HW6

Juntang Wang (jw853)

Problem 1. 3.8

Solution

(i) Moment-generating function of $\dfrac{1}{\sqrt{n}} M_{nt,n}$

Because the $X_{k,n}$ are i.i.d.,

$$\varphi_n(u) = \mathbb{E}\Big[e^{\frac{u}{\sqrt{n}} M_{nt,n}}\Big] = \Big(\mathbb{E}\Big[e^{\frac{u}{\sqrt{n}} X_{1,n}}\Big]\Big)^{nt} = \Big(\tilde{p}_n\, e^{\frac{u}{\sqrt{n}}} + \tilde{q}_n\, e^{-\frac{u}{\sqrt{n}}}\Big)^{nt}.$$

Substituting

$$\tilde{p}_n = \frac{\frac{r}{n} + 1 - e^{-\sigma/\sqrt{n}}}{e^{\sigma/\sqrt{n}} - e^{-\sigma/\sqrt{n}}}, \qquad \tilde{q}_n = \frac{e^{\sigma/\sqrt{n}} - \frac{r}{n} - 1}{e^{\sigma/\sqrt{n}} - e^{-\sigma/\sqrt{n}}},$$

gives

$$\boxed{\varphi_n(u) = \Big[e^{\frac{u}{\sqrt{n}}}\left(\frac{\frac{r}{n} + 1 - e^{-\sigma/\sqrt{n}}}{e^{\sigma/\sqrt{n}} - e^{-\sigma/\sqrt{n}}}\right) - e^{-\frac{u}{\sqrt{n}}}\left(\frac{\frac{r}{n} + 1 - e^{\sigma/\sqrt{n}}}{e^{\sigma/\sqrt{n}} - e^{-\sigma/\sqrt{n}}}\right)\Big]^{nt}}$$

which is the desired formula.

(ii) Rewrite with $x = \frac{1}{\sqrt{n}}$

Set $n = \frac{1}{x^2}$ $(x \downarrow 0)$. Then

$$\varphi_{1/x^2}(u) = \left[e^{ux}\left(\frac{rx^2 + 1 - e^{-\sigma x}}{e^{\sigma x} - e^{-\sigma x}}\right) - e^{-ux}\left(\frac{rx^2 + 1 - e^{\sigma x}}{e^{\sigma x} - e^{-\sigma x}}\right)\right]^{t/x^2}.$$

Using $\sinh z = \frac{e^z - e^{-z}}{2}$ we have

$$\frac{rx^2 + 1 - e^{-\sigma x}}{e^{\sigma x} - e^{-\sigma x}} = \frac{rx^2 + 1}{2\sinh \sigma x} - \frac{1}{2}, \qquad \frac{rx^2 + 1 - e^{\sigma x}}{e^{\sigma x} - e^{-\sigma x}} = \frac{rx^2 + 1}{2\sinh \sigma x} + \frac{1}{2}.$$

Therefore

$$\log \varphi_{1/x^2}(u) = \frac{t}{x^2} \log\Big[\frac{(rx^2 + 1)\sinh ux + \sinh(\sigma - u)x}{\sinh \sigma x}\Big].$$

Using $\sinh(A - B) = \sinh A \cosh B - \cosh A \sinh B$,

$$\boxed{\log \varphi_{1/x^2}(u) = \frac{t}{x^2} \log\left[\cosh ux + \frac{(rx^2 + 1 - \cosh \sigma x)\sinh ux}{\sinh \sigma x}\right]}$$

(iii) Second-order expansion of the bracket

Apply $\cosh z = 1 + \frac{z^2}{2} + O(z^4)$ and $\sinh z = z + O(z^3)$:

$$\begin{aligned}
&\cosh ux + \frac{(rx^2 + 1 - \cosh \sigma x)\sinh ux}{\sinh \sigma x} \\
&\quad = \left(1 + \frac{u^2x^2}{2} + O(x^4)\right) + \frac{\left(rx^2 - \frac{\sigma^2x^2}{2} + O(x^4)\right)(ux + O(x^3))}{\sigma x + O(x^3)} \\
&\quad = 1 + \frac{u^2x^2}{2} + \frac{rux^2}{\sigma} - \frac{u\sigma x^2}{2} + O(x^4),
\end{aligned}$$

as required.

