

シミュレーション

～宿題2: Buffon's needle problem～

e055717 金城佑典

2008/12/23

目次

1	Buffon's needle problem	3
1.1	証明	3
1.2	円周率の求め方	4
2	シミュレーションプログラム	5
2.1	ヘッダファイル (buffon.h)	5
2.2	シミュレーション本体 (buffon.c)	6
2.3	実行結果	8

1 Buffon's needle problem

ビュッフォンの針の問題 (Buffon's needle problem) はモンテカルロ法 (Monte Carlo method, MC) (乱数を用いて行なうシミュレーションの総称) の一種で、円周率 π を求める問題である。

「一定の間隔 d で並ぶ平行線の上に長さ $l (l < d)$ の針を投げたとき、針と平行線が交差する確率は $\frac{2l}{d\pi}$ である。」というもので、これを用いて円周率 π を予測する事ができる。

1.1 証明

針と平行線が交差する場合としない場合を図示する。

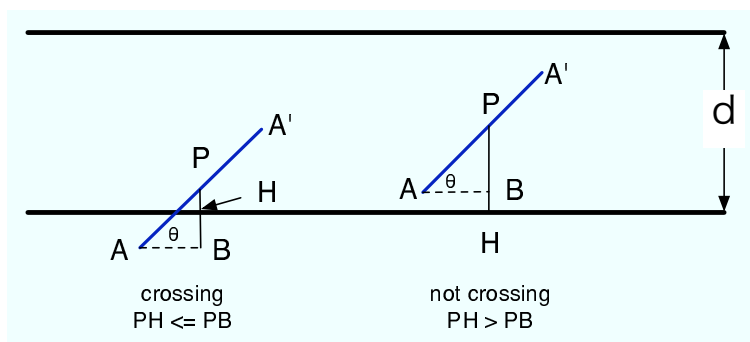


図 1 cross or not cross

- 平行線の間隔を d とする
- 針 AA' の長さを l とする (問題の定義より、 $l < d$)
- 針の midpoint P から平行線へ下ろした垂線の足を H とする
- 線分 PH の長さを x とする
- A から線分 PH へ下ろした垂線の足を B とする
- $\angle PAB = \theta$ とする

ここで、変域は

- $0 \leq x \leq \frac{d}{2}$
- $0 \leq \theta < \pi$

針が平行線と交差する条件は $PH \leq PB$ である。

$$\begin{aligned}
 PH &= x \\
 PB &= \frac{l}{2} \sin\theta \\
 PH \leq PB &\longrightarrow x \leq \frac{l}{2} \sin\theta
 \end{aligned}$$

これを図示すると下図のようになる

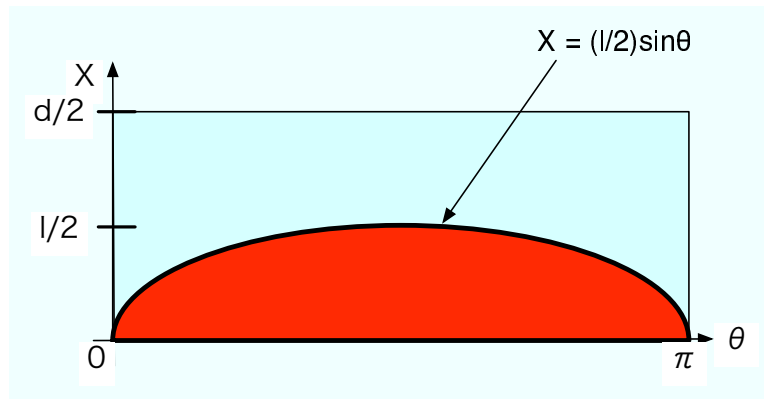


図 2 針と平行線が交わる条件

よって交差する確率 Pr は

$$\begin{aligned}
 Pr &= \frac{\text{楕円部分の面積}}{\text{長方形部分の面積}} \\
 &= \frac{\int_0^\pi \frac{l}{2} \sin\theta d\theta}{\frac{d}{2}\pi} \\
 &= \frac{l}{\frac{d}{2}\pi} \\
 &= \frac{2l}{d\pi}
 \end{aligned}$$

1.2 円周率の求め方

ビュッフォンの針の問題を用いて円周率を求める事ができる。

$$\begin{aligned}
 Pr &= \frac{2l}{d\pi} \\
 \rightarrow Pr * \pi &= \frac{2l}{d} \\
 \rightarrow \pi &= \frac{2l}{d * Pr}
 \end{aligned}$$

x と θ をランダムに生成し、針を平行線の上に投げるシミュレーションを行う。その際、針が平行線と交差した回数 (N_c) をカウントし、全体の試行回数 (n) で割れば「針が平行線と交差する確率 Pr 」が求められる。その Pr を上記の式に代入すれば円周率 π を求める事ができる。

