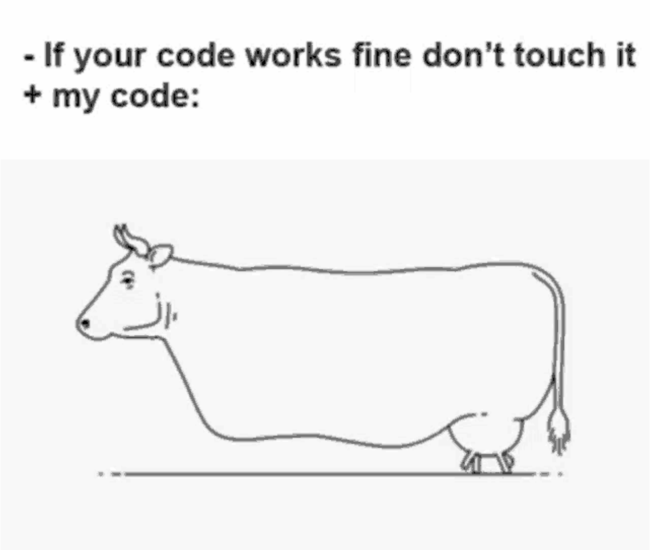
國立臺北大學 National Taipei University

(U4284) Python程式設計
Flow Control



Speaker: 吳淳硯

python™

If Loop

- If + elif + else + nested if syntax

```
if condition1:
    statement1
        ⋮
    statement1
elif condition2
    if condition21:
        statement21
            ⋮
        statement21
    else:
        statement21*
            ⋮
        statement21*
else:
    statement3
        ⋮
    statement3
```

First condition
This is executed if the 1st condition is true

Nested if part

False branch
This is executed if none of the conditions are true

While Loop & Conditional Expression

- Conditional Expression syntax

$$\text{variable} = \underbrace{\text{statement}}_{\text{True branch}} \text{ if condition else } \underbrace{\text{statement}}_{\text{False branch}}$$

- A while loop is used when you want to perform a task indefinitely, until a particular condition is met. It's a condition-controlled loop.

- While statement syntax

```
while condition:
    statement
        ⋮
    statement
else:
    statement*
        ⋮
    statement*
```

Loop body
It is executed as long as the condition is true

Else clause
It is executed if the condition become false

For Loop

- For loop syntax

```
for var in iterable:
    statement
        ⋮
    statement
else:
    statement*
        ⋮
    statement*
```

Loop body

It is executed once for each item in iterable

Else clause

It is executed if the loop terminates naturally

- Nested for loop syntax

```
for var1 in iterable1:
    statement1
        ⋮
    statement1
    for var2 in iterable2:
        statement2
            ⋮
        statement2
```

Break and Continue

- *break* statement
  - used to exit the loop immediately. It simply jumps out of the loop altogether, and the program continues after the loop.

```
x = 6
while x:
    print(x)
    x -= 1
    if x == 3:
        break
# Prints 6 5 4
```

- *continue* statement
  - skips the current iteration of a loop and continues with the next iteration.

```
x = 6
while x:
    x -= 1
    if x % 2 != 0:
        continue
    print(x)
# Prints 4 2 0
```

Handling Exception

- Try…Except…finally syntax

```
try:
    statement1
    ⋮
    statement1
except Exception as e:
    print(e)
    statement2
    ⋮
    statement2
else:
    statement3
    ⋮
    statement3
finally:
    statement4
    ⋮
    statement4
```

