# 1 Itô integral (I)

$$E\left[\sum_{i=1}^{n} X_{t_{i-1}} \Delta W_{t_{i-1}} \cdot \sum_{i=1}^{n} Y_{t_{i-1}} \Delta W_{t_{i-1}}\right] = \sum_{i=1}^{n} E\Big[X_{t_{i-1}} Y_{t_{i-1}} \Delta W_{t_{i-1}}^2\Big]$$

$$= \sum_{i=1}^{n} E\Big[X_{t_{i-1}} Y_{t_{i-1}} E[\Delta W_{t_{i-1}}^2 | \mathcal{F}_t]\Big] = \sum_{i=1}^{n} E\Big[X_{t_{i-1}} Y_{t_{i-1}} \Delta t\Big]$$

# 2 Itô integral (II)

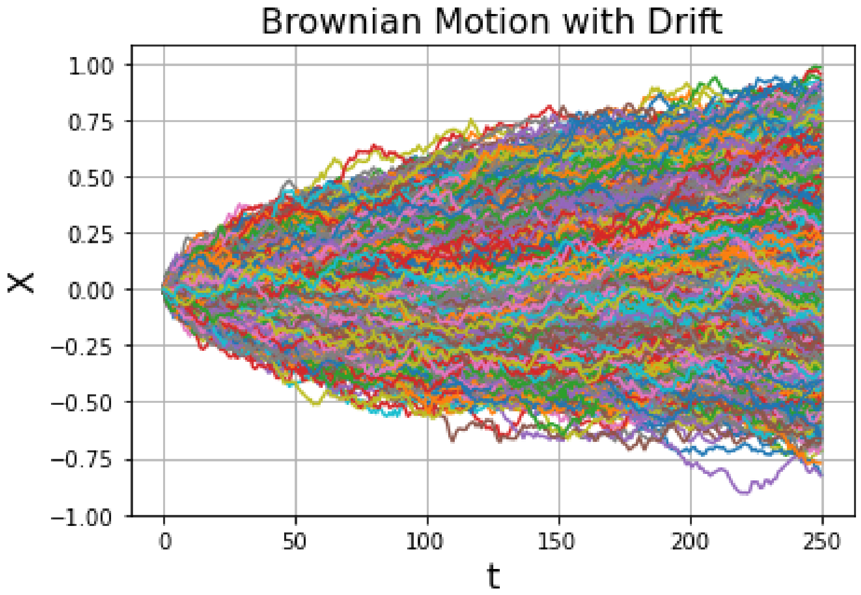
$$Var\left(\int_0^t X_s dW s\right) = E\left[\left(\int_0^t X_s dW_s\right)^2\right] = E\left[\int_0^t X_S dW_s \cdot \int_0^t X_S dW_s\right]$$

$$= \int_0^t E\Big(X_s^2 dW_s^2\Big) = \int_0^t E\left[X_s^2 E\Big(dW_s^2 | \mathcal{F}_s\Big)\right] = \int_0^t E\Big(X_s^2\Big) dS = X_t^2$$

# 3 Brownian Motion simulation

```
import numpy as np
import pandas as pd
from math import sqrt
from pylab import plot, show, grid, xlabel, ylabel, title
## loop data
k = 10000
## year data
T_year = 1
N = 250
h = T_year/N
mu = 0.1
var_year = 0.25
X0 = 0
## daily data
T_day = 1/250
var_day = var_year/sqrt(250)
def BM(N, h, var_year):
    dt = h
    random_increments = np.random.normal(0, 1*var_year, N)*sqrt(dt)
    brownian_motion = np.cumsum(random_increments)
    brownian_motion = np.insert(brownian_motion, 0, 0)
    return brownian_motion, random_increments
```

```
def BM_with_drift(mu, N, h):
    W, _ = BM(N, h, var_year)
    dt = h
    time_steps = np.linspace(0, T_year, N+1)
    X = mu*time_steps + W
    return X
for i in range(k):
    X = BM_with_drift(mu, N, h)
    plot(X)
xlabel('t', fontsize = 16)
ylabel('X', fontsize = 16)
title("Brownian Motion with Drift", fontsize = 16)
grid(True)
show()
```



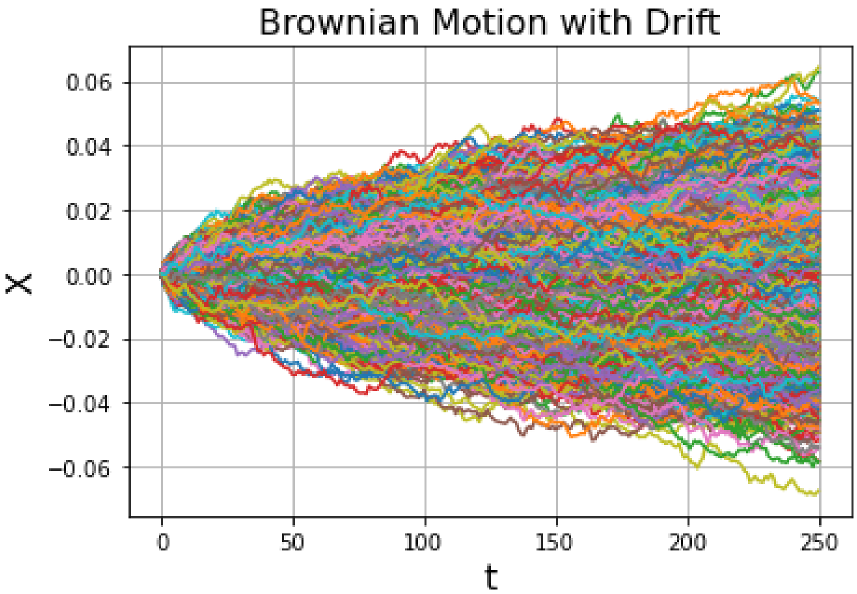
# Brownian Motion simulation with daily parameters

```
import numpy as np
import pandas as pd
from math import sqrt
from pylab import plot, show, grid, xlabel, ylabel, title
## loop data
k = 10000
## year data
T_year = 1
N = 250
h = T_year/N
```

```
mu = 0.1
var_year = 0.25
X0 = 0
## daily data
T_day = 1/250
var_day = var_year/sqrt(250)

def BM_with_drift_daily(mu, N, h):
    W, _ = BM(N, h, var_day)
    dt = h
    time_steps = np.linspace(0, T_day, N+1)
    X = mu*time_steps + W
    return X

for i in range(k):
    X = BM_with_drift_daily(mu/250, N, h)
    plot(X)
xlabel('t', fontsize = 16)
ylabel('X', fontsize = 16)
title("Brownian Motion with Drift", fontsize = 16)
grid(True)
show()
```


Brownian Motion with Drift
X
t