

2006年6月15日(木) 実施

モンテカルロ法

中性子が物質の中を動き回る様子を探るために、John von Neumann により考案されたモンテカルロ法は、乱数を用いたシミュレーションの手法として、確率的に起こる事象等、解析的には解けない問題等に適用される。

例題 1

$0 \leq x \leq 1$, $0 \leq y \leq 1$ の範囲で乱数を発生させて、分布を見る。C 言語では、`int rand(void)` によって、 $0 \sim \text{RAND_MAX}$ の範囲の整数の疑似乱数が返される。ここで、`RAND_MAX` は `stdlib.h` の中でマクロ定義されている 32767 以上の整数である。C 言語のプログラムを翻訳編集した後、複数回実行してみる。

ex7-1.c

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

int main(void)
{
    int i;
    FILE *output;

    output=fopen("random.data", "w");

    for (i=0;i<100;i++) {
        fprintf(output, "%f %f\n", (double)rand()/RAND_MAX,
                (double)rand()/RAND_MAX);
    }

    fclose(output);

    return 0;
}
```

例題 2

例題 1 と同様に $0 \leq x \leq 1$, $0 \leq y \leq 1$ の範囲で乱数を発生させて、分布を見る。C 言語では、`void srand(unsigned int seed)` によって、`rand` 関数で発生させる疑似乱数の発生系列を変更できる。そこで、乱数の種 (random seed) として、`time(NULL)` によって得られる現在時刻を与えて、実行の都度異なる乱数が得られるようにする。C 言語のプログラムを翻訳編集した後、複数回実行して、その都度分布が異なることを確認すること。

ex7-2.c

自然科学シミュレーション ノート

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

int main(void)
{
    int i;
    FILE *output;

    output=fopen("random2.data", "w");
    srand((unsigned int)time(NULL));

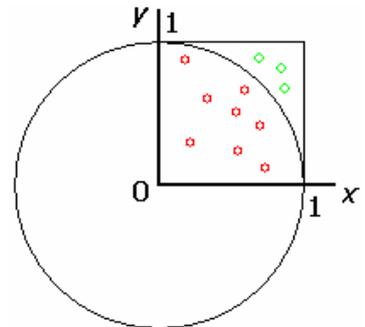
    for (i=0;i<100;i++) {
        fprintf(output, "%f %f\n", (double)rand()/RAND_MAX,
                (double)rand()/RAND_MAX);
    }

    fclose(output);

    return 0;
}
```

例題 3

例題 2 を応用して、 $0 \leq x \leq 1$, $0 \leq y \leq 1$ の範囲で乱数を発生させて、円周率 π を求める。右の図のように、半径 1 の円を考えると、その面積が π となる。その 4 分の 1 となる四分円を囲む正方形に多くの点をばら撒いて、円の内側に入った点の数 n を数え、全体の点の数を M とする。 M が充分大きな数であれば、 n/M は四分円の面積と正方形の面積との比 $\pi/4$ に近づく。



C 言語のプログラムを翻訳編集した後、複数回実行して、その都度分布と π の値とが異なることを確認すること。

ex7-3.c

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

#define M 10000

int main(void)
{
```

```

int i,n=0;
double x,y;
FILE *output_in,*output_out;

output_in=fopen("circle_in.data","w");
output_out=fopen("circle_out.data","w");
srand((unsigned int)time(NULL));

for (i=0;i<M;i++) {
    x=(double)rand()/RAND_MAX;
    y=(double)rand()/RAND_MAX;

    if (sqrt(x*x+y*y)<=1.0) {
        fprintf(output_in,"%f %f\n",x,y);
        n++;
    }
    else {
        fprintf(output_out,"%f %f\n",x,y);
    }
}

printf("π=%f",4.0*n/M);

fclose(output_in);
fclose(output_out);

return 0;
}

```

例題 4

例題 2 を応用して、 $0 \leq x \leq 1$ 、 $0 \leq y \leq 1$ の範囲で乱数を発生させて、右図の面積を求める。

C 言語のプログラムを翻訳編集した後、複数回実行して、その都度、分布と積分値とが異なることを確認すること。

ex7-4.c

```

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

```

```

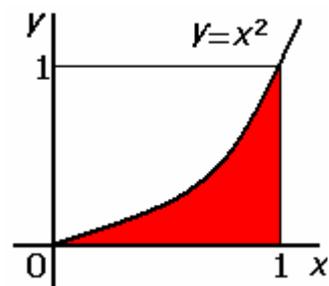
#define M 30000

```

```

int main(void)
{

```



```
int i,n=0;
double x,y;
FILE *output_in,*output_out;
output_in=fopen("para_in.data","w");
output_out=fopen("para_out.data","w");
srand((unsigned int)time(NULL));

for (i=0;i<M;i++) {
    x=(double)rand()/RAND_MAX;
    y=(double)rand()/RAND_MAX;

    if (y<=x*x) {
        fprintf(output_in,"%f %f\n",x,y);
        n++;
    }
    else {
        fprintf(output_out,"%f %f\n",x,y);
    }
}

printf("S=%f", (double)n/M);

fclose(output_in);
fclose(output_out);

return 0;
}
```

例題 5

例題 2 を応用して、 $0 \leq x \leq 1$ 、 $0 \leq y \leq 1$ 、 $0 \leq z \leq 1$ の範囲で乱数を発生させて、球の体積を求める。C 言語のプログラムを翻訳編集した後、複数回実行して、その都度、体積の値が異なることを確認すること。

ex7-5.c

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

#define M 30000

int main(void)
{
    int i,n=0;
```

自然科学シミュレーション ノート

```
double x,y,z;
srand((unsigned int)time(NULL));

for (i=0;i<M;i++) {
    x=(double)rand()/RAND_MAX;
    y=(double)rand()/RAND_MAX;
    z=(double)rand()/RAND_MAX;

    if (sqrt(x*x+y*y+z*z)<=1.0) {
        n++;
    }
}

printf("V=%f",8.0*n/M);

return 0;
}
```