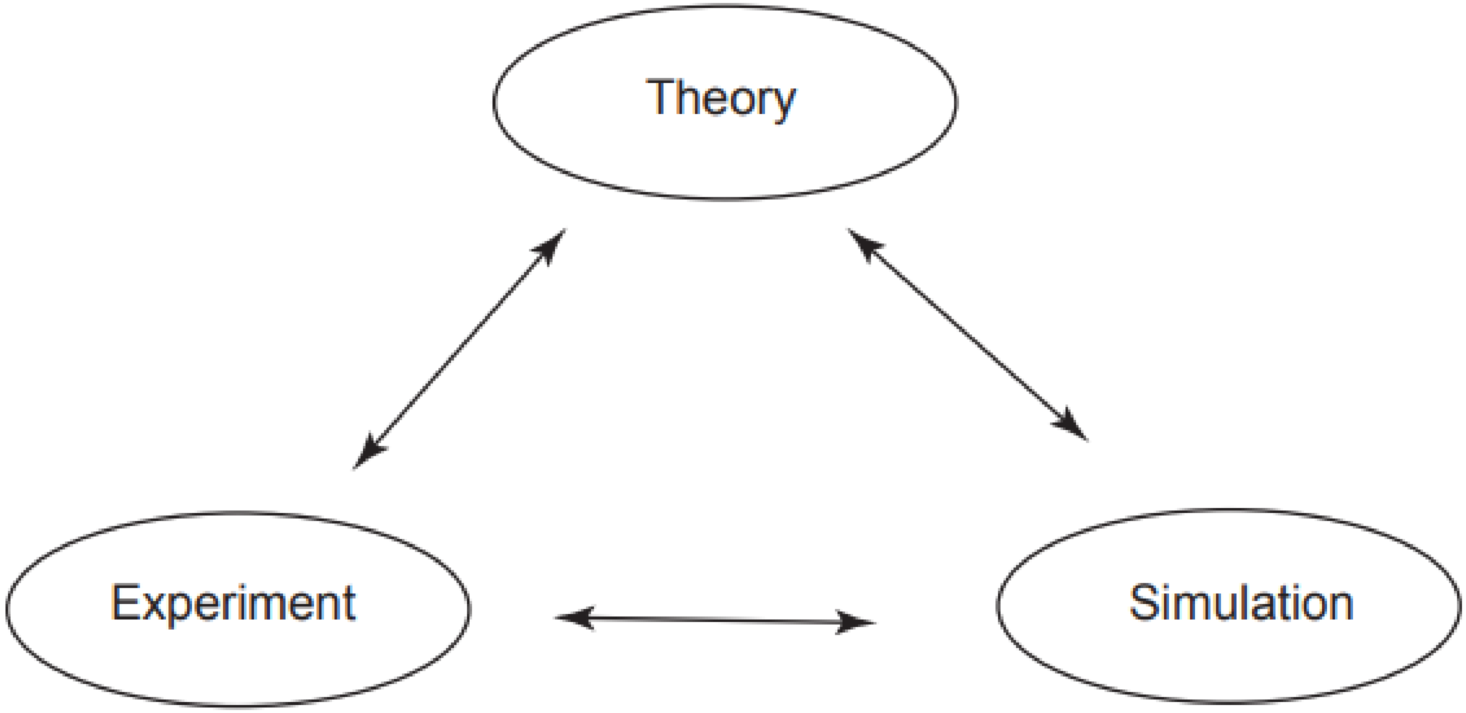
Execute this when there is an exception

Execute this only if no exceptions are raised

Always execute this

Simulation

- How do we obtain knowledge in science?



Pseudo-random numbers

- A sequence of deterministic numbers which have the same relevant statistical properties of random number.

- Random number should be
  - Uniformly distributed
  - Statistically independent
  - Reproducible

- Linear Congruential Method
  Let $a, c, m \in \mathbb{Z}$, X_0 is starting value (seed)
  $$X_{i+1} = aX_i + c \pmod{m}$$
  - a is multiplier factor
  - c is increment factor
  - m is modulus and $m > a, c, X_0$

Ref. Sheldon Ross - Simulation

Law of the Large Number (LLN)

- Strong law of large numbers
  If $X_1, X_2, \cdots$ are iid random variables with finite expectation μ, then
  $$\frac{X_1 + \cdots + X_n}{n} \to \mu \quad \text{a.e.}$$

- Proof Concept
  $$\sum_{i=1}^{n} X_i = \sum_{i=1}^{n} (X_i - Y_i) + \sum_{i=1}^{n} (Y_i - E(Y_i)) + \sum_{i=1}^{n} E(Y_i)$$
  $$P(X_n \neq Y_n \text{ i.o.}) = 0$$
  $$\frac{1}{n} \sum_{i=1}^{n} (Y_i - E(Y_i)) \to 0 \quad \text{a.e.}$$
  $$\frac{1}{n} \sum_{i=1}^{n} E(Y_i) \to 0 \quad \text{a.e.}$$

Ref. Robert Ash – Probability and Measure Theory

The Inverse Transform Method (Discrete)

