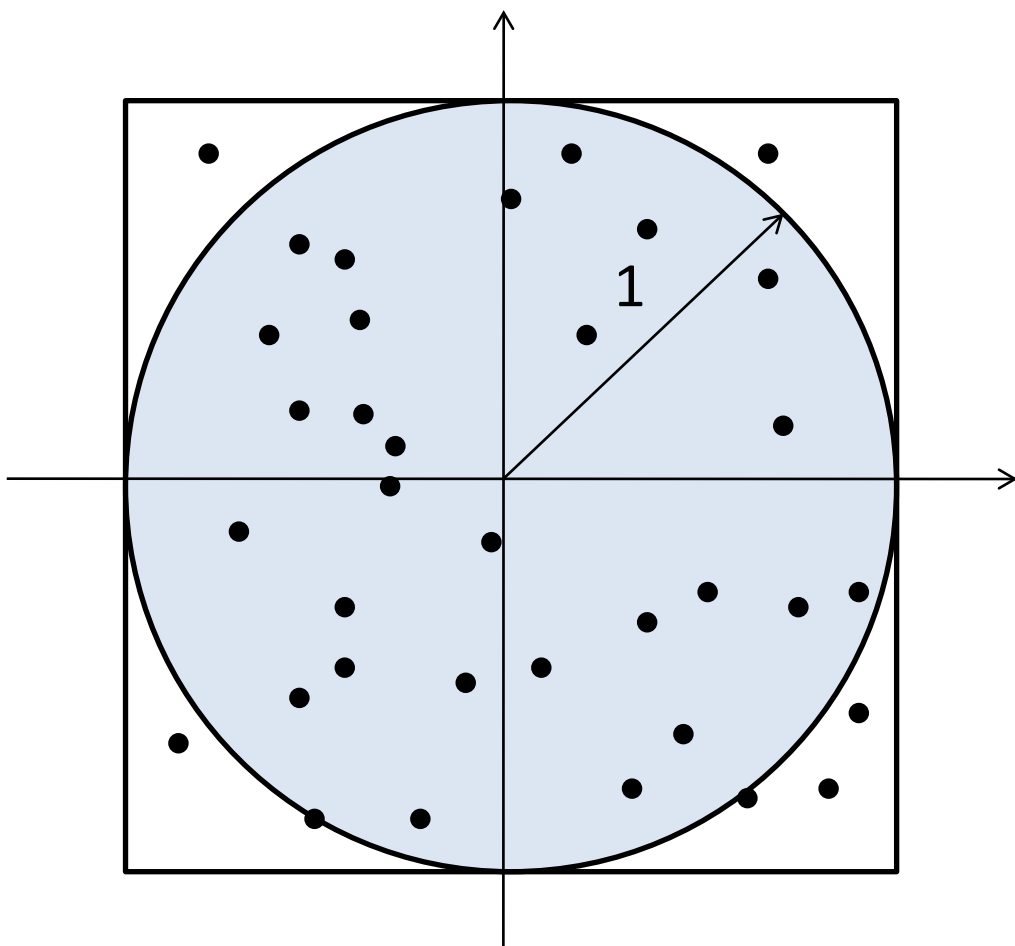


# 数値解析法

TA による解答例の解説

# 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

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

# Include と define と変数宣言

`#include <stdio.h>` → 入出力とか  
`#include <stdlib.h>` → 乱数発生とか  
`#include <math.h>` → ルートの計算とか

`#define N 1000000` → 1000000個の点を置く

```
int main(){  
    double a,b,l,pi;  
    int j=0; → 円内の点を数えるためのカウンタ
```

# πの計算

`for(int i=0; i<N; i++){` → N個の点を置きたい

`a = rand()/(RAND_MAX+1.0);`

`b = rand()/(RAND_MAX+1.0);`

`a = 2.0*a - 1.0;`

`b = 2.0*b - 1.0;`

→ a,b を-1.0~1.0の範囲に移動

`l = sqrt ((a*a) + (b*b));` → (0,0)から(a,b)の長さを計算

`if(l<1.0)`

`j++;`

`}`

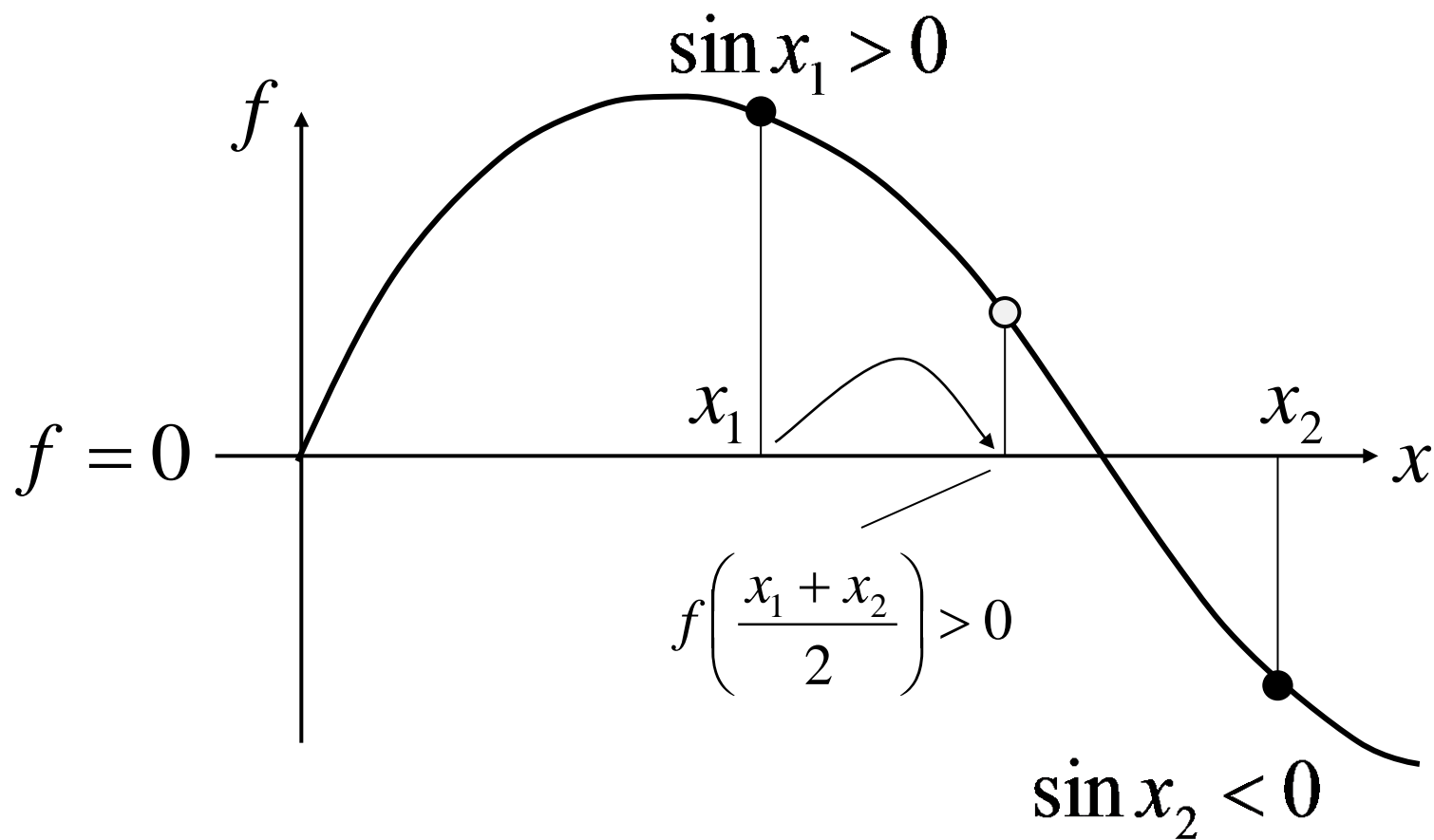
→ 点が円内であれば、jをインクリメント

`pi = 4.0*double(j)/double(N);` →  $\pi = 4*j / N$

`printf("π = %lf", pi);`

`}`


# 二分法



# Includeとか変数とか

#include <stdio.h>  printfに必要

#include <math.h>  sinの計算に必要

#define N 100  計算を100回行う

```
int main(){
```

double pi, x1, x2;  変数の宣言

```
pi = 0.0;
```

```
x1 = 2.0;
```

```
x2 = 4.0;
```

 変数の初期化

# 二分法の計算

```
for(int i=0; i<N; i++){  $\longrightarrow$  N回計算を行う
```

```
    if( sin((x1+x2)/2.0) > 0)
```

```
        x1 = (x1 + x2) / 2.0;  $\longrightarrow$  x1を更新
```

```
    else
```

```
        x2 = (x1 + x2) / 2.0;  $\longrightarrow$  x2を更新
```

```
}
```

```
pi = (x1 + x2) / 2.0;
```

```
printf("π = %lf", pi);
```

```
}
```