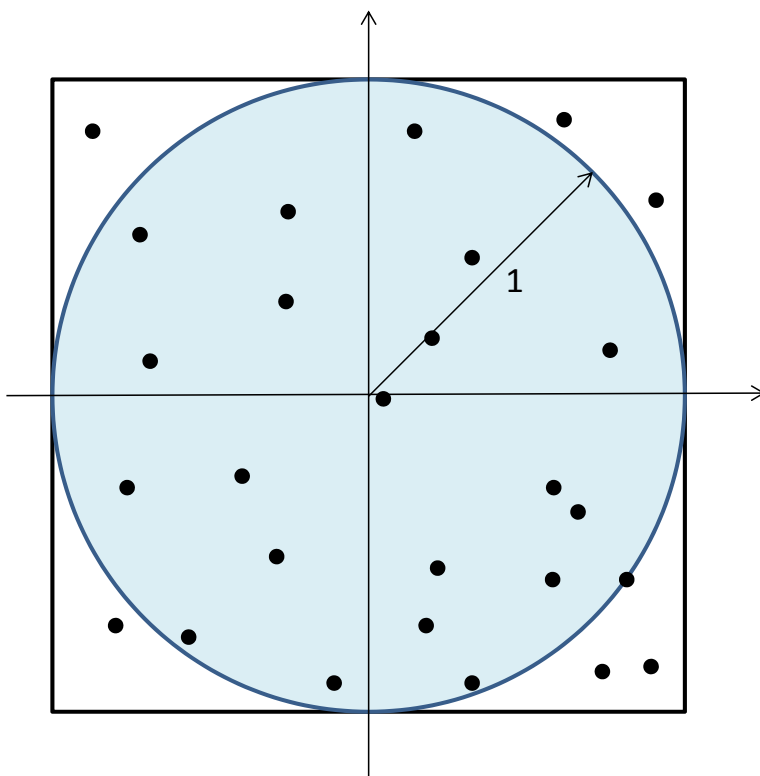


プログラミング基礎

1210 課題解説

1

モンテカルロ積分



円の面積 四角形の面積

$$\pi/4 = j/N$$

円内に含まれる点 全部の点

$$\pi = (4*j)/N$$

2

モンテカルロ積分

```
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
#define N 10000000

int main(void){
    double a,b,l,pi;
    int i,num=0;

    srand( (unsigned int)time( NULL ) );

    for(i=0; i<N; i++){
        a = rand() / (RAND_MAX + 1.0);
        b = rand() / (RAND_MAX + 1.0);
        a = 2.0 * a - 1.0;
        b = 2.0 * b - 1.0;

        if((a*a + b*b) < 1.0) num++;
    }

    pi = (double)num/N*4.0;
    printf("Pi=%lf\n", pi);
}
```

rand(),srand()

time()

rand():
[0,RAND_MAX] の擬似乱数を発生させる

srand(unsigned seed):
擬似乱数の発生系列を変更

[0,1) の乱数

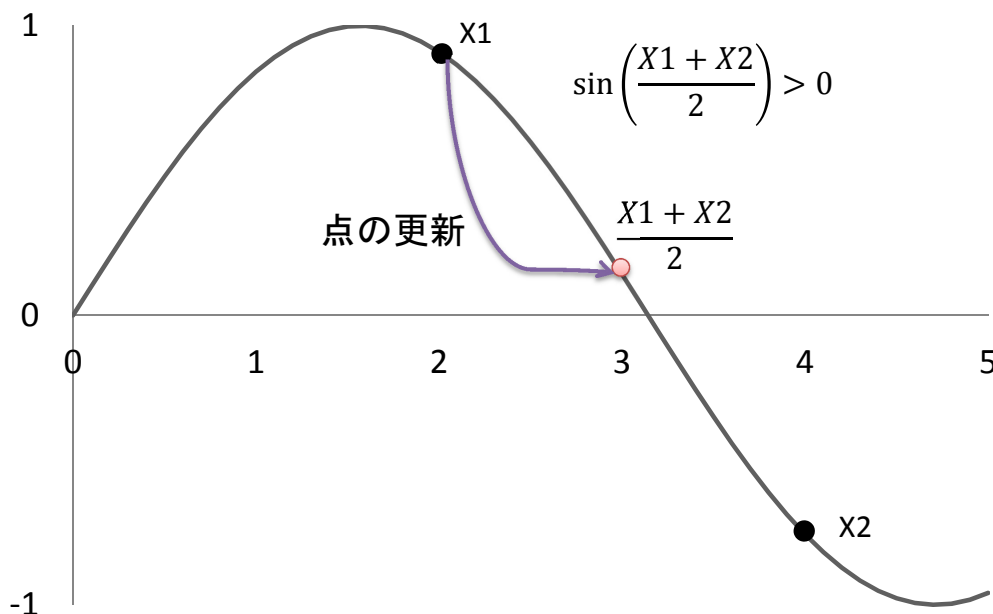
-1~1 の間の乱数a,bを発生させる

(0,0)と(a,b)の間の距離の自乗が1未満ならnumを1増やす

πの計算

3

二分法



1. $f(x_1)$ と $f(x_2)$ とで符号が異なるような区間下限と区間上限を定める
2. x_1 と x_2 の中間点 x_M を求める
3. $f(x_M)$ の符号が $f(x_1)$ と同じであれば x_1 を x_M で置き換え、 $f(x_2)$ と同じであれば x_2 を x_M で置き換える
4. 2.に戻って操作を繰り返す

4

二分法

```
#include <stdio.h>
#include <math.h>
#define th 1.0e-8

int main(){
    double pi, x1, x2, m;
    int count=0;
    x1 = 2.0;
    x2 = 4.0;
    while (fabs(x1-x2) > th && count<100){
        m = (x1 + x2) / 2.0;
        if( sin(m) > 0) x1 = m;
        else x2 = m;

        count++;
    }
    pi = (x1 + x2) / 2.0;
    printf("Pi=%.15f¥n", pi);
}
```

sin(),fabs()

初期値

(x1-x2)が閾値以下になるか、
計算を100回繰り返すまで計算を続ける

1. x1とx2の midpoint を計算
2. midpoint での関数値が
正であれば, x1を midpoint の座標に更新
負であれば, x2を midpoint の座標に更新