- Suppose we want to generate the value of a discrete random variable X having

$$P(X = x_j) = p_j, \qquad j = 0,1,\cdots, \qquad \sum_j p_j = 1$$

- We generate a $U \sim U(0,1)$ and

$$X = \begin{cases} x_0, & U < p_0 \\ x_1, & p_0 \le U < p_0 + p_1 \\ & \vdots \\ x_j, & \displaystyle\sum_{i=0}^{j-1} p_i \le U < \sum_{i=0}^{j} p_i \\ & \vdots \end{cases}$$

$$P(X = x_j) = P\left(\sum_{i=0}^{j-1} p_i \le U < \sum_{i=0}^{j} p_i\right) = p_j$$

and so X has the desired distribution

Ref. Sheldon Ross - Simulation

The Inverse Transform Method (Continuous)

- Let $U \sim U(0,1)$. For any continuous distribution function F the random variable X define by $X = F^{-1}(U)$ has distribution F.

$$F_X(x) = P(X \leq x) = P(F^{-1}(U) \leq x)$$
$$= P\left(F\left(F^{-1}(U)\right) < F(x)\right) = P\left(U < F(x)\right) = F(x)$$

- Consider $X \sim Cauchy(0,1)$

$$F_X(x) = \int_{-\infty}^{x} \frac{1}{\pi}\left[\frac{\sigma}{(t-\mu)^2 + \sigma^2}\right] dt = \frac{1}{\pi}\int_{-\infty}^{x} \frac{1}{1 + \left(\frac{t-\mu}{\sigma}\right)^2} d\left(\frac{t-\mu}{\sigma}\right)$$

$$= \frac{\tan^{-1}\left(\frac{x-\mu}{\sigma}\right)}{\pi} + \frac{1}{2}$$

then we can generate Cauchy by uniform with the following relation

$$U = \frac{\tan^{-1}\left(\frac{x-\mu}{\sigma}\right)}{\pi} + \frac{1}{2} \Leftrightarrow X = \tan\left\{\left(U - \frac{1}{2}\right)\pi\right\}\sigma + \mu$$

Brownian Motion (BM)

- Let $(\Omega, \mathcal{F}, P)$ be a Prob. space. A Stoc. Proc. is a measurable function $X(t, \omega)$ defined on the $[0, \infty) \times \Omega$
- A Stoc. Proc. $B(t, \omega)$ is called BM if
  1. $\mathrm{P}(\omega; B(0, \omega) = 0) = 1$
  2. For any $0 \leq s < t,\ B(t) - B(s) \sim N(0, t - s)$
  3. For any $0 \leq t_1 < t_2 < \cdots < t_n$
     $B(t_1), B(t_2) - B(t_1), \cdots, B(t_n) - B(t_{n-1})$ are independent
  4. $\mathrm{P}(\omega; B(t, \omega)$ is continuous$) = 1$
- $B(t)$ is a Martingale with filtration $\mathcal{F}_s = \sigma(B(u), u \leq s)$

