Molecular Monte Carlo Simulation - 1

고려대학교 화공 생명공학과
강정원

Topics

- Random number generators
- Importance Sampling and Simulating Distribution
- Markov chain
- Markov chain and Monte-Carlo Method
- (Project -1) Description of 2-D Ising Model

MC Method …

- 1953, Nicolaus Metropolis
- 50th anniversary in 2003 !
- Monte Carlo method refers any method that make use of random number

  - Simulation of natural phenomena
  - Simulation of experimental apparatus
  - Numerical analysis

1. Random Number ….

- What is random number ? Is 3 ?
  - There is no such thing as single random number
- Random number
  - A set of numbers that have nothing to do with the other numbers in the sequence
- In a uniform distribution of random numbers in the range [0,1] , every number has the same chance of turning up.
  - 0.00001 is just as likely as 0.5000

How to generate random numbers ?

- Use some chaotic system (Balls in a barrel – Lotto)
- Use a process that is inherently random
  - Radioactive decay
  - Thermal noise
  - Cosmic ray arrival
- Tables of a few million random numbers
- Hooking up a random machine to a computer.

Pseudo Random number generators

- The closest random number generator that can be obtained by computer algorithm.
- Usually a uniform distribution in the range [0,1]
- Most pseudo random number generators have two things in common
  - The use of large prime numbers
  - The use of modulo arithmetic
- Algorithm generates integers between 0 and M

$$X_n = I_n / M$$

An early example (John Von Neumann,1946)

- To generate 10 digits of integer
  - Start with one of 10 digits integers
  - Square it and take middle 10 digits from answer
  - Example) $5772156649^2 = 33317\underline{7923805949}09291$
- The sequence is appears to be random, but each number is determined from the previous → not random
- Serious problem : Small numbers (0 or 1) are lumped together, it can get itself to a short loop.
  - Example)
    - $6100^2 = 37\underline{2100}00$
    - $2100^2 = 04\underline{4100}00$
    - $4100^2 = 16\underline{8100}00$
    - $5100^2 = 65\underline{6100}00$

Linear Congruential Method

- Lehmer, 1948
- Most typical compiler-supplied so-called random number generator
- Algorithm : $$I_{n+1} = (aI_n + c)\,\mathrm{mod}(m)$$
  - *$a, c >= 0$, $m > I_0, a, c$*
- *Advantage :*
  - *Very fast*
- *Problem :*
  - *Poor choice of the constants can lead to very poor sequence*
  - *The relationship will repeat when a period no greater than m (around m/4)*
    - *Ex) C complier RAND_MAX : m = 32767*

