInt*IA + IB) % IM;
     randomDouble = (double)randomInt/IM;
     printf("%f\n", randomDouble);         <— 出力
   }

   return 0;
}
```

疑似乱数を3つ生成して

3次元空間上の点として表現する

x, y, z 座標を空白で区切って出力

データの可視化

gnuplot でデータを読み込み視覚化する

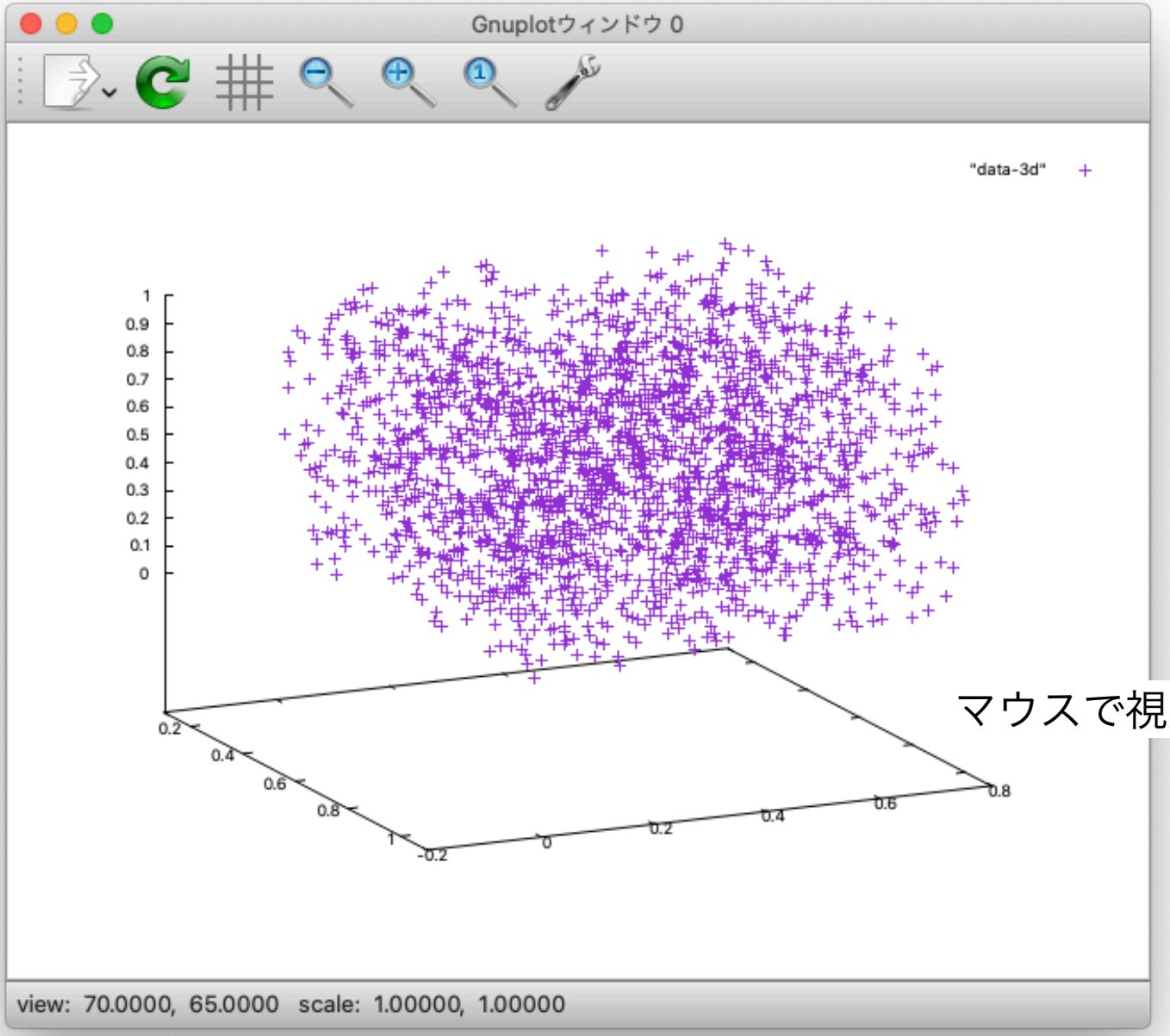
```
% gnuplot
```

<— ターミナルからgnuplotを起動

```
        G N U P L O T
        Version 5.2 patchlevel 8    last modified 2019-12-01