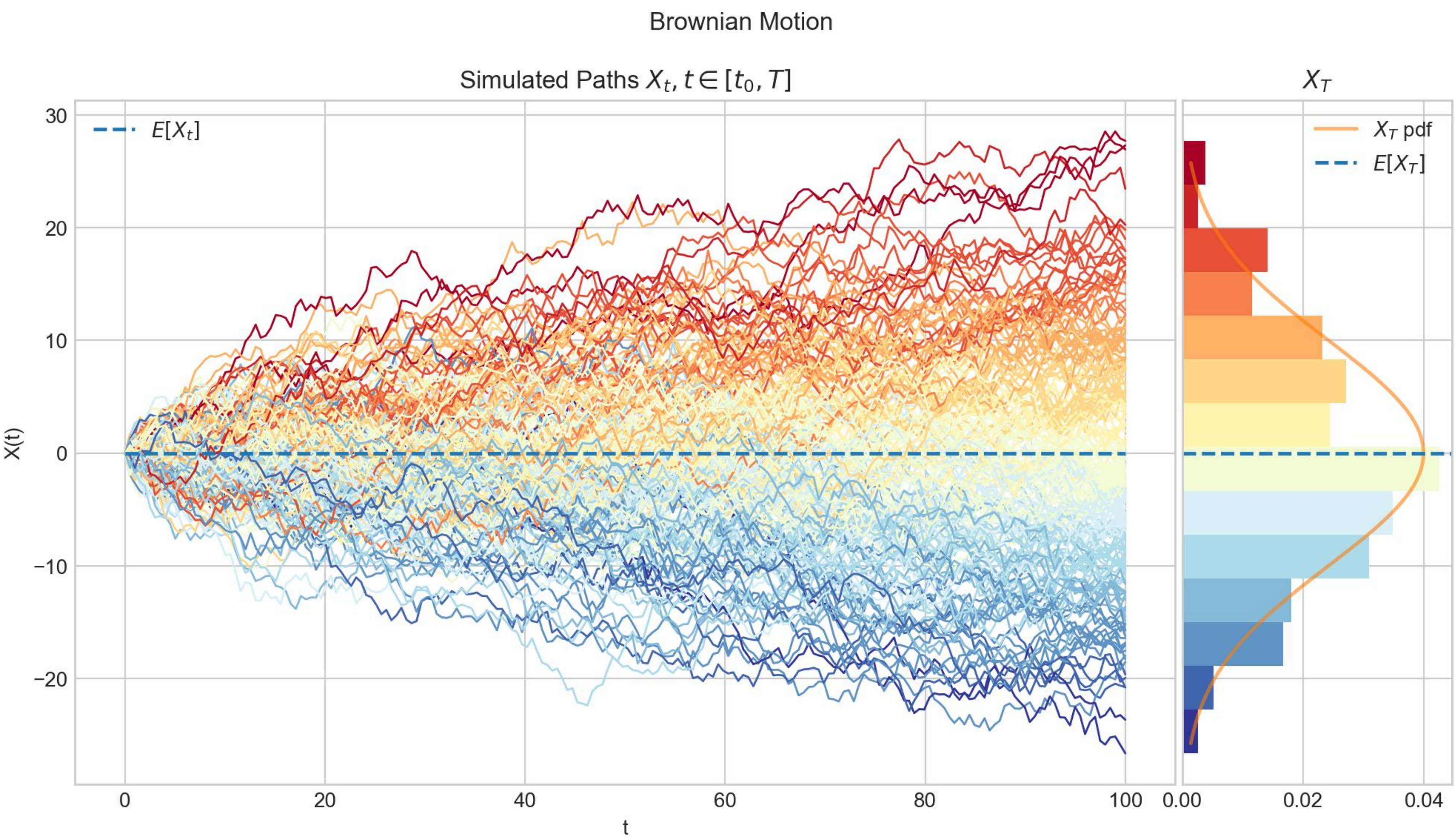
$$E(B(t)|\mathcal{F}_s) = B(s)$$

- Quadratic Variation of BM over $[0, t]$ is t

$$[B, B]([0, t]) = \lim_{\delta_n \to 0} \sum_{i=1}^{n} \left(B(t_i^n) - B(t_{i-1}^n)\right)^2 = t, \qquad \delta_n = \max_i (t_{i+1}^n - t_i^n)$$

Ref. Hui-Hsiung Kuo - Introduction to Stochastic Integration

Fig. Brownian Motion



SDE (Stochastic diffusion equation)

- SDE

$$X(t+\Delta)-X(t)\approx\mu(X(t),t)\Delta+\sigma(X(t),t)\{B(t+\Delta)-B(t)\}$$
$$\Downarrow$$
$$dX(\mathrm{t})=\mu(X(t),t)dt+\sigma(X(t),t)dB(t)$$

- It is just a symbolic expression and is interpreted as meaning the stochastic integral equation

$$X(t)-X(a)=+\int_a^t\mu(X(s),s)ds+\int_a^t\sigma(X(s),s)dB(s)$$

- Let $X(t)$ have SDE

$$dX(t)=\mu(t)dt+\sigma(t)dB(t)$$

If $f\in C^2$

$$df\big(X(t)\big)=f^{(1)}\big(X(t)\big)dX(t)+\frac{1}{2}f^{(2)}(X(t),t)d[X,X](t)$$
$$=\left\{f^{(1)}\big(X(t)\big)\mu(t)+\frac{1}{2}f^{(2)}(X(t),t)\sigma^2(t)\right\}\mathrm{dt}+f^{(1)}\big(X(t)\big)\sigma(t)dB(t)$$