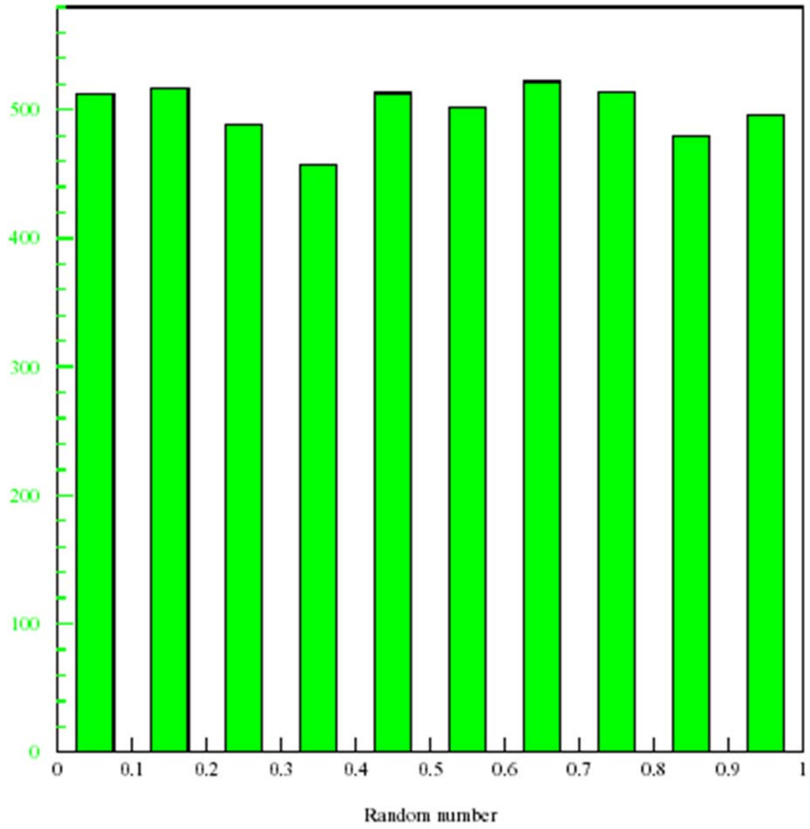
RANDU Generator

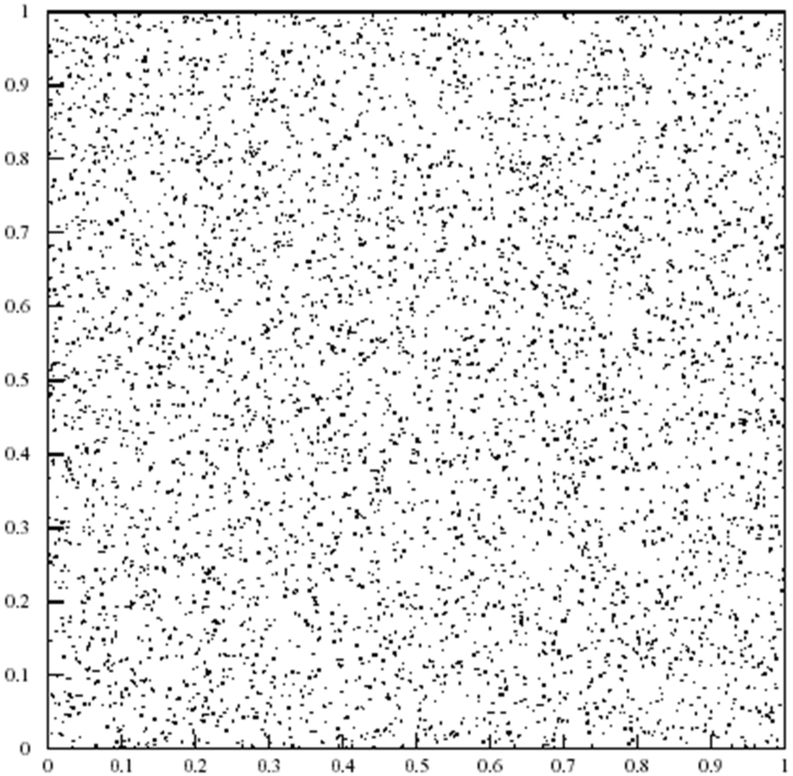
- 1960's IBM
- Algorithm

$$I_{n+1} = (65539 \times I_n) \bmod (2^{31})$$

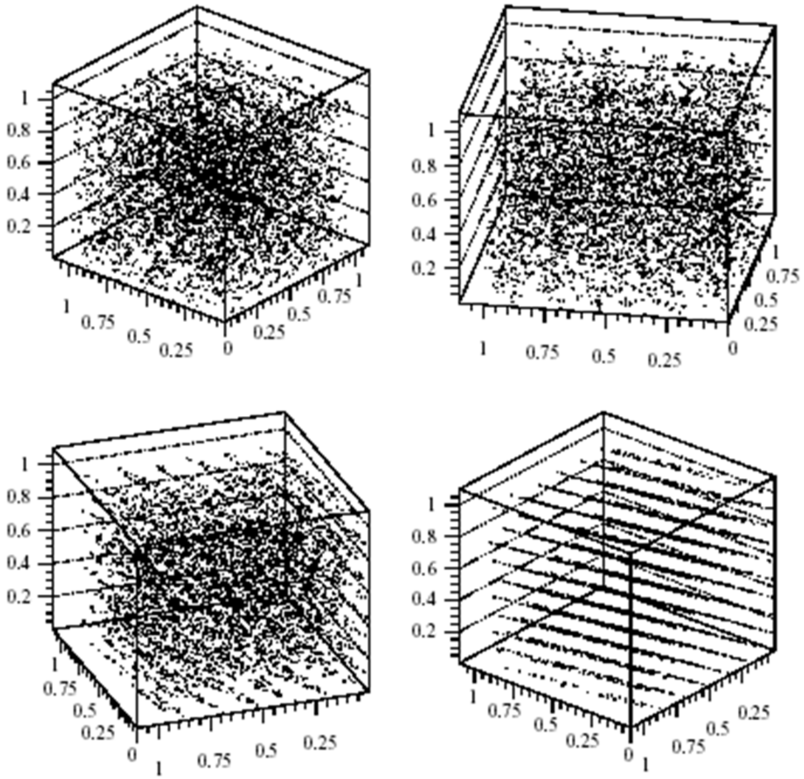
- This generator was later found to have a serious problem