        Copyright (C) 1986-1993, 1998, 2004, 2007-2019
        Thomas Williams, Colin Kelley and many others

        gnuplot home:     http://www.gnuplot.info
        faq, bugs, etc:   type "help FAQ"
        immediate help:   type "help"  (plot window: hit 'h')

Terminal type is now 'qt'
gnuplot> splot "data-3d"
gnuplot>
```

<— ファイル“data-3d” を読み込み splot で視覚化

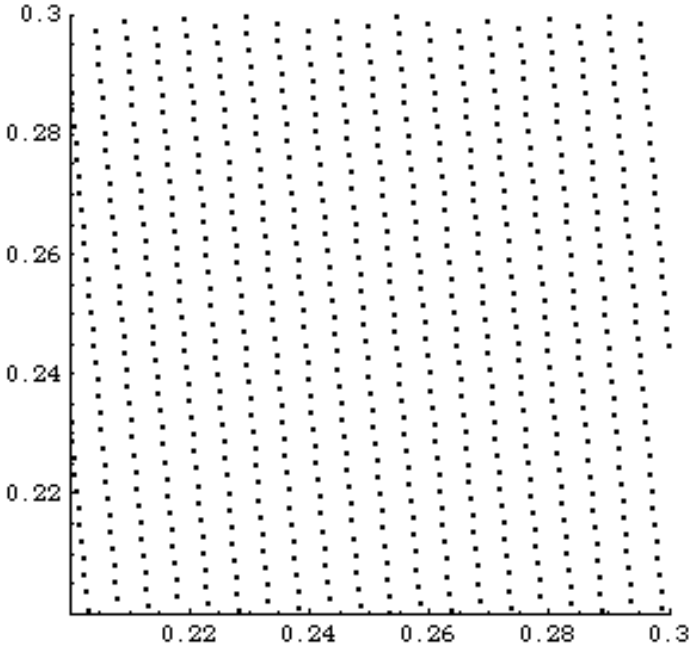
データの可視化



マウスで視点を変えてみる！

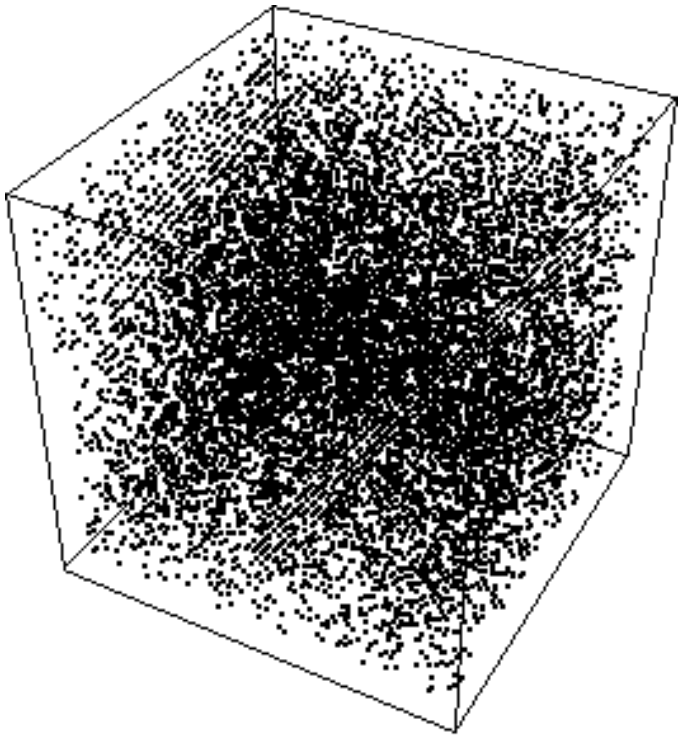
Mathematica による視覚化

```
data = ReadList[“data-2d”, {Real, Real}];
ListPlot[data, AspectRatio -> 1, PlotRange->{{0.2,0.3},{0.2,0.3}}]
```



拡大すると点 (Xn, Xn+1) が
規則的に配置

点 (Xn, Xn+1, Xn+2) が規則的に配置？



線形合同法は奨励できない