Application

- Consider a Stoc. Proc. $S(t)$ as stock price with following SDE
$$dS(t) = rS(t)dt + \sigma S(t)dB(t), \qquad S(0) = 1$$
- Let r is risk-free interest rate and σ is volatility. Define $R(t)$ as return
$$R(t) = \frac{dS(t)}{S(t)} = rdt + \sigma dB(t)$$
- By Ito lemma
$$dR(t) = d\ln S(t) = \frac{1}{S(t)}dS(t) - \frac{1}{2}\frac{1}{S(t)^2}d[S,S](t)$$
$$= \left(r - \frac{1}{2}\sigma^2\right)dt + \sigma dB(t)$$
- In integral representation
$$R(t) = R(0) + \left(r - \frac{1}{2}\sigma^2\right)t + \sigma B(t)$$
$$S(t) = S(0)e^{\left(r-\frac{1}{2}\sigma^2\right)+\sigma B(t)}$$

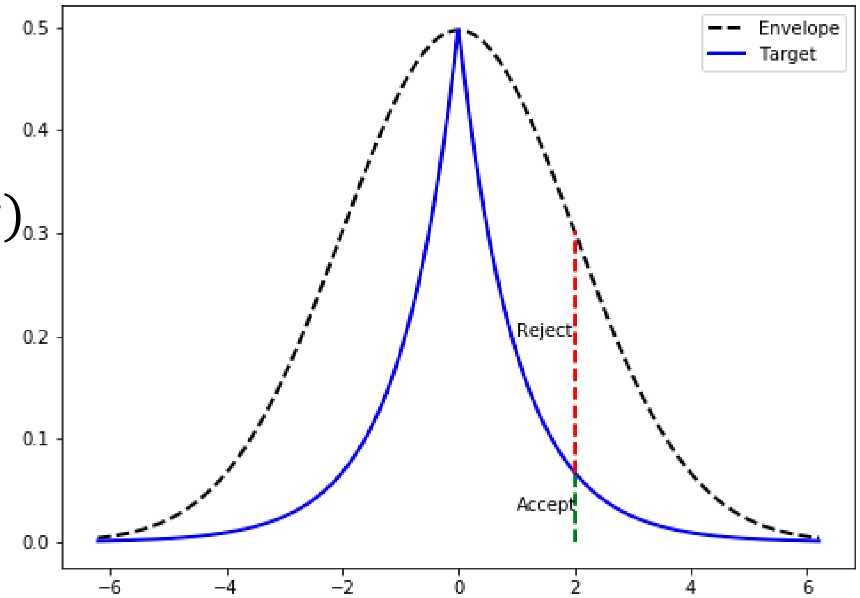
Rejection Sampling

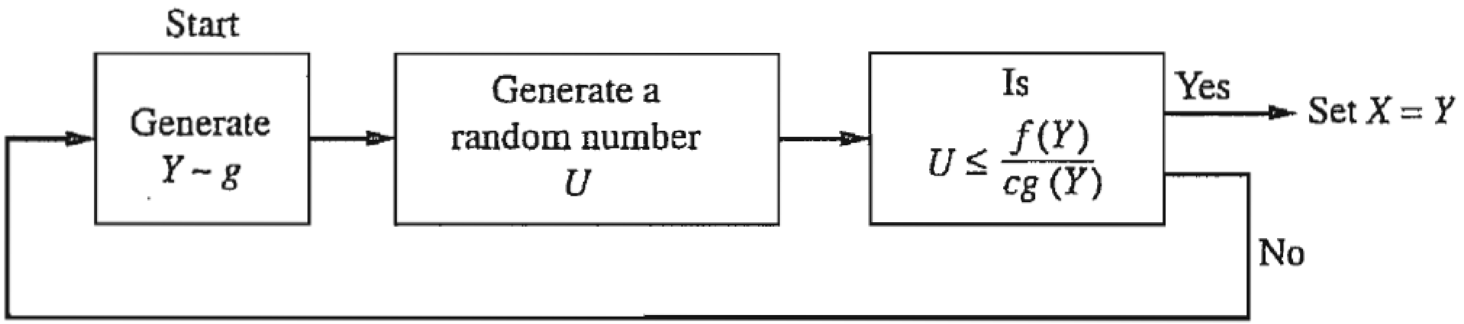
- If the target density is $f(x)$, find a density $g(x)$ satisfy

