# Simulate random numbers using Halton sequence

Group 7 Project No.1

# Outline

# Introduction about Halton sequence

- In statistics, Halton sequences are sequences used to generate points in space for numerical methods such as Monte Carlo simulations. Although these sequences are deterministic , they are of low discrepancy, that is, appear to be random for many purposes. They were first introduced in 1960 and are an example of a quasi-random number sequence.

# Example About Halton sequence

- Example of Halton sequence used to generate points in $(0,1) \times (0,1)$ in $R^2$ .The Halton sequence is constructed according to a deterministic method that uses coprime number as its bases. As a simple example, let us take one dimension of the Halton sequence to be based on 2 and the other on 3. To generate the sequence for 2, we start by dividing the interval $(0,1)$ in half, then in fourths, eighths, etc., which generates

$$\frac{1}{2}, \frac{1}{4}, \frac{3}{4}, \frac{1}{8}, \frac{1}{8}, \frac{5}{8}, \frac{3}{8}, \frac{7}{8}, \frac{1}{16}, \frac{9}{16} \cdots$$

# Implementation in codes in R and MATLAB

- Our group simulate n random numbers with base b between 0 and 1 using Halton sequence , the R code are as follows

```
HaltonSingle<-function(n,b){
                n0<-n
                hn<-0
                f<-1/b
                while (n0>0) {
                        n1<-floor(n0/b)
                        r<-n0-n1*b
                        hn<-hn+f*r
                        f<-f/b
                        n0<-n1
                }
                hn
                }
HaltonSingle(5,2)
```

# Implementation in codes in R and MATLAB

- if $n = 5, b = 2$

```
haltontotal<-function(n,b){
                hs<-rep(0,n)
                for(i in 1:n){
                                hs[i]<-HaltonSingle(i,b)

                }
                hs
}
haltontotal(5,2)
```

# Implementation in codes in R and MATLAB

- The MATLAB code are as follows

```
function hs = haltontotal(n,b)
hs =zeros(n,1);
for i=1:n
hs(i)=HaltonSingle(i,b)
end
```

```
function hn = haltonSingle(n,b)
n0=n;
hn=0;
f=1/b;
while(n0>0)
n1=floor(n0/b);
r=n0-n1*b;
hn=hn+f*r;
f=f/b;
n0=n1;
end
```

# Conclusion

- From R, We can get the output results

```
> Haltonsingle(5,2)

[1] 0.625

> haltontotal(5,2)

[1] 0.500 0.250 0.750 0.125 0.625
```

Also, In MATLAB, we can get the same results. Then we finish our simulation.