```
data = ReadList[“data-3d”, {Real, Real, Real}];
seqPoint = Map[Point, data];
g = Show[Graphics3D[seqPoint]]
```

rand 関数

- 多くの C 言語処理系で実装される rand 関数は線形合同法により擬似乱数を生成

sample-rand.c

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

int main(void)
{
  unsigned seed;
  int randomInt, i;
  double randomDouble;

  seed = (unsigned)time(NULL);

  srand(seed);
  printf("RAND_MAX is %d\n", RAND_MAX);
  printf("Seed is %d\n", seed);

  for(i=0; i<3000; i++){
    randomInt = rand();
    randomDouble = randomInt/(RAND_MAX + 1.0);
    printf("%d, %f\n", randomInt, randomDouble);
  }

  return 0;
}
```

void srand(unsigned)
乱数の種（初期値）を指定

int rand(void)
擬似乱数を生成

rand 関数は奨励できない

メルセンヌ・ツイスタ

- より高品質な擬似乱数を生成する Mersenne Twister

sample-MT.c

void init_genrand(long)
乱数の種（初期値）を指定

double genrand_real2(void)
擬似乱数 [0, 1) を生成

両関数とも外部ファイル
ran-MT.c で定義されている

乱数の種を決めて初期化することに注意！

```c
#include <stdio.h>
#include <time.h>

extern void init_genrand(long);
extern double genrand_real2(void);

int main(void)
{
  long seed;
  int i;
  double rand;

  seed = (long)time(NULL); // seed の設定

  init_genrand(seed); // seed で初期化

  for(i=0; i<3000; i++){
    rand = genrand_real2();
    printf("%.20f\n", rand);
  }

  return 0;
}
```

メルセンヌ・ツイスタ

メルセンヌ・ツイスタの本体を取得

```
% cc sample-MT.c ran-MT.c
```
2つのファイルをコンパイル＋リンクする

```
% ./a.out > data-1d-MT
```
実行結果をファイルに書き出す

```
% gnuplot
```
gnuplot で視覚化

メルセンヌ・ツイスタについて詳しく知りたい場合は本家本元を参照

http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/mt.html

モンテカルロ法 Monte Carlo Method

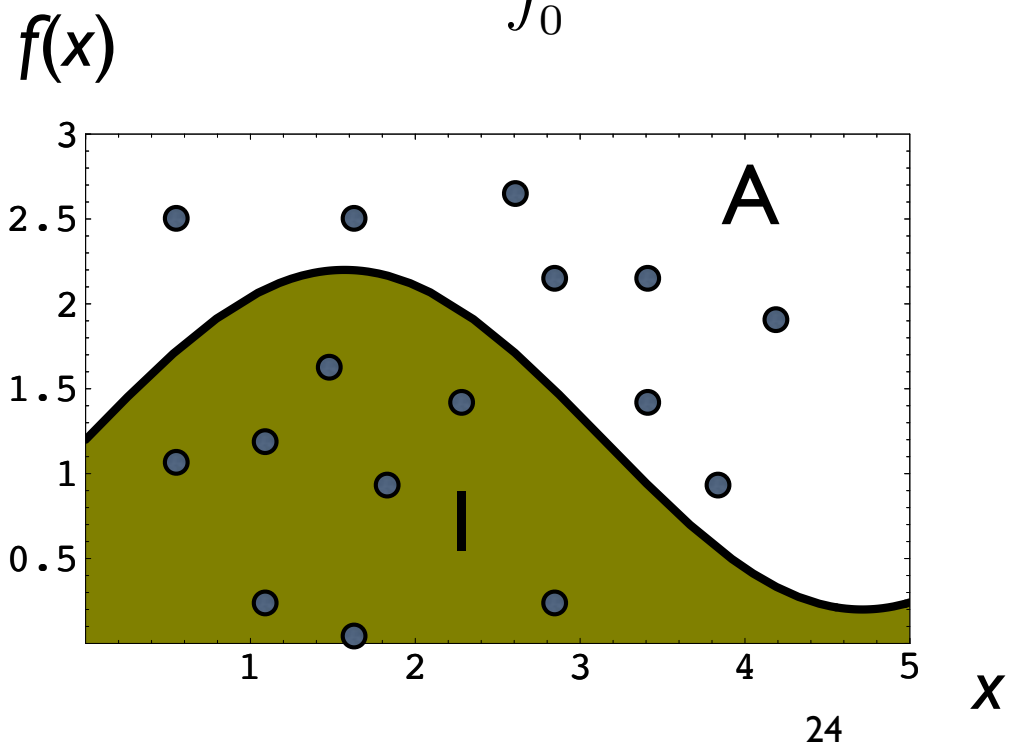
- 乱数を用いて数値計算や確率論的シミュレーションを行う手法
- 数値解析（モンテカルロ積分）や統計学、確率モデルなどで用いられる
- 一様乱数 U[0, 1) から様々な乱数を生成できる