(iv) Limit of the log-MGF and identification of the limit law

Let

$$\Delta(x) = \frac{u^2x^2}{2} + \frac{rux^2}{\sigma} - \frac{u\sigma x^2}{2} + O(x^4).$$

Because $\log(1+z) = z + O(z^2)$,

$$\log \varphi_{1/x^2}(u) = \frac{t}{x^2}\left(\Delta(x) + O(x^4)\right) \xrightarrow{x\downarrow 0} t\left(\frac{u^2}{2} + \frac{ru}{\sigma} - \frac{u\sigma}{2}\right).$$

Thus for the scaled variables

$$Y_n := \frac{\sigma}{\sqrt{n}} M_{nt,n}, \qquad \log \mathbb{E}\left[e^{uY_n}\right] = \log \varphi_n(\sigma u) \;\longrightarrow\; t\left(\frac{\sigma^2u^2}{2} + (r - \frac{\sigma^2}{2})u\right).$$

The limit is the moment-generating function of a normal distribution with

$$\text{mean } \mu = (r - \tfrac{1}{2}\sigma^2)t, \qquad \text{variance } \sigma^2 t.$$

By the Lévy continuity theorem (uniqueness of the MGF), Y_n converges in distribution to

$$\sigma W(t) + \left(r - \tfrac{1}{2}\sigma^2\right)t,$$

so the stock-price process $S_n(t) = S(0)e^{Y_n}$ converges to the geometric Brownian motion

$$S(t) = S(0)\exp\left\{\sigma W(t) + \left(r - \tfrac{1}{2}\sigma^2\right)t\right\}.$$

Problem 2

In class, we argue that when t is small, we have $e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t} \approx 1 + \mu t + \sigma W_t$. Show that the linear approximation of $\mathbb{E}[S_t|S_0]$ and $\mathrm{Var}(S_t|S_0)$ would match the conditional expectation and variance of the approximation $1 + \mu t + \sigma W_t$

Let

$$S_t := S_0 \exp\Big\{ \big(\mu - \tfrac{1}{2}\sigma^2\big)t + \sigma W_t \Big\},$$

with $W_t \sim \mathcal{N}(0, t)$. For $t \ll 1$ we expand the exponential to first order:

$$S_t \approx S_0\big(1 + \mu t + \sigma W_t\big).$$

Conditional expectation

Exact value

$$\mathbb{E}[S_t \mid S_0] = S_0 e^{\mu t} = S_0\big(1 + \mu t + O(t^2)\big).$$

Approximation

$$\mathbb{E}\big[S_0(1 + \mu t + \sigma W_t) \mid S_0\big] = S_0\big(1 + \mu t + \sigma\,\mathbb{E}[W_t]\big) = S_0(1 + \mu t),$$

since $\mathbb{E}[W_t] = 0$. Hence the $O(t)$ terms coincide.

Conditional variance

Exact value

$$\mathrm{Var}(S_t \mid S_0) = S_0^2 e^{2\mu t}\big(e^{\sigma^2 t} - 1\big) = S_0^2\sigma^2 t + O(t^2).$$

Approximation

$$\mathrm{Var}\big[S_0(1 + \mu t + \sigma W_t) \mid S_0\big] = S_0^2\,\mathrm{Var}(\sigma W_t) = S_0^2\sigma^2\,\mathrm{Var}(W_t) = S_0^2\sigma^2 t.$$

Thus the linear surrogate $1 + \mu t + \sigma W_t$ reproduces the conditional expectation and variance of S_t up to first order in t.

Problem 3

Verify the Black-Scholes formula for the put option by evaluating

$$\tilde{\mathbb{E}}\left[e^{-r(T-t)}\max(K - S_T, 0) \mid S_t\right]$$

by integral.