  $$f(x) \leq cg(x) \text{ for some } c > 0$$

  - step1 :
    Generate $u^{(i)} \sim U(0,1)$ and $x^{(i)} \sim g(x)$
  - step2 :
    Collect $(u^{(i)}, x^{(i)})$ which satisfy

    $$u^{(i)} \leq \frac{f\left(x^{(i)}\right)}{cg\left(x^{(i)}\right)}$$





Ref. Sheldon Ross - Simulation

Rationale

- $U \sim U(0,1)$ and assume that $c = \sup_x \frac{f(x)}{g(x)} < \infty$

$$P(\text{accept}|X = x) = P\left(U \le \frac{f(x)}{cg(x)} \middle| X = x\right) = \frac{f(x)}{cg(x)}$$

The unconditional acceptance probability is the proportion of proposed samples which are accepted which is

$$P(\text{accept}) = \begin{cases} \int_x P(\text{accept}|X = x)g(x)\,dx = \int_x \frac{f(x)}{cg(x)} g(x)\,dx = \frac{1}{c} \\ \sum_x P(\text{accept}|X = x)P(X = x) = \sum_x \frac{f(x)}{cg(x)} g(x) = \frac{1}{c} \end{cases}$$

- Distribution of the accepted values from the rejection sampling follows the target density $f(x)$

$$P(X = x|\text{accept}) = \frac{P(\text{accept}|X = x)P(X = x)}{P(\text{accept})} = \frac{\frac{f(x)}{cg(x)} g(x)}{\frac{1}{c}} = f(x)$$

Exercise

- Generate random number from

$$f(x) = \frac{\pi}{2}\sin(\pi x), \qquad x \in (0,1)$$

  - Candidate 1:

$$g \sim U(0,1)$$

  - Candidate 2:

$$g = \frac{U_1 + U_2}{2}, \qquad U_1, U_2 \sim U(0,1)$$

☺Generate the target distribution from different candidates and compare the efficacy of the sampling.

What are Monte Carlo Methods?

- It can be viewed as a branch of experimental mathematics in which uses random numbers to conduct experiments.
- The Mean Value Theorem states that if f is an integrable function on the $[a, b]$ then

$$\int_a^b f(x)\,dx = (b-a)\bar{f}$$

- If the n random number $x_1, \cdots, x_n$ are chosen uniformly from the $[a, b]$ and the statistic

$$\frac{1}{n}\sum_{k=1}^{n} f(x_k) = \hat{f}_n \to \bar{f}$$

Therefore, by sampling the interval $[a, b]$ at n points, we can

$$\int_a^b f(x)\,dx \approx (b-a)\left(\frac{1}{n}\sum_{k=1}^{n} f(x_k)\right)$$

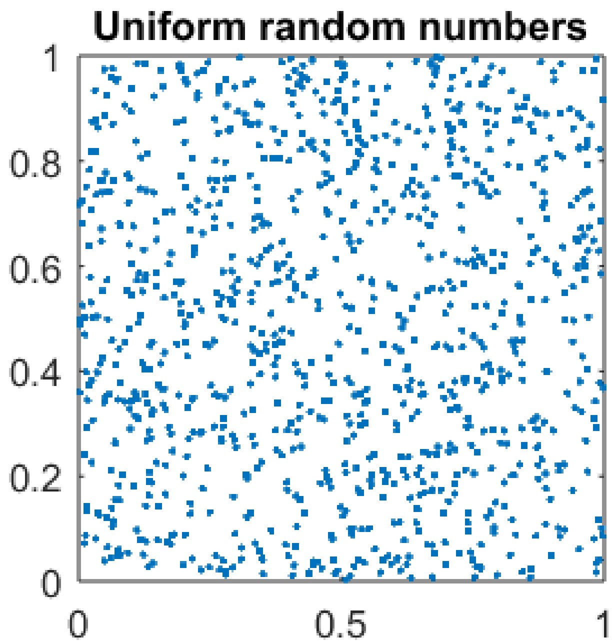
as an estimate of the area under the curve of the function f