1D and 2D Distribution of RANDU





3D Distribution from RANDU



Problems seen when observed at the right angle

The Marsaglia effect

- 1968, Marsaglia
- Randon numbers fall mainly in the planes

- The replacement of the multiplier from 65539 to 69069 improves performance significantly

Warning

- The authors of "Numerical Recipes" have admitted that random number generator, RAN1 and RAN2 in the first edition are "at best mediocre"

  → 평범한, 이류의

- In their second edition, these are replaced by ran0, ran1, ran2, which have much better properties

One way to improve the behavior of random number generator

$$I_n = (a \times I_{n-1} + b \times I_{n-2}) \bmod (m)$$

⟶ Has two initial seed and can have a period greater than m

The RANMAR generator

- Available in the CERN Library
  - Requires 103 initial seed
  - Period : about 10^{43}
  - This seems to be the ultimate random number generator

Warning on the use of random number generators

- Compiler optimizer is trying to remove multiple calls to random number generator

```
X = RAND(IDUM) + RAND(IDUM)
```

↓

```
X = 2.0 * RAND(IDUM)
```

```
DO 1 I = 1,100
        X = RAND(IDUM)
1 CONTINUE
```

↓

Not evaluating every steps
Evaluated only once

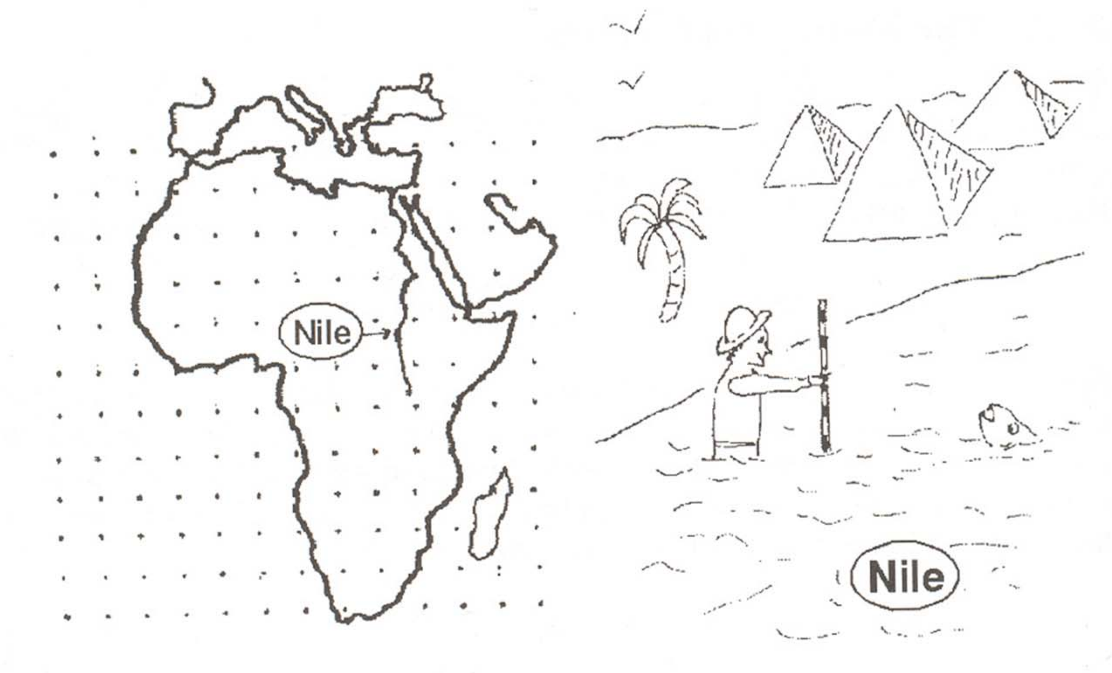
→ *<u>You have to change the dummy parameter for each calls</u>*

Homework - 1

- Find best random number generator on the web and post on the IP board

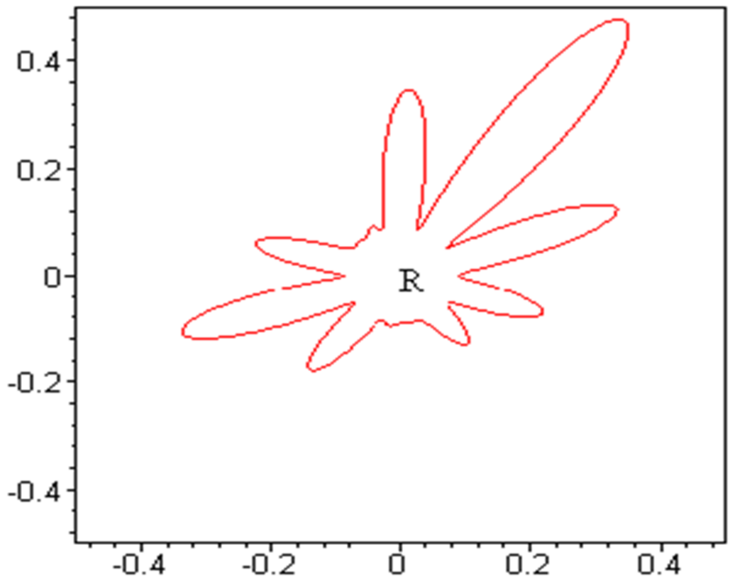
2. Importance Sampling and Simulating Distributions

- Nature of the problem …



Shape of High Dimensional Region

- Two (and Higher) dimensional shape can be complex
- How to construct weighted points in a grid that covers the region R ?



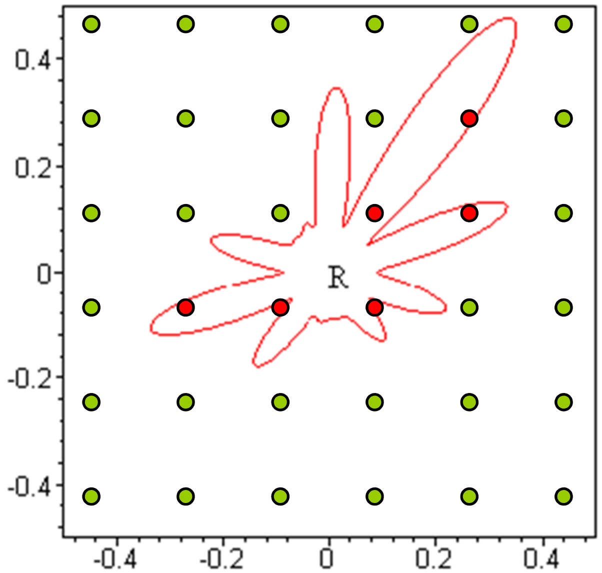
Problem :
mean-square distance from the origin

$$\langle r^2 \rangle = \frac{\int\int (x^2 + y^2)\,dxdy}{\int\int dxdy}$$

Integration over simple shape ?

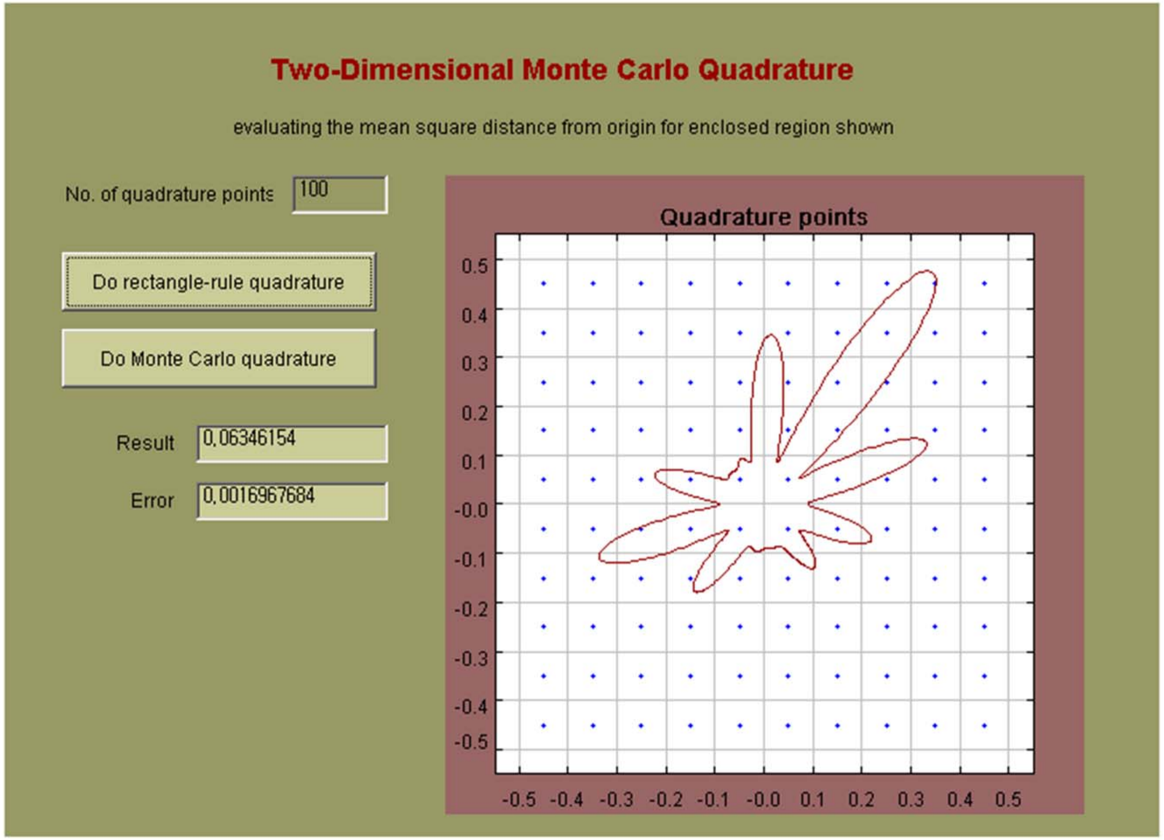
$$s = \begin{cases} 1 & \text{inside R} \\ 0 & \text{outside R} \end{cases}$$

$$\left\langle r^2 \right\rangle = \frac{\int_{-0.5}^{+0.5} dx \int_{-0.5}^{+0.5} dy (x^2 + y^2) s(x,y)}{\int_{-0.5}^{+0.5} dx \int_{-0.5}^{+0.5} dy s(x,y)}$$

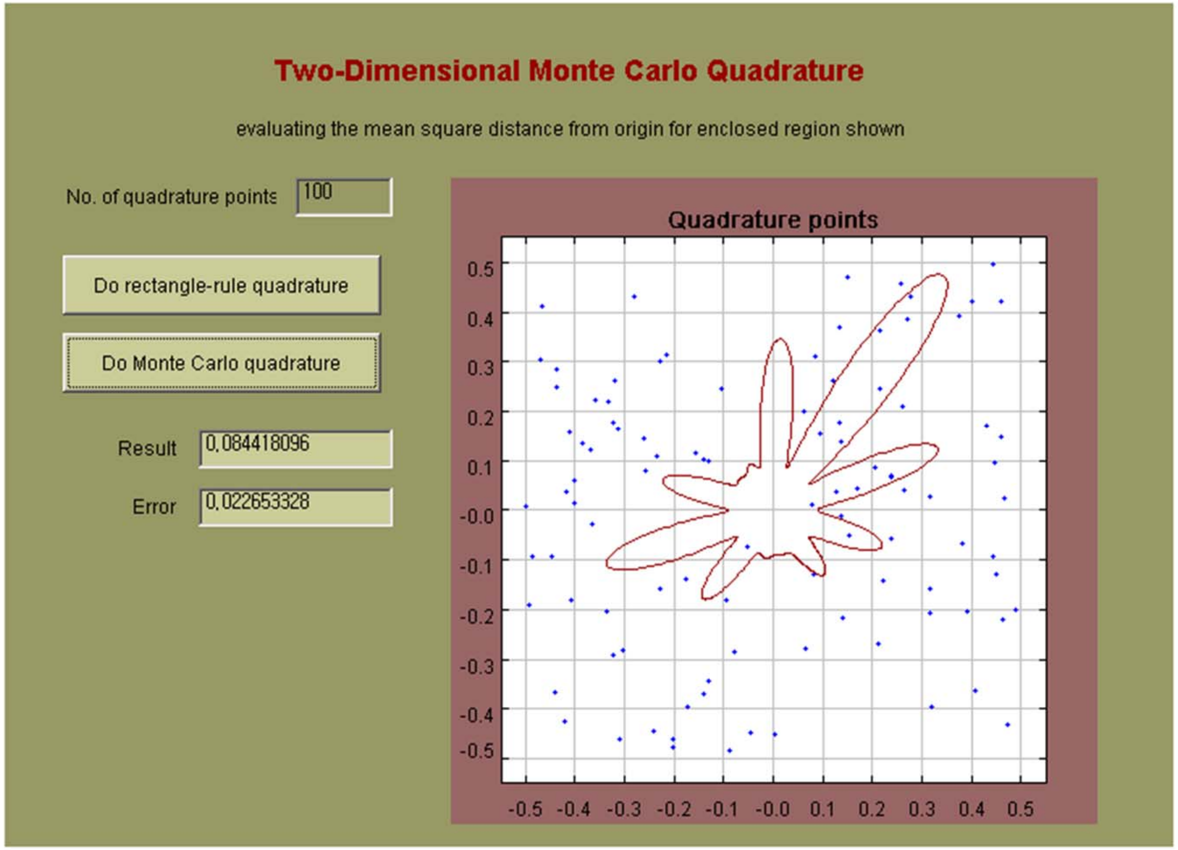


→ Grid must be fine enough !

Sample Integration



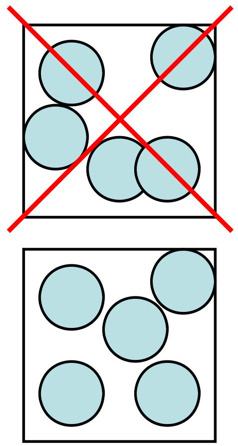
Sample Integration



Integration over simple shape ?

- Statistical mechanics integrals typically have significant contribution from miniscule regions of the integration space.

- Ex) 100 spheres at freezing fraction = 10^{-260}



Importance Sampling – Inversion Technique

- This method is only applicable for relatively simple distribution functions
  - Normalize distribution function , so that it becomes probability distribution function (PDF)
  - Integrate PDF form minimum x to an arbitary x
    - This value represents chosing a value less than x
  - Evaluate this to a uniform random number, and solve for x, resulting x will be distributed according to PDF

Inversion formula

$$\frac{\int_{x_{\min}}^{x} f(x)dx}{\int_{x_{\min}}^{x_{\max}} f(x)dx} = \lambda$$

Examples

- Evaluate x between 0 and 4 according to f(x) = x^(-1/2)

- Evaluate x between 0 and infinity according to f(x) = exp(-x)

Importance Sampling

- Return to 1-D integral example

$$I = \int_0^1 3x^2 dx$$

- A linear form is one possibility
- How to generate random points according to the distribution ?

$$\pi(x) = 2x$$