モンテカルロ積分

- 乱数を用いた $f(x)$ の積分の数値計算

領域 A ($0 \le x \le 5, 0 \le y \le 3$) でランダムな点をとる: x = 5 U[0, 1), y = 3 U[0, 1)
積分 I は、ランダムな点が曲線 $f(x)$ の下側に落ちる割合 P と領域 A の面積 5 x 3 の積で与えられる（と予想される）

$$I = \int_0^5 f(x)dx = A \times P = A \times \frac{n_I}{n_{total}}$$



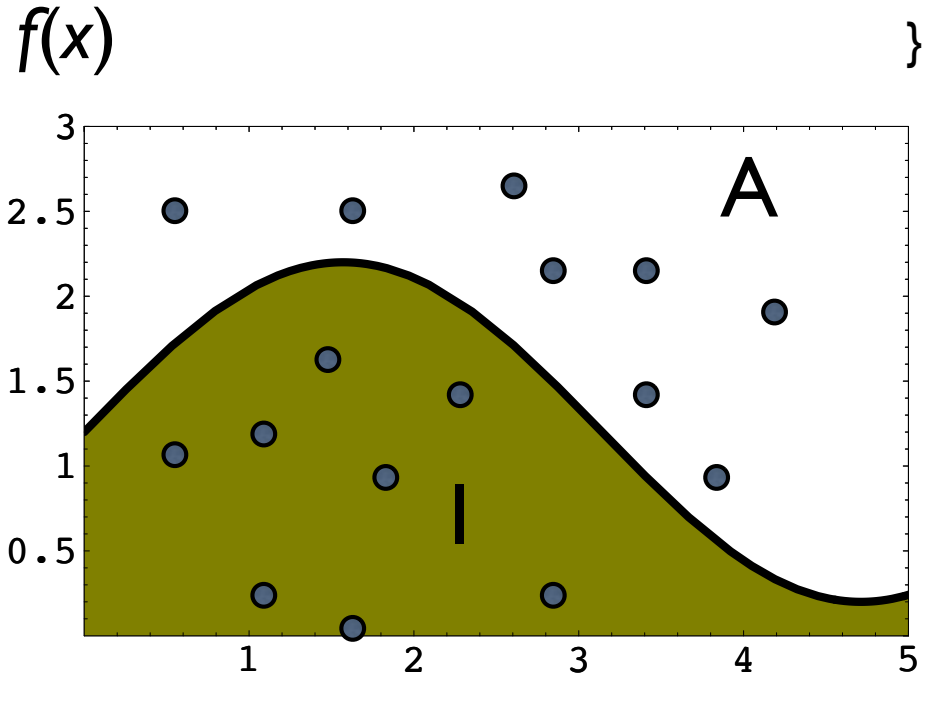
n_I : 領域 I に落ちた点の数
n_{total} : 点の総数（十分大きく取る）

領域 A でランダムな点をとるとは、領域 A 内のどの場所にも同じ確からしさで点を打つこと。

モンテカルロ積分

- アルゴリズム

```
int count = 0; // 領域 I に落ちた点を数える変数
for (点の数だけ) {
    x = U[0, 1)*5;
    y = U[0, 1)*3; // 領域 A に点を打つ

    f = myFunc(x);
    if( y <= f ) count++;
}
```



$$\int_0^5 f(x)dx \approx A \times \frac{\mathrm{count}}{N}$$

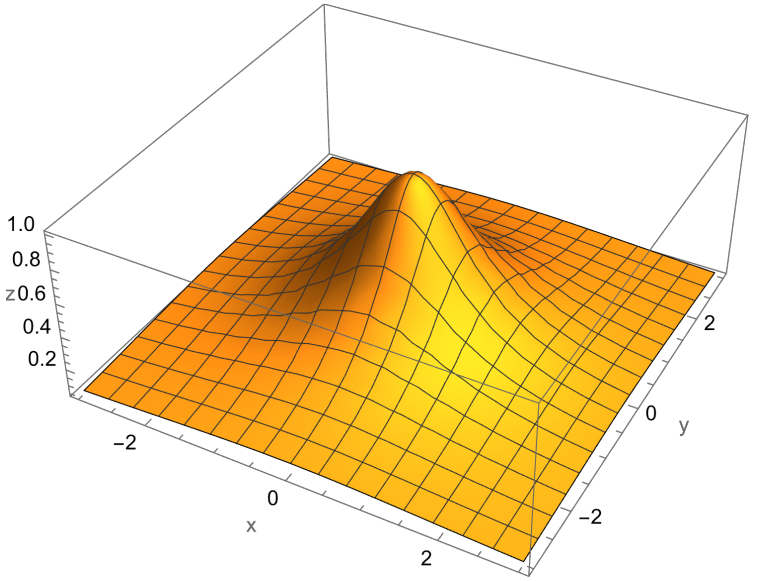
積分値（面積）は
矩形面積 A x 領域 I に落ちた点の割合

モンテカルロ積分の用途

- 関数 $z = f(x, y)$ の下側の体積もモンテカルロ積分で計算できる

$$V = \iint_{\Omega} f(x, y) dx dy \qquad \Omega : -3 \leq x, y \leq 3$$

$$z = f(x, y) = \frac{1}{1 + x^2 + y^2}$$



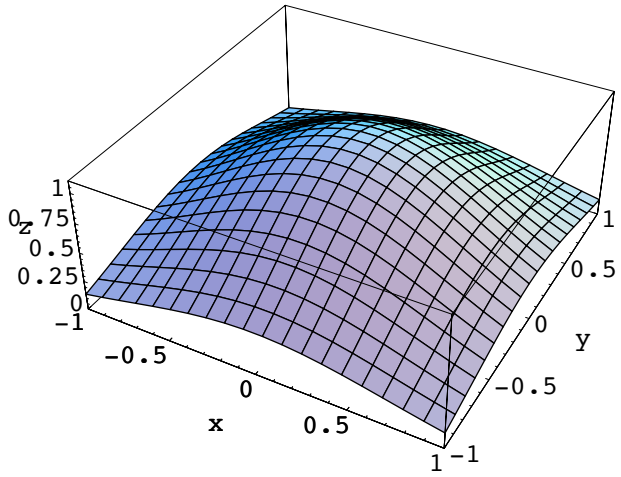
モンテカルロ積分の用途

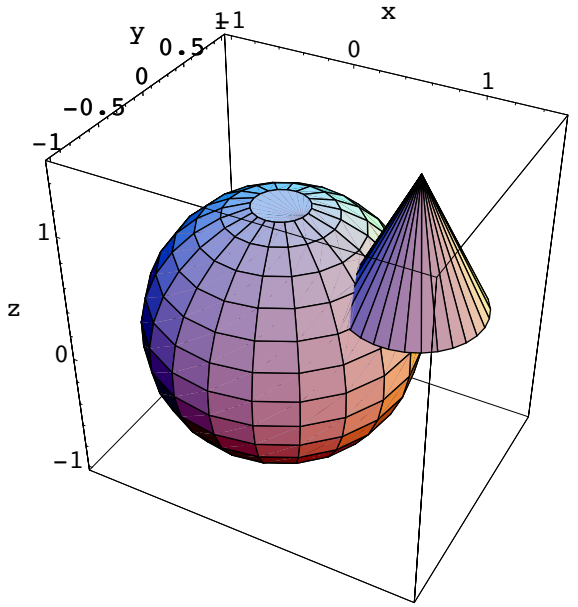
- 重積分など被積分関数 f の評価や積分範囲の数式表現が困難な場合に用いられるモンテカルロ積分
- 擬似乱数を用いて積分値の近似値を求めることができる

$$V = \iint_{\Omega} f(x,y)dxdy$$

$$\Omega : -1.5 \leq x, y \leq 1.5$$

被積分関数と積分領域の解析表示は困難





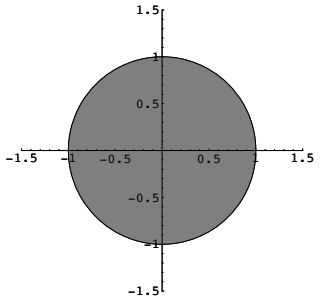
問題 1

- メルセンヌ・ツイスタを用いて [0, 1) の一様擬似乱数を N ~ 10,000 生成せよ
- [0, 1) を区間幅 0.1 で 10 個の区間に区切り、各区間に落ちた擬似乱数の数を数えてファイルに書き出せ
- 乱数が一様であれば、各区間に落ちる数の平均は N/10 と予想される。生成した擬似乱数が統計的に一様であるか、どのようにして判定したら良いか考えよ
- メルセンヌ・ツイスタにより生成した [0, 1) の一様乱数列 $\{X_1, X_2, ...,\}$ を 2次元もしくは 3次元上に描いて、視覚的に乱数の質を調べよ
- 一様乱数列において X_k と X_{k+1} は相関が無いはずである。このことをどうやって確認した良いか？X_k と X_{k+2} 、X_k と X_{k+3} ・・・も同様である。

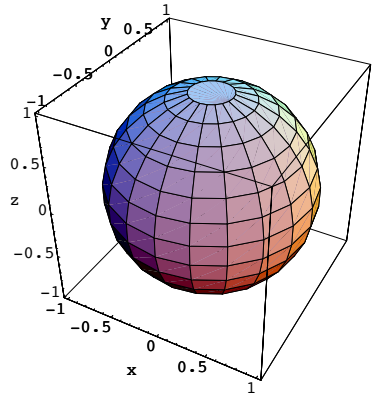
問題 2

- モンテカルロ法により以下の積分の近似値を求め、真の解と比較せよ

$$2\int_{-1}^{1}\sqrt{1-x^2}dx = \iint_{x^2+y^2\leq 1} dxdy$$

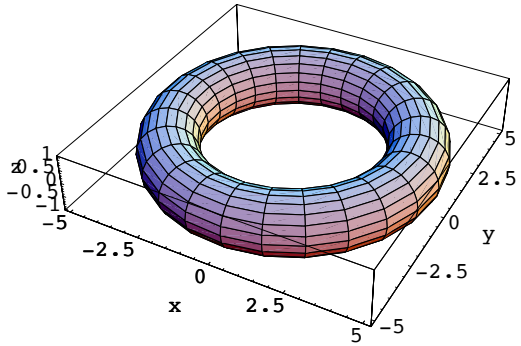


$$\iint_{x^2+y^2\leq 1}\sqrt{1-x^2-y^2}dxdy = \iiint_{x^2+y^2+z^2\leq 1} dxdydz$$

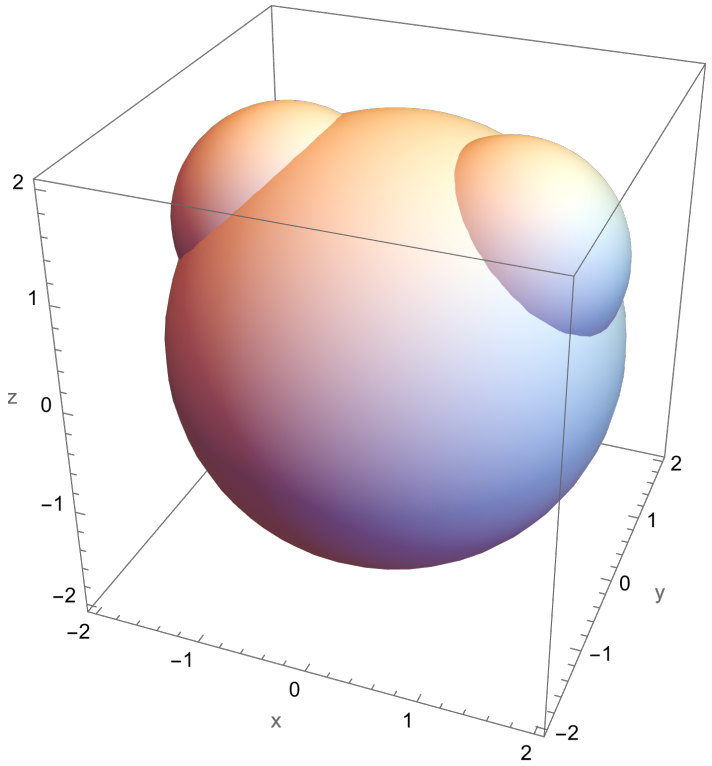


半径 1 の円を z 軸の周りで半径 4 で回転させて出来る
トーラスの体積

$$\iiint_{(\sqrt{x^2+y^2}-4)^2+z^2\leq 1} dxdydz$$



問題 2 続き

- モンテカルロ法により以下物体の体積の近似値を求めよ

原点 (0, 0, 0) を中心とする半径 2 の球、
(1, 0, 1) を中心とする半径 1 の球、
(1, 0, -1) を中心とする半径 1 の球の
合成部分の体積