2 シミュレーションプログラム

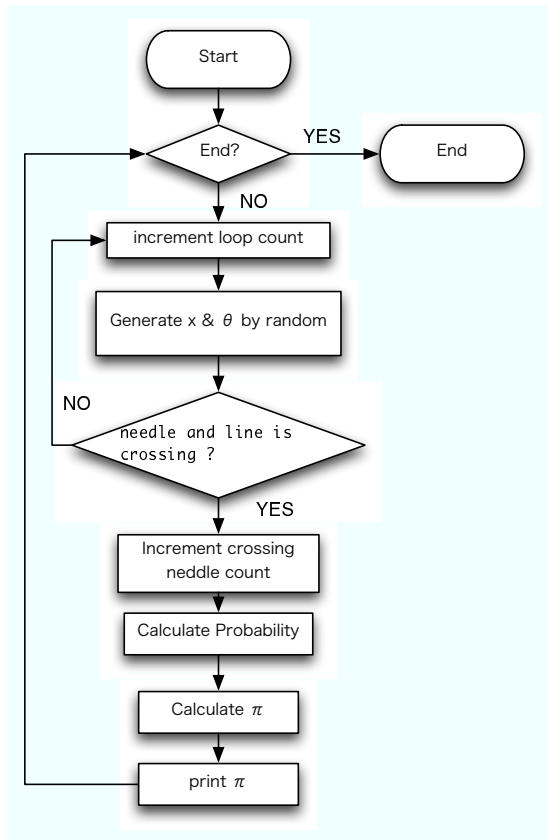


図 3 Simulation flow

終了条件 ($n < N$) に達するまでの間、以下を繰り返す。

1. 試行回数 (n) をインクリメント
2. ランダムな x と θ の生成
3. 針と平行線が交差していなければ 1 にもどる

$$x \leq \frac{l}{2} \sin(\theta)?$$

4. 交差していた針の数 (N_c) をインクリメント
5. 交差確率 (Pr) の計算

$$Pr = \frac{NC}{c}$$

6. 円周率 (π) の計算

$$\pi = \frac{2l}{d * Pr}$$

7. 円周率 (π) の出力
8. 1 にもどる

2.1 ヘッダファイル (buffon.h)

2.1.1 ソースコード

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

```
#include <getopt.h> //実行時オプションの判定
```

```
#include <time.h>//rand用 seed
```

```
#define DEFAULT_NUM 1000
```

```
#define DEFAULT_DISTANCE 2.0
```

```
#define DEFAULT_NEEDLE 1.5
```

```
#define PI_MAX 6.0
```

```
#define THETA_MAX 180.0
```

```
void usage(char *myname);
```

2.2 シミュレーション本体 (buffon.c)

2.2.1 ソースコード

```
#include "buffon.h"
```

```
int main(int argc, char **argv){
```

```
    srand((unsigned)time(NULL));
```

```
    /*Initialize*/
```

```
    double
```

```
        d = DEFAULT_DISTANCE, // distance that between line and line
```

```
        l = DEFAULT_NEEDLE, // length of needle
```

```
        x = 0.0f, // distance that between needle and line
```

```
        theta = 0.0f, // corner that needle and line make
```

```
        Pr = 0.0f, // probability of needle cross line
```

```
        pi = (rand() * PI_MAX) / (RAND_MAX); //pi(first random)
```

```
    int
```

```
        N = DEFAULT_NUM, //MAX Num
```

```
        n = 0, //counter
```

```
        Nc = 0; // number of needle cross line
```

```
    //Read Exec options (実行オプションの読み込み)
```

```
    int option;
```

```
    while( (option = getopt(argc, argv, "n:d:l:h")) != -1 ){
```

```
        switch(option){
```

```
            case 'n':
```

```
                N = atoi(optarg);
```

```

        break;
    case 'd':
        d = atoi(optarg);
        break;
    case 'l':
        l = atoi(optarg);
        break;
    case 'h':
    default:
        usage(argv[0]);
        return 0;
        break;
    }
}

if(l>d){
    printf("# Warning!! Needle length must be less than Line distance!!\n");
    d = DEFAULT_DISTANCE;
    l = DEFAULT_NEEDLE;
}

printf("# Loop Number %d\n",N);
printf("# Line Dinstance %f\n",d);
printf("# Needle Length %f\n",l);

/*Start solve Buffon's Needle probrem*/
printf("# n Probability pi\n");
for(n=1;n<N;n++){
    // Generate random x and theta number
    //0 <= x <= Line distance/2
    x = ((rand()*(d/2))/(RAND_MAX));

    //0 <= theta <= 180 [deg]
    theta = (rand() * THETA_MAX) / (RAND_MAX);

    // convert theta to radian
    //(0 <= theta <= pi [rad])
    theta = (theta / 180.0) * pi;

    //Needle is crossing line?