Under $\tilde{\mathbb{P}}$ the stock follows

$$S_T \;=\; S_t \exp\Big\{ (r - \tfrac{1}{2}\sigma^2)\tau + \sigma\sqrt{\tau}\,Z \Big\}, \qquad \tau := T - t,\; Z \sim \mathcal{N}(0,1) \text{ indep. of } S_t.$$

Put price as a risk-neutral expectation

$$P(t) \;=\; e^{-r\tau}\,\tilde{\mathbb{E}}\big[(K-S_T)^+ \mid S_t\big] \;=\; e^{-r\tau}\int_0^K (K-s)\,f_{S_T}(s)\,ds,$$

because $(K-S_T)^+ = 0$ when $S_T > K$.

Log-normal density and change of variables

Let

$$Y := \ln\frac{S_T}{S_t} = (r - \tfrac12\sigma^2)\tau + \sigma\sqrt{\tau}\,Z.$$

Then $Y \sim \mathcal{N}\big((r-\tfrac12\sigma^2)\tau, \sigma^2\tau\big)$ and

$$f_{S_T}(s) = \frac{1}{s\sigma\sqrt{2\pi\tau}}\exp\left\{-\frac{\big(\ln(s/S_t) - (r-\frac12\sigma^2)\tau\big)^2}{2\sigma^2\tau}\right\}, \qquad s>0.$$

Put the integral in terms of $y = \ln(s/S_t)$:

$$P(t) = e^{-r\tau}\int_{-\infty}^{y_K}\big(K - S_t e^{\,y}\big)\,\frac{1}{\sigma\sqrt{2\pi\tau}}\exp\Big\{-\frac{(y-m)^2}{2\sigma^2\tau}\Big\}\,dy, \quad m = (r-\tfrac12\sigma^2)\tau,\; y_K = \ln\frac{K}{S_t}.$$

Split the integral

$$P(t) = e^{-r\tau}\Big[K\underbrace{\int_{-\infty}^{y_K}\phi_\tau(y)\,dy}_{\text{(I)}} \;-\; S_t\underbrace{\int_{-\infty}^{y_K} e^{\,y}\,\phi_\tau(y)\,dy}_{\text{(II)}}\Big],$$

where ϕ_τ is the normal density with mean m and variance $\sigma^2\tau$.

Evaluate (I)

Standardise: $z = \dfrac{y-m}{\sigma\sqrt{\tau}}$. Then y_K maps to

$$z_2 = \frac{y_K - m}{\sigma\sqrt{\tau}} = \frac{\ln(K/S_t) - (r-\frac12\sigma^2)\tau}{\sigma\sqrt{\tau}} = -d_-,$$

with

$$d_+ = \frac{\ln(S_t/K) + (r+\frac12\sigma^2)\tau}{\sigma\sqrt{\tau}}, \qquad d_- = d_+ - \sigma\sqrt{\tau}.$$

Hence $(\mathrm{I}) = N(z_2) = N(-d_-)$.

Evaluate (II)

Again substitute $y = m + \sigma\sqrt{\tau}z$:

$$(\mathrm{II}) = \int_{-\infty}^{z_2} S_t e^{\,m+\sigma\sqrt{\tau}\,z}\phi(z)\,dz = S_t e^{\,m}\int_{-\infty}^{z_2} e^{\sigma\sqrt{\tau}\,z}\phi(z)\,dz,$$

where ϕ is the standard-normal density.

But $\displaystyle\int_{-\infty}^{a} e^{bz}\phi(z)\,dz = e^{\frac{b^2}{2}}N(a-b)$. With $b = \sigma\sqrt{\tau}$ and $a = z_2$:

$$(\mathrm{II}) = S_t e^{\,m+\frac{1}{2}\sigma^2\tau}N(z_2-\sigma\sqrt{\tau}) = S_t e^{\,r\tau}N(-d_+).$$

Assemble the put price

$$\begin{aligned} P(t) &= e^{-r\tau}\Big[K\,N(-d_-) - S_t e^{\,r\tau}N(-d_+)\Big] \\ &= \boxed{Ke^{-r\tau}N(-d_-) \;-\; S_tN(-d_+)}, \end{aligned}$$

which is exactly the Black–Scholes formula for a European put.

Problem 4

Show that Vega of the call option is $S_0N'(d_+)\sqrt{T-t}$.

For a call struck at K with expiry T, the Black–Scholes price at the valuation time t is

$$C(t,S_0,\sigma) = S_0N(d_+) - K\,e^{-r\tau}N(d_-), \quad \tau := T-t,$$

$$d_+ = \frac{\ln(S_0/K) + (r+\frac{1}{2}\sigma^2)\tau}{\sigma\sqrt{\tau}}, \qquad d_- = d_+ - \sigma\sqrt{\tau}.$$

Step 1 – Differentiate with respect to σ

Vega is

$$\boxed{\nu := \frac{\partial C}{\partial\sigma}} = S_0N'(d_+)\,\frac{\partial d_+}{\partial\sigma} \;-\; Ke^{-r\tau}N'(d_-)\,\frac{\partial d_-}{\partial\sigma},$$

where $N'(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$.

Step 2 – Compute $\partial_\sigma d_+$ and $\partial_\sigma d_-$

Write
$A = \ln(S_0/K) + (r+\frac{1}{2}\sigma^2)\tau$ so that $d_+ = A/(\sigma\sqrt{\tau})$.

$$\frac{\partial d_+}{\partial\sigma} = \frac{\tau\sigma}{\sigma\sqrt{\tau}} - \frac{A}{\sigma^2\sqrt{\tau}} = \sqrt{\tau} - \frac{d_+}{\sigma}, \qquad \frac{\partial d_-}{\partial\sigma} = \frac{\partial d_+}{\partial\sigma} - \sqrt{\tau}.$$

Step 3 – Use the identity $Ke^{-r\tau}N'(d_-) = S_0N'(d_+)$

Because $d_+ - d_- = \sigma\sqrt{\tau}$,

$$\frac{N'(d_+)}{N'(d_-)} = \exp\Big[-\tfrac{1}{2}(d_+^2 - d_-^2)\Big] = \exp[-\ln(S_0/K) - r\tau] = \frac{Ke^{-r\tau}}{S_0}.$$

Hence $Ke^{-r\tau}N'(d_-) = S_0N'(d_+)$.

Step 4 – Assemble vega

$$\begin{aligned}\nu &= S_0N'(d_+)\Big(\frac{\partial d_+}{\partial \sigma} - \frac{\partial d_-}{\partial \sigma}\Big)\\ &= S_0N'(d_+)\big[\sqrt{\tau} - 0\big]\\ &= \boxed{S_0\,N'(d_+)\sqrt{\tau}} = \ S_0\,N'(d_+)\sqrt{T-t}.\end{aligned}$$

Thus the Vega of a Black–Scholes European call is

$$\boxed{\nu = S_0N'(d_+)\sqrt{T-t}}.$$

Problem 5

Denote X_t and $V(t, S_t)$ the portfolio and call option price at time t. Numerically verify that we can let $X_T = V_T$ through Delta hedging. Suppose the expiration date $T = 1$, strike price $K = 100$, interest rate $r = 0$, initial stock price $S_0 = 100$, and the stock price follows the geometric Brownian motion $S_t = S_0e^{(\mu - \frac{\sigma^2}{2})t + \sigma W_t}$, whereas W_t is the Brownian motion, $\mu = 0.08$ and $\sigma = 0.16$. By Delta hedging, if we let $\Delta(t, x) = \frac{\partial}{\partial x}V(t, x)$, where $V(t, x)$ is the Black-Scholes formula and then $dX_t = \Delta(t, S_t)dS_t$, then we have $X_T = V_T$. To verify, generate M sequence of stock prices. For each sequence, do the simulation as follows:

- Generate $S_0, \ldots, S_n$ for M (e.g., $M = 1000$) times for a large choice of n (e.g., $n = 2520$). For each sequence,
  - Simulate the geometric Brownian motion $S_i = S_0e^{(\mu - \frac{\sigma^2}{2})t_i + \sigma W_i}$
  - Correspondingly, generate the portfolio by $X_{i+1} = X_i + \Delