# M86 Homework 2

## Alex Craig & Ganqi Li

## Exercise 1

### Wilmott, Chapter 3, Problem 8

A share price is currently \$180. At the end of one year, it will be either \$203 or \$152. The risk-free interest rate is 3% p.a. with continuous compounding. Consider an American put on this underlying. Find the exercise price for which holding the option for the year is equivalent to exercising immediately. This is the break-even exercise price. What effect would a decrease in the interest rate have on this break-even price?

### Solution

Let's denote:

- $S_0 = \$180$ as the current share price,
- $S_u = \$203$ as the share price at the end of the year if it goes up,
- $S_d = \$152$ as the share price at the end of the year if it goes down,
- $r = 3\%$ as the continuous compounding annual risk-free interest rate,
- $K$ as the exercise price of the American put option.

The value of exercising the option immediately is $V_0 = K - S_0$. We know $K \geq S_0$, or else exercising immediately yields a negative return, while holding the put for a year yields a minimum return of 0 if the put is not exercised, and no equivalence can be formed.

The value of the option at the end of the year in each scenario is:

- If the price goes up: $V_u = \max(K - S_u, 0)$,
- If the price goes down: $V_d = \max(K - S_d, 0)$.

The present/discounted value of the option exercised at the end of the year ($T = 1$) at rate $r$, for both upward and downward movements, is given by:

$$PV_u = V_u \cdot e^{-r} = \max(K - S_u, 0) \cdot e^{-r}$$
$$PV_d = V_d \cdot e^{-r} = \max(K - S_d, 0) \cdot e^{-r}$$

We first find the risk-neutral probabilities for the share price going up, denoted as $q_u$ and going down, denoted as $q_d$.

The up-factor, denoted as $u$, and the down-factor, denoted as $d$, are given by:

$$u = \frac{S_u}{S_0} = \frac{203}{180} = 1.127778, \quad d = \frac{S_d}{S_0} = \frac{152}{180} = 0.844444$$

The risk-neutral probabilities are given by:

$$q_u = \frac{e^r - d}{u - d} = \frac{e^{0.03} - 0.844444}{1.127778 - 0.844444} = 0.6565$$

$$q_d = 1 - q_u = 0.3435$$

The method given in Wilmott, Ch.3, p.72, for finding the risk-neutral probabilities is as follows:

$$e^{-r} \cdot (q_u \cdot S_u + (1 - q_u) \cdot S_d) = S_0$$

$$e^{-0.03} \cdot (q_u \cdot 203 + (1 - q_u) \cdot 152) = 180$$

$$q_u = 0.6565, \quad q_d = (1 - q_u) = 0.3435$$

To find the break-even exercise price $K$ for which holding the option for the year is equivalent to exercising immediately, we set the immediate exercise value equal to the present value of exercising at the end of the year.

Thus, the equation to solve for $K$ is:

$$K - S_0 = e^{-r}(q_u \cdot \max(K - S_u, 0) + q_d \cdot \max(K - S_d, 0))$$

$$K - 180 = e^{-0.03}(0.6565 \cdot \max(K - 203, 0) + 0.3435 \cdot \max(K - 152, 0))$$

Using the solver on WolframAlpha, we get the **break-even strike price,** $K = \$194$.

**Decreasing Interest Rate** If the interest rate is lowered, the break-even strike price should increase. Because a lower interest rate reduces the return to exercising early and earning interest, and a higher break-even strike price would compensate by increasing the return to exercising early. For instance, if the interest rate is lowered to 2%, the break-even strike price would increase to \$196.6.

## Exercise 2

### Wilmott, Chapter 7, Problems 1, 8, 9

**Question 7.1** Consider an option with value $V(S,t)$, which has payoff at time $T$. Reduce the Black–Scholes equation, with final and boundary conditions, to the diffusion equation, using the following transformations:

**(a)** $S = Ee^x, \quad t = T - \frac{2\tau}{\sigma^2}, \quad V(S,t) = Ev(x,\tau)$

**(b)** $v = e^{\alpha x + \beta\tau}u(x,\tau)$

For some $\alpha$ and $\beta$. What is the transformed payoff? What are the new initial and boundary conditions? Illustrate with a vanilla European call option.

### Solution

We know the Black-Scholes equation:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$$

and we need to derive expressions for (1) $\frac{\partial V}{\partial t}$, (2) $rS\frac{\partial V}{\partial S}$, (3) $\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}$, (4) $-rV$.

**(1)** Since $x$ and $t$ are independent, we have:

$$\frac{\partial V}{\partial t} = \frac{\partial V}{\partial \tau} \cdot \frac{\partial \tau}{\partial t},$$

where $\frac{\partial V}{\partial \tau} = E(e^{\alpha x + \beta\tau})(u_\tau + \beta u)$ and $\frac{\partial \tau}{\partial t} = \frac{-\sigma^2}{2}$, so:

$$\frac{\partial V}{\partial t} = E(e^{\alpha x + \beta\tau})(u_\tau + \beta u) \cdot \frac{-\sigma^2}{2}$$

**(2)** We also have:

$$\frac{\partial V}{\partial S} = \frac{\partial V}{\partial x} \cdot \frac{\partial x}{\partial S},$$

where $\frac{\partial V}{\partial x} = E(e^{\alpha x + \beta\tau})(u_x + \alpha u)$, $\frac{\partial x}{\partial S} = \frac{1}{Ee^x}$, $S = Ee^x$, so:

$$\frac{\partial V}{\partial S} = E(e^{\alpha x + \beta\tau})(u_x + \alpha u) \cdot \frac{1}{Ee^x} = \frac{\partial V}{\partial x} \cdot \frac{\partial x}{\partial S} = e^{\alpha x + \beta t}(u_x + \alpha u)$$

And thus:

$$rS\frac{\partial V}{\partial S} = r(Ee^x) \cdot \frac{\partial V}{\partial S} = r(E(e^{\alpha x + \beta \tau})(u_x + \alpha u))$$

**(3)** We also have:

$$\frac{\partial^2 V}{\partial S^2} = \frac{\partial x}{\partial S} \cdot \frac{\partial}{\partial x}(\frac{\partial V}{\partial S})$$

where $ \frac{\partial}{\partial x}(\frac{\partial V}{\partial S}) = (\alpha -1)e^{(\alpha -1)x + \beta t}(u_x + \alpha u) + e^{(\alpha -1)x + \beta t}(u_{xx}+\alpha u_x)$, so:

$$\frac{\partial^2 V}{\partial S^2} = \frac{1}{Ee^x} \cdot (\alpha - 1)e^{(\alpha-1)x+\beta t}(u_x + \alpha u) + e^{(\alpha-1)x+\beta t}(u_{xx} + \alpha u_x)$$

And thus:

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = \frac{\sigma^2}{2}(Ee^x)^2 \frac{1}{Ee^x} \cdot (\alpha - 1)e^{(\alpha-1)x+\beta t}(u_x + \alpha u) + e^{(\alpha-1)x+\beta t}(u_{xx} + \alpha u_x)$$

which simplifies to

$$\frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} = \frac{\sigma^2 E}{2} \cdot (\alpha - 1)e^{\alpha x+\beta t}(u_x + \alpha u) + e^{\alpha x+\beta t}(u_{xx} + \alpha u_x)$$

**(4)** We have:

$$-rV = -rEe^{\alpha x \beta t}u$$

**Combining (1)-(4)** We can transform $\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$ into:

$$Ee^{\alpha x+\beta t}(u(\frac{-\beta\sigma^2}{2} + \alpha r + \frac{\sigma^2}{2}(\alpha-1)\alpha - r) + u_t(\frac{-\sigma^2}{2}) + u_{xx}\frac{\sigma^2}{2} + u_x(r + \frac{\sigma^2}{2}(\alpha - 1) + \frac{\sigma^2\alpha}{2})) = 0$$

We notice that by setting the coefficients on $u$ and $u_x$ to 0, we have a homogeneous diffusion equation:

$$Ee^{\alpha x+\beta t}\frac{\sigma^2}{2}(u_{xx} - u_t) = 0$$

where $u_{xx} - u_t = 0$. Thus, we can solve for the expressions of $\alpha$ and $\beta$ using:

$$\frac{-\beta\sigma^2}{2} + \alpha r + \frac{\sigma^2}{2}(\alpha - 1)\alpha - r = 0$$

$$r + \frac{\sigma^2}{2}(\alpha - 1) + \frac{\sigma^2\alpha}{2} = 0$$

Using the two equations, we get:

$$\alpha = \frac{1}{2} - \frac{r}{\sigma^2}$$

$$\beta = \frac{-1}{4} - \frac{r}{\sigma^2} - \frac{r^2}{\sigma^4}$$

**Payoff** Since $V(s,t) = Ev(x,\tau) = Ee^{\alpha x+\beta\tau}u(x,\tau)$, $\quad \alpha = \frac{1}{2} - \frac{r}{\sigma^2}$, $\quad \beta = \frac{-1}{4} - \frac{r}{\sigma^2} - \frac{r^2}{\sigma^4}$, we have the new payoff function:

$$V(s,t) = Ee^{(\frac{1}{2}-\frac{r}{\sigma^2})x+(\frac{-1}{4}-\frac{r}{\sigma^2}-\frac{r^2}{\sigma^4})\tau}u(x,\tau),$$

where $u(x,\tau)$ is the solution formula to a homogeneous heat equation:

$$u(x,\tau) = \frac{1}{\sqrt{4\pi\tau}} \int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{4\tau}} u_0(y)dy$$

**Initial and Boundary Conditions** We know the payoff for a European call option is $\max(S-E,0) = \max(Ee^x, 0)$ at time $T$. Therefore its final value (or initial condition in terms of $u(x,\tau)$) is

$$V(x,T=2,\tau=0) = Ee^{(\frac{1}{2}-\frac{r}{\sigma^2})x}u(x,\tau=0) = Ee^{(\frac{1}{2}-\frac{r}{\sigma^2})x}\max(e^x-1,0) = Ee^{(\frac{1}{2}-\frac{r}{\sigma^2})x}u_0(x).$$

For boundary conditions, since $E = S/e^x$, as $x \to -\infty$, we have $V(x,t) \to 0$, because the strike price goes to $+\infty$. And as $x \to \infty$, we have $V(x,t) \to S$, because the strike price goes to 0.

**Question 7.8** Show that if

$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2}$, on $-\infty < x < \infty$, $\tau > 0$,

with

$u(x,0) = u_0(x) > 0$,

then $u(x,\tau) > 0$ for all $\tau$.

Use this result to show that an option with positive payoff will always have a positive value.

**Solution:** The solution to the heat equation is given by:

$$u(x,\tau) = \frac{1}{\sqrt{4\pi\tau}} \int_{-\infty}^{\infty} e^{-\frac{(x-y)^2}{4\tau}} u_0(y)dy$$

Since $u_0(x) > 0$, the integral is a positive number, and the exponential term is always positive. Also, because $\tau > 0$, the fraction is always positive. Thus, the value of $u(x,\tau)$ is always positive. And since $V(S,t) = Ee^{\alpha x+\beta\tau}u(x,\tau)$, we know all components of the formula are positive, the option with positive payoff will also always have a positive value

**Question 7.9** If $f(x,\tau) \geq 0$ in the initial value problem

$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial x^2} + f(x,\tau)$, on $-\infty < x < \infty$, $\tau > 0$,

with

$u(x,0) = 0$, and $u \to 0$ as $|x| \to \infty$,

then $u(x,\tau) \geq 0$. Hence show that if $C_1$ and $C_2$ are European calls with volatilities $\sigma_1$ and $\sigma_2$ respectively, but are otherwise identical, then $C_1 > C_2$ if $\sigma_1 > \sigma_2$.

Use put-call parity to show that the same is true for European puts.

**Solution** From question 7.1, we have the Black-Scholes equation:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0$$

But since we now have a nonhomogeneous PDE setup for the initial value problem, or $u_\tau - u_{xx} = f(x,\tau)$, we have a new Black-Scholes set up:

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = F(S,t)$$

So if we set up the nonhomogeneous Black-Scholes equation for $C_1$ and $C_2$ and subtract the second from tyhe first, we will have:

$$\frac{\partial(C_1 - C_2)}{\partial t} + rS\frac{\partial(C_1 - C_2)}{\partial S} + \frac{1}{2}(\sigma_1^2 - \sigma_2^2)S^2\frac{\partial^2(C_1 - C_2)}{\partial S^2} - rV = F - F = 0$$

Similar to question 7.1, by solving the above setup, we will obtain **a positive value for** $C_1 - C_2$. And since $C_1 - C_2 > 0$, we know the call with a higehr volatility will be more expensive.

Under the **put-call parity** for European options, we have $C_t + PV(E) = P_t + S_t$, so if the value of a European call increases due to higher volatility, the value of a European put also rises.

## Exercise 3

Suppose $W_t$ and $Z_t$ are two independent Brownian motions. This means that each satisfies the 4 defining properties of Brownian motion and the increments $Z_t - Z_s$ and $W_t - W_s$ are independent random variables for all $0 \le s < t$. For a constant $\rho$ with $-1 \le \rho \le 1$, consider the random process:

$$X_t = \rho W_t + \sqrt{1 - \rho^2}Z_t.$$

Is $X_t$ a Brownian motion? Explain your answer.

**Solution** The four conditions of Brownian motion are:

1. $W_0 = 0$
2. The distribution of the increment $W_{t2} - W_{t1}, \quad 0 \le t_1 \le t_2$ is normal with mean 0 and variance $t_2 - t_1$
3. For non-overlapping intervals $t_1 < t_2 \le t_3 < t_4$, the increments $W_{t2} - W_{t1}$ and $W_{t4} - W_{t3}$ are independent random variables
4. The function $t \to W_t$ is almost surely continuous

Let's go through each of these conditions for the process $X_t = \rho W_t + \sqrt{1 - \rho^2}Z_t$:

1. $X_0 = \rho W_0 + \sqrt{1 - \rho^2}Z_0 = \rho \cdot 0 + \sqrt{1 - \rho^2} \cdot 0 = 0$. This condition is satisfied.
2. The distribution of the increment $X_{t2} - X_{t1}, \quad 0 \le t_1 \le t_2$ is normal with mean 0 and variance $t_2 - t_1$:

$$X_{t2} - X_{t1} = \rho(W_{t_2} - W_{t_1}) + \sqrt{1 - \rho^2}(Z_{t_2} - Z_{t_1})$$

Let's take the expectation and variance of this increment:

$$E[X_{t2} - X_{t1}] = \rho E[W_{t_2} - W_{t_1}] + \sqrt{1 - \rho^2}E[Z_{t_2} - Z_{t_1}] = \rho \cdot 0 + \sqrt{1 - \rho^2} \cdot 0 = 0$$

$$Var[X_{t2} - X_{t1}] = \rho^2 Var[W_{t_2} - W_{t_1}] + (1 - \rho^2)Var[Z_{t_2} - Z_{t_1}] = \rho^2(t_2 - t_1) + (1 - \rho^2)(t_2 - t_1) = t_2 - t_1$$

1. For non-overlapping intervals $t_1 < t_2 \le t_3 < t_4$, the increments $X_{t2} - X_{t1}$ and $X_{t4} - X_{t3}$ are independent random variables:

$$X_{t2} - X_{t1} = \rho(W_{t_2} - W_{t_1}) + \sqrt{1 - \rho^2}(Z_{t_2} - Z_{t_1}), \quad X_{t4} - X_{t3} = \rho(W_{t_4} - W_{t_3}) + \sqrt{1 - \rho^2}(Z_{t_4} - Z_{t_3})$$

Given that $W_t$ and $Z_t$ describe Brownian motion, the increments $W_{t2} - W_{t1}$ and $W_{t4} - W_{t3}$ are independent random variables, and the increments $Z_{t2} - Z_{t1}$ and $Z_{t4} - Z_{t3}$ are independent random variables. Since $W_t$ and $Z_t$ are independent, the increments $W_{t2} - W_{t1}$ and $Z_{t2} - Z_{t1}$ are independent random variables. Therefore, the increments $X_{t2} - X_{t1}$ and $X_{t4} - X_{t3}$ are independent random variables.

1. The function $t \to X_t$ is almost surely continuous:

The function $t \to X_t$ is a linear combination of two Brownian motions, $W_t$ and $Z_t$, and is therefore continuous.

## Exercise 4

If $W_t$ is the Wiener process, show that the limit

$$\lim_{\Delta t \to 0} \frac{(\Delta W)^2}{\Delta t} = \lim_{\Delta t \to 0} \frac{(W_{t+\Delta t} - W_t)^2}{\Delta t}$$

does not exist.

**Solution:** We know as $\Delta t \to 0$, we have $(W_{t+\Delta t} - W_t) \to 0$, because the Weiner process is continuous. Since both numerator and denominator go to 0, we have $\lim_{\Delta t \to 0} \frac{(W_{t+\Delta t} - W_t)^2}{\Delta t} = \frac{0}{0}$. Thus, we need to apply L'Hôpital's rule.

Taking the derivative of the denominator with respect to $\Delta t$, we have $\frac{d\Delta t}{d\Delta t} = 1$. Taking the derivative of the numerator, we have $\frac{d(W_{t+\Delta t} - W_t)^2}{d\Delta t} = 2(W_{t+\Delta t} - W_t)\frac{dW_{t+\Delta t}}{d\Delta t}$. However, the Weiner process (Brownian motion) is nowhere differentiable almost surely (See slide: L4 quadratic variation, p.3), so $\frac{dW_{t+\Delta t}}{d\Delta t}$ is undefined. The is because **for all $\Delta t > 0$, the value of $|\Delta W|$ can be arbitrarily large**. Thus, the limit does not exist.

## Exercise 5

Use Ito's Lemma to derive a formula for $\int_0^T W^3 dW$.

**Solution:** Ito's Lemma states that if a process is an **Ito process**, i.e. it is a random process of the form

$$X_t = X_0 + \int_0^t Y dt + \int_0^t Z dW, \quad t \in [0, T]$$

or in shorthand

$$dX = Y dt + Z dW$$

where $Y$ and $Z$ are stochastic processes, and $f(x)$ is a smooth function of $x \in \mathbb{R}$, then $f(X_T)$ is an Ito process that satisfies

$$f(X_T) = X_0 + \int_0^T \left( f'(X)Y + \frac{1}{2} f''(X)Z^2 \right) dt + \int_0^T f'(X) Y dW$$

or in shorthand

$$df(X) = f'(X)dX + \frac{1}{2} f''(X)(dX)^2$$

To apply Ito's Lemma for deriving a formula for $\int_0^T W^3 dW$, let's consider a function $f(W_t)$ of the Brownian motion $W_t$, specifically $f(W) = \frac{1}{4}W^4$. We aim to express the differential $df(W)$ in terms of Ito's Lemma and then integrate to find the desired integral.

Given $f(W) = \frac{1}{4}W^4$, we compute the first and second derivatives with respect to $W$:

- The first derivative, $f'(W) = W^3$,
- The second derivative, $f''(W) = 3W^2$.

Using Ito's Lemma, for an Ito process $W_t$, we have:

$$df(W) = f'(W)dW + \frac{1}{2} f''(W)(dW)^2$$

Substituting $f'(W) = W^3$ and $f''(W) = 3W^2$, we get:

$$df(W) = W^3 dW + \frac{1}{2} \cdot 3W^2 \cdot dW^2$$

Because $dW^2 = dt$ (when inside an integral), the equation simplifies to:

$$df(W) = W^3 dW + \frac{3}{2} W^2 dt$$

To find the integral $\int_0^T W^3 dW$, we look at the differential form $df(W)$. Since $f(W) = \frac{1}{4}W^4$, integrating both sides from 0 to $T$ gives:

$$f(W_T) - f(W_0) = \int_0^T W^3 dW + \frac{3}{2} \int_0^T W^2 dt$$

Given $f(W) = \frac{1}{4}W^4$, we have $f(W_T) - f(W_0) = \frac{1}{4}W_T^4 - \frac{1}{4}W_0^4$, assuming $W_0 = 0$ (the starting point of Brownian motion is often taken as 0), this simplifies to:

$$\frac{1}{4}W_T^4 = \int_0^T W^3 dW + \frac{3}{2} \int_0^T W^2 dt$$

Therefore, the formula for $\int_0^T W^3 dW$ can be expressed as:

$$\int_0^T W^3 dW = \frac{1}{4}W_T^4 - \frac{3}{2} \int_0^T W^2 dt$$

This formula gives us the integral of $W^3$ with respect to $dW$ over the interval $[0, T]$, utilizing Ito's Lemma for the derivation.

## Exercise 6

For a stochastic price process $X_t : [0, T] \to \mathbb{R}$ define the approximate quadratic variation $Q_m$ as

$$Q_m = \sum_{j=1}^{m} (X_{t_j} - X_{t_{j-1}})^2$$

with $t_j = \frac{jT}{m}$ for $j = 0, 1, 2, \ldots, m$, partitioning $[0, T]$ into intervals of equal size $\Delta t = \frac{T}{m}$.

### (a)

Assume $X_t$ is an Ito process with $dX = \mu dt + \sigma dW$ with constant $\mu, \sigma$. Derive formulas for $E(Q_m)$ and $\text{Var}(Q_m)$ for large $m$.

### (b)

Explain why the distribution of $Q_m$ is approximately normal.

### (c)

Use your code from Problem Set 1 Problem (4) that generates a random stock price path for one year (with $\alpha = 10\%, \sigma = 30\%$). Modify your code to use $N = 2000$ time steps instead of $N = 252$. Then add a function that calculates $Q_{50}$ and $Q_{250}$ for $\log(S)$ for the price data generated by the previous function (effectively using $\Delta t = 5$ days and $\Delta t = 1$ day respectively, assuming there are 250 trading days in the year). Generate 1000 random stock price paths and calculate $Q_{50}$ and $Q_{250}$ for each path. Plot the two histograms for the distributions of the values of $Q_{50}$ and $Q_{250}$ for the 1000 sample paths.

### (d)

What are the average and standard deviation for the empirical distributions of $Q_{50}$ and $Q_{250}$ respectively? Do the empirical results confirm your calculation in item (a)?

### (e)

In practice, is $Q_{50}$ or $Q_{250}$ a better estimator of quadratic variation? Why?

### Solution

### (a)

**Deriving $E(Q_m)$** Since the Ito process $X_t$ is described by $dX = \mu dt + \sigma dW$, we have:

$$(X_{t_j} - X_{t_{j-1}})^2 = (\Delta X)^2 = (\mu \Delta t + \sigma \Delta W)^2,$$

which we can expand into:

$$(X_{t_j} - X_{t_{j-1}})^2 = (\Delta X)^2 = \mu^2 \Delta t^2 + \sigma^2 \Delta W^2 + 2\mu \Delta t \sigma \Delta W$$

Using $E(\Delta W) = 0, \quad E(\Delta W^2) = \Delta t$, we get:

$$E(\Delta X) = \mu^2 \Delta t^2 + \sigma^2 \Delta t + 0.$$

Since the dift term $\mu$ is negligible for small $\Delta t$, we have $E(\Delta X) \approx \sigma^2 \Delta t$. Thus, summing up $E(\Delta X)$ across all time-steps, we have:

$$E(Q_m) = \sum_{j=1}^{m} E((X_{t_j} - X_{t_{j-1}})^2) = \sum_{j=1}^{m} (\sigma^2 \Delta t) = \sigma^2 (m \Delta t) = \sigma^2 T$$

**Deriving Var$(Q_m)$** We know $\text{Var}(\Delta X) = \text{Var}(\mu^2 \Delta t^2 + \sigma^2 \Delta W^2 + 2\mu \Delta t \sigma \Delta W)$, and that $\mu$ and $\sigma$ are constant terms, and $\Delta t$ is fixed. We can set $\mu \approx 0$ for large $m$, so we have:

$$\text{Var}(\Delta X) = \sigma^4 \cdot \text{Var}((\Delta W^2))$$

Since we know $\Delta W$ is an identically and independently distributed variable, we have \$\sigma^4 \text{Var}((\Delta W^2) = \sigma^4 \cdot 2 (\Delta t)^2 \$, and thus:

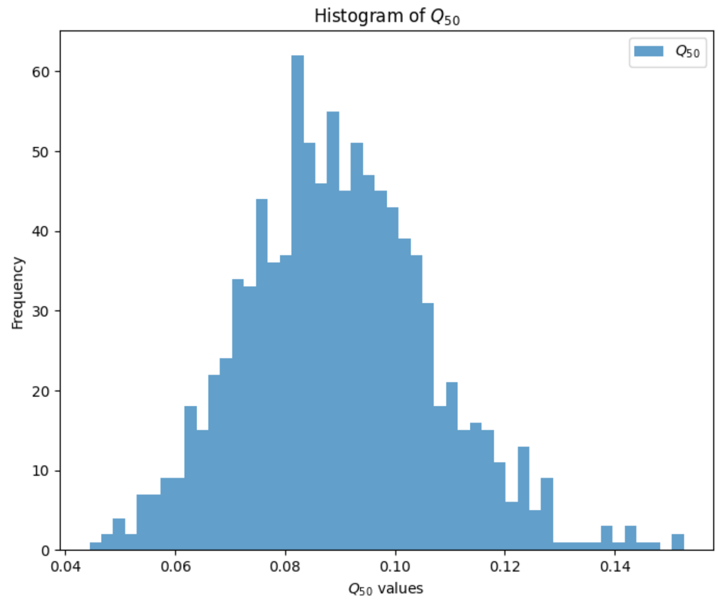
$$\text{Var}(Q_m) = \sum_{j=1}^{m} \sigma^4 \cdot 2(\Delta t)^2 = 2\sigma^2 \frac{T^2}{m}.$$

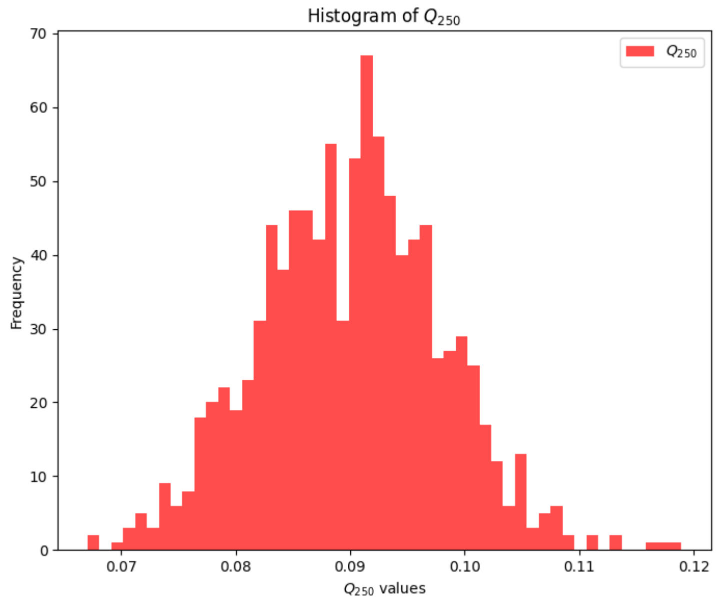
**(b)** The distribution of $Q_m$ approximates a normal distribution due to the Central Limit Theorem (CLT), which states that the sum of a large number of independent random variables, each with finite mean and variance, tends toward a normal distribution, regardless of the original distribution. In the case of $Q_m$, the squared increments of the Ito process $S_t$, defined by $dS = \alpha S dt + \sigma S dW$, are not identically distributed due to the process's multiplicative nature. However, they are independent because the increments of the Wiener process $dW$ are independent. As $m$ increases, the sum of these squared increments forms a large sample of independent random variables. This sum, according to the CLT, tends toward a normal distribution as the number of intervals $m$ grows large.

**(c)**

```
import numpy as np
import matplotlib.pyplot as plt

# Parameters
alpha = 0.10  # Annualized mean rate-of-return
sigma = 0.30  # Volatility
S0 = 100  # Initial stock price
N = 2000  # Number of time-steps
dt = 1 / N  # time-step in years
T = 1  # Total time in years

# Calculating the up and down factors and probability (using Wilmott's method because we want alpha to be mean
u = 1 + sigma * np.sqrt(dt)
d = 1 - sigma * np.sqrt(dt)
p = 0.5 + alpha * np.sqrt(dt) / (2 * sigma)

# Function to simulate a binomial path
```

```python
def simulate_binomial_path():
    path = [S0]
    for _ in range(N):
        if np.random.rand() < p:
            path.append(path[-1] * u)
        else:
            path.append(path[-1] * d)
    return path


# Function to calculate quadratic variation
def calculate_quadratic_variation(path, delta_t):
    m = int(T / delta_t)  # Number of intervals based on delta_t
    delta_index = int(N / m)  # Number of steps per interval
    Q = 0
    for i in range(1, m + 1):
        log_return = np.log(path[i * delta_index]) - np.log(path[(i - 1) * delta_index])
        Q += log_return ** 2
    return Q


# Simulating 1000 random paths
paths = [simulate_binomial_path() for _ in range(1000)]

# Calculating Q_50 and Q_250
Q_50_values = [calculate_quadratic_variation(path, 5 / 250) for path in paths]  # delta_t = 5 days
Q_250_values = [calculate_quadratic_variation(path, 1 / 250) for path in paths]  # delta_t = 1 day

# Plotting the histograms
plt.figure(figsize=(14, 6))

plt.subplot(1, 2, 1)
plt.hist(Q_50_values, bins=50, alpha=0.7, label='$Q_{50}$')
plt.xlabel('$Q_{50}$ values')
plt.ylabel('Frequency')
plt.title('Histogram of $Q_{50}$')
plt.legend()

plt.subplot(1, 2, 2)
plt.hist(Q_250_values, bins=50, alpha=0.7, color='r', label='$Q_{250}$')
plt.xlabel('$Q_{250}$ values')
plt.ylabel('Frequency')
plt.title('Histogram of $Q_{250}$')
plt.legend()

plt.tight_layout()
plt.show()
```




**(d)**

```
import numpy as np

# Given parameters
sigma = 0.30  # Volatility
T = 1  # Total time in years, assuming 250 trading days


# Function to calculate the expected value of Q from derived formula
def calculate_expected_Q(sigma, T):
    res = sigma**2 * T

    return res


# Function to calculate the variance of Q from derived formula
def calculate_variance_Q(sigma, T, m):
    res = 2*(sigma**4)*(T**2)/m

    return res


# Assuming delta_t for Q_50 and Q_250
delta_t_50 = 5 / 250  # 5 days in terms of a year for 250 trading days
delta_t_250 = 1 / 250  # 1 day in terms of a year for 250 trading days

# Example path calculation (using the first path as an example)
path_example = paths[0]

# Calculating the expected value and variance of Q_50 and Q_250 from formula
q_50_mean_form = calculate_expected_Q(sigma, T)
q_50_variance_form = calculate_variance_Q(sigma, T, 50)

q_250_mean_form = calculate_expected_Q(sigma, T)
q_250_variance_form = calculate_variance_Q(sigma, T, 250)

# Empirical average and standard deviation for Q_50 and Q_250
Q_50_mean_emp = np.mean(Q_50_values)
Q_50_std_emp = np.std(Q_50_values)
```

```python
Q_250_mean_emp = np.mean(Q_250_values)
Q_250_std_emp = np.std(Q_250_values)

# Compare the results
print('Q_50 mean (empirical):', Q_50_mean_emp)
print('Q_50 mean (formula):', q_50_mean_form)
print('Q_50 standard deviation (empirical):', Q_50_std_emp)
print('Q_50 standard deviation (formula):', np.sqrt(q_50_variance_form))
print()
print('Q_250 mean (empirical):', Q_250_mean_emp)
print('Q_250 mean (formula):', q_250_mean_form)
print('Q_250 standard deviation (empirical):', Q_250_std_emp)
print('Q_250 standard deviation (formula):', np.sqrt(q_250_variance_form))
```

```
Q_50 mean (empirical): 0.08992040096513583
Q_50 mean (formula): 0.09
Q_50 standard deviation (empirical): 0.017203763033852838
Q_50 standard deviation (formula): 0.018

Q_250 mean (empirical): 0.09008142712316732
Q_250 mean (formula): 0.09
Q_250 standard deviation (empirical): 0.007885638640706778
Q_250 standard deviation (formula): 0.008049844718999243
```

**(e)** Our partition size does not actually have an effect on the accuracy of the quadratic variation, and so $Q_{50}$ and $Q_{250}$ are equally good estimators of the quadratic variation. The reason is that the quadratic variation is a measure of the variance of the process, and the variance is a property of the process itself, not the partition size. Therefore, the quadratic variation is a property of the process, and not of the partition size. But with respect to uncertainty, a larger $m$ results in observations closer to the true expected value.

## Exercise 7

**(a)** Modify your code from Problem Set 1, Exercise 5, so that in one loop through the binomial tree it calculates the price of a European put option as well as of an American put option.

**(b)** In a single graph, plot the price of a European and American put option with $S_0 = \$100, T = 1, \sigma = 45\%, r = 5\%$, as a function of strike $K$ (take strikes at regular intervals \$5 apart).

### Solution

```python
import numpy as np

def binomial_tree_option_pricing(S0, K, T, N, sigma, r, option_type='call', option_style='European'):
    """
    Calculate the European or American call/put option price using a binomial tree model.

    Parameters:
    - S0 (float): Initial stock price.
    - K (float): Strike price of the option.
    - T (float): Time to expiration in years.
    - N (int): Number of time steps in the binomial tree.
    - sigma (float): Annualized volatility of the stock.
    - r (float): Annualized risk-free interest rate.
    - option_type (str): Specifies the option type, 'call' or 'put'.
    - option_style (str): Specifies the option style, 'European' or 'American'.

    Returns:
    - float: Calculated price of the option.
    """
```

```
    ## Calculate the size of each time step
    dt = T / N

    ## Calculate the up and down factors
    u = np.exp(sigma * np.sqrt(dt))
    d = 1 / u

    ## Calculate the risk-neutral probability
    q = (np.exp(r * dt) - d) / (u - d)

    ## Initialize the stock price tree
    S = np.zeros((N + 1, N + 1))
    for i in range(N + 1):
        for j in range(i + 1):
            S[j, i] = S0 * (u ** (i - j)) * (d ** j)

    ## Initialize the option value tree
    if option_type == 'put':
        V = np.maximum(K - S, 0)
    else:  ## Call option
        V = np.maximum(S - K, 0)

    ## Backward induction to calculate option price
    for i in range(N - 1, -1, -1):
        for j in range(i + 1):

            ## Calculate the early exercise value
            if option_type == 'put':
                exercise_value = K - S[j, i]
            else:  # Call option
                exercise_value = S[j, i] - K

            ## Calculate the continuation value
            continuation_value = np.exp(-r * dt) * (q * V[j, i + 1] + (1 - q) * V[j + 1, i + 1])

            ## Update the option value at the current node
            if option_style == 'American':
                V[j, i] = max(exercise_value, continuation_value, 0)
            else:  ## European option
                V[j, i] = continuation_value

    return V[0, 0]  ## Return the option price at t=0

## Example usage
S0 = 100
K = 100
T = 1
N = 252
sigma = 0.45
r = 0.05

american_put_price = binomial_tree_option_pricing(S0, K, T, N, sigma, r, option_type='put', option_style='Amer
print("American Put Price:", american_put_price)

european_put_price = binomial_tree_option_pricing(S0, K, T, N, sigma, r, option_type='put', option_style='Euro
print("European Put Price:", european_put_price)

american_call_price = binomial_tree_option_pricing(S0, K, T, N, sigma, r, option_type='call', option_style='Am
print("American Call Price:", american_call_price)
```

```python
european_call_price = binomial_tree_option_pricing(S0, K, T, N, sigma, r, option_type='call', option_style='Eu
print("European Call Price:", european_call_price)
```

```
American Put Price: 15.551980750327091
European Put Price: 15.017397066459594
American Call Price: 19.894454616387513
European Call Price: 19.894454616387513
```

```python
import numpy as np
import matplotlib.pyplot as plt

## Strike prices
K_values = np.arange(75, 126, 5)

## Prices
european_put_prices = [binomial_tree_option_pricing(S0, K, T, N, sigma, r, 'put', 'European') for K in K_value
american_put_prices = [binomial_tree_option_pricing(S0, K, T, N, sigma, r, 'put', 'American') for K in K_value

## Plotting
plt.figure(figsize=(10, 6))
plt.plot(K_values, european_put_prices, label='European Put', marker='o')
plt.plot(K_values, american_put_prices, label='American Put', marker='x')
plt.title('European vs American Put Option Prices')
plt.xlabel('Strike Price ($)')
plt.ylabel('Option Price ($)')
plt.legend()
plt.grid(True)
plt.show()
```