```

```

    if(!(x <= (1/2)*sin(theta))) continue;

    // number of needle cross line
    Nc++;//Needle count ++

    // probability of needle cross line
    Pr = ((double)Nc / (double)n);

    // it is trouble when Pr = 0
    if(Pr == 0){printf("#trouble !! set Pr=0.01\n"); Pr = 0.01f;}

    // calculate pi
    pi = (double)((2 * l) / (d * Pr));

    //print pi
    printf("%d %f %f\n",n,Pr,pi);
}

return 0;
}

void usage(char *myname){
    printf("%s [ndlh]\n",myname);
    printf("\t -n <Loop Number>:Loop Number (default:10000)\n");
    printf("\t -d <Line distance>:Line distance (default:2.0)\n");
    printf("\t -l <Needle length>:Needle length (default:1.5)\n");
    printf("\t -h :show this message\n");
}

```

2.3 実行結果

```

#Loop Number 1000
#Line Dinstance 2.000000
#Needle Length 1.500000
#n Probability pi
1 1.000000 2.283659
6 0.333333 1.500000
(中略)

```


992 0.483871 3.093946
 999 0.481481 3.100000

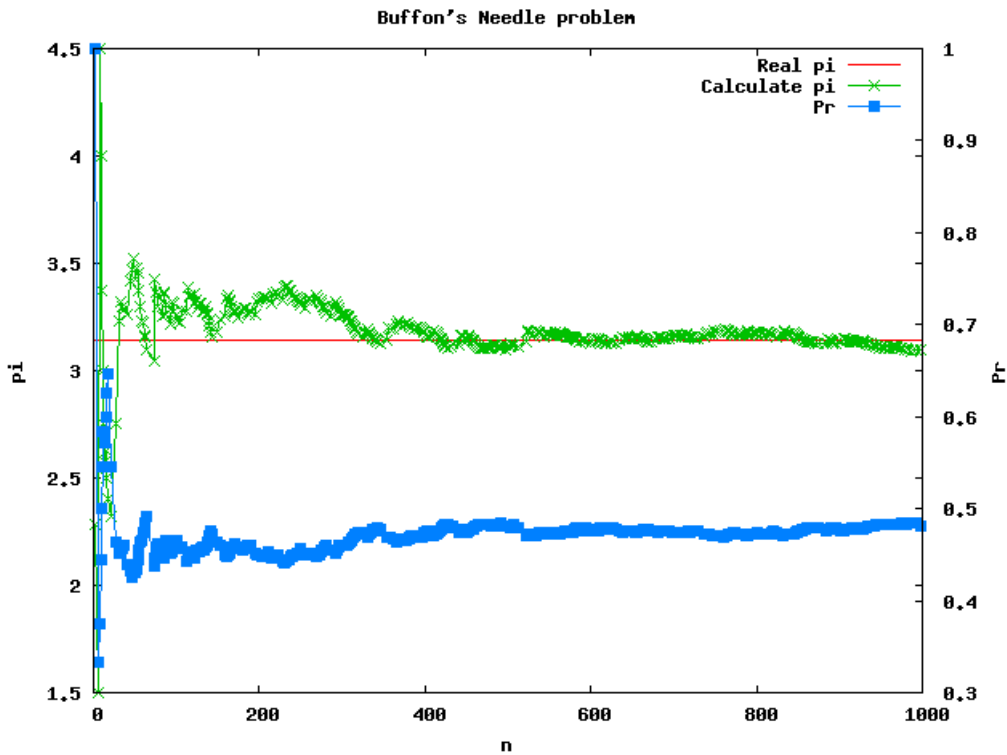


図4 実行結果 (上) 円周率 (π) (下) Probability(Pr)

最後に、このプログラムを複数回実行した際の pi の予測値を下表にまとめる。

表1 複数回実行した場合の結果

N	反復回数	π
1000	1	6.723101
10000	1	6.327215
1000	100	3.743795
10000	100	3.484913

初期値がランダムなので N を増やしても初期値によっては意味がなくなってしまうことがわかる、よってシミュレーションを行う場合は複数回行った平均を取るのがよい。

参考文献

- [1] Buffon's Needle
http://www.ie.u-ryukyu.ac.jp/~e035736/C_html/buffon_needle.html
- [2] コロキウム室 (ビュッフォンの針の実験)
<http://www.junko-k.com/cthema/38buffon.htm>
- [3] Buffon's Needle
<http://www.mste.uiuc.edu/reese/buffon/buffon.html>