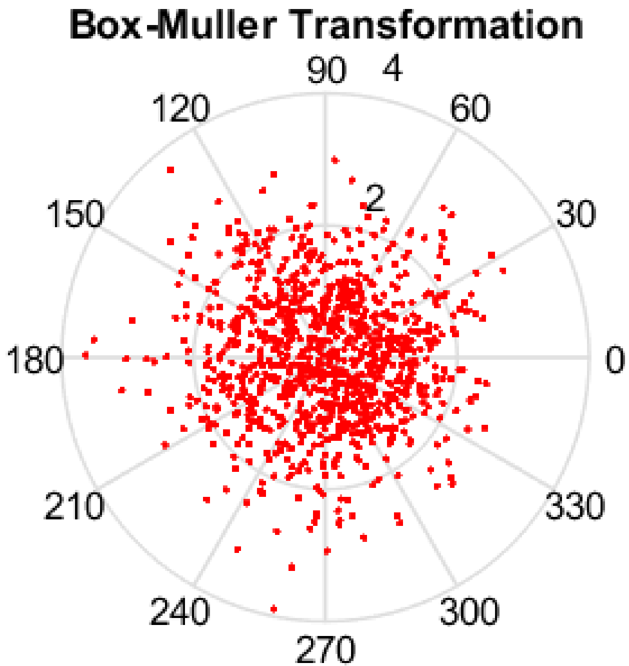
Monte Carlo Integration

- Basic of Monte Carlo
  - Draw random variables
    $$X \sim f(x)$$
    Sometimes with unknown normalizing constant
  - Estimate the integral
    $$I(h) = \int_{E_X} h(x) f(x)\, dx = E_X\big(h(X)\big)$$
- Example : Estimate the $I(h) = \int_0^1 h(x)\, dx$
  - Draw $u_1, \cdots, u_n$ iid $\sim U(0,1)$
  - By Law of Large numbers
    $$\frac{h(u_1) + \cdots + h(u_n)}{n} = \hat{I} \rightarrow I(h), \qquad \text{as } n \rightarrow \infty$$
  - Error Computation:
    $$\mathrm{Var}\big(\hat{I}\big) = \frac{1}{\mathrm{n}} \int_0^1 [h(x) - I]^2\, dx \text{ and } E\big(\hat{I}\big) = I$$

Box-Muller method (Polar method)

- Classic method to generate i.i.d normal distribution.
  - Generate $U_1, U_2 \sim U(0,1)$ and $U_1 \perp U_2$
  - Set $R = \sqrt{-2\ln(U_1)}$ and $\Theta = 2\pi U_2$
  - Set $X = R\cos(\Theta)$ and $Y = R\sin(\Theta)$
- We have $X \perp Y,\ X, Y \sim N(0,1)$





Ref. Sheldon Ross - Simulation

EM algorithm

- Offers a simple way to finding an MLE when the likelihood function is complex. It is often applied to the case where the model involves *hidden/latent* units.

- For $\boldsymbol{\theta} \in \boldsymbol{\Theta} \subseteq \mathbb{R}^p$
$$\boldsymbol{y} \sim f(\boldsymbol{y}|\boldsymbol{\theta}) = f(\mathbf{y}_{\text{obs}}, \boldsymbol{y}_{\text{mis}}|\boldsymbol{\theta}) = f_1(\mathbf{y}_{\text{obs}}|\boldsymbol{\theta}) f_2(\boldsymbol{y}_{\text{mis}}|\mathbf{y}_{\text{obs}}, \boldsymbol{\theta})$$
Thus it follows that
$$\ell_{\text{obs}}(\boldsymbol{\theta}|\mathbf{y}_{\text{obs}}) = \ell(\boldsymbol{\theta}|\boldsymbol{y}) - \ln f_2(\boldsymbol{y}_{\text{mis}}|\mathbf{y}_{\text{obs}}, \boldsymbol{\theta})$$
  - E-step : calculates the expected log likelihood
$$Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(r)}\right) = E\left(\ell(\boldsymbol{\theta}|\boldsymbol{y})\middle|\boldsymbol{y}_{\text{obs}}, \boldsymbol{\theta}^{(r)}\right)$$
  - M-step : finds its maximum.
$$\boldsymbol{\theta}^{(r+1)} = \arg\max_{\boldsymbol{\theta}} Q\left(\boldsymbol{\theta}\middle|\boldsymbol{\theta}^{(r)}\right)$$

- (Ascending property of EM)
The sequence $\boldsymbol{\theta}^{(r)}$ satisfies
$$\ell_{\text{obs}}\left(\boldsymbol{\theta}^{(r+1)}\middle|\boldsymbol{y}_{\text{obs}}\right) \geq \ell_{\text{obs}}\left(\boldsymbol{\theta}^{(r)}\middle|\boldsymbol{y}_{\text{obs}}\right)$$

Ref. Casella – Statistical Inference

Example

- Let $\mathbf{Y} = (y_1, \cdots, y_m | y_{m+1}, \cdots, y_n)^\top = (\boldsymbol{Y}_{\text{obs}} | \boldsymbol{Y}_{\text{mis}})^\top$ and $\mathrm{Y_i} \sim^{\text{i.i.d}} N(\theta, \sigma^2)$

$$\ell(\theta|\boldsymbol{Y}) = \ln \mathcal{L}(\theta|\boldsymbol{Y}) = -\frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{n} (Y_i - \theta)^2$$

- Define the Q function

$$Q\left(\theta|\theta^{(r)}\right) = E\left(-\frac{n}{2} ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \textstyle\sum_{i=1}^{n} (Y_i - \theta)^2 \middle| \boldsymbol{Y}_{\text{obs}}, \theta^{(r)}\right)$$

$$= -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \theta)^2 - \frac{1}{2\sigma^2} \sum_{i=m+1}^{n} E\left((Y_i - \theta)^2 | \boldsymbol{Y}_{\text{obs}}, \theta^{(r)}\right)$$

$$= -\frac{n}{2} \ln(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_{i=1}^{m} (y_i - \theta)^2 - \frac{n-m}{2\sigma^2} \left\{\left(\sigma^2 + \left[\theta^{(r)}\right]^2\right) - 2\theta^{(r)}\theta + \theta^2\right\}$$

- In M-step, maximize the Q function

$$\frac{\partial}{\partial\theta} Q\left(\theta|\theta^{(r)}\right) = 0 \Leftrightarrow \frac{1}{\sigma^2} \sum_{i=1}^{m} (y_i - \theta) - \frac{n-m}{\sigma^2} \left\{\theta - \theta^{(r)}\right\} = 0$$

which yield that

$$\theta^{(r+1)} = \frac{1}{n} \left\{\sum_{i=1}^{m} y_i + (n-m)\theta^{(r)}\right\}$$

Metropolis Hastings Algorithm

- For any target $\pi(x)$, the M-H algorithm proceeds as follows.

Algorithm 1 Metropolis-Hastings algorithm

Initialize $x^{(0)} \sim q(x)$
for iteration $i = 1, 2, \ldots$ **do**
  Propose: $x^{cand} \sim q(x^{(i)}|x^{(i-1)})$
  Acceptance Probability:
    $$\alpha(x^{cand}|x^{(i-1)}) = \min\left\{1, \frac{q(x^{(i-1)}|x^{cand})\pi(x^{cand})}{q(x^{cand}|x^{(i-1)})\pi(x^{(i-1)})}\right\}$$
  $u \sim \text{Uniform}\ (u; 0, 1)$
  if $u < \alpha$ **then**
    Accept the proposal: $x^{(i)} \leftarrow x^{cand}$
  else
    Reject the proposal: $x^{(i)} \leftarrow x^{(i-1)}$
  end if
end for

Siddhartha CHIB and Edward GREENBERG

The American Statistician - Understanding the Metropolis-Hastings Algorithm

M-H Rationale

- $q(y|x)$ is candidate generating density. It can be interpreted as when a process is at the point x, the density generates a value y from it.

- We refer $\alpha(y|x)$ as the probability of move. Thus transitions from x to y are made according to
$$p_{\mathrm{MH}}(y|x) = q(y|x)\alpha(y|x)$$

- Note that if
$$\pi(x)q(y|x) > \pi(y)q(x|y)$$
It tells us that movement from y to x is not made often enough. We should therefore define $\alpha(x|y)$ to be as large as possible. But now $\alpha(y|x)$ is determined by requiring that $p_{\mathrm{MH}}(y|x)$ satisfies the reversibility condition
$$\pi(x)q(y|x)\alpha(y|x) = \pi(y)q(x|y)\alpha(x|y) = \pi(y)q(x|y)1$$
We now see
$$\alpha(y|x) = \frac{\pi(y)q(y|x)}{\pi(x)q(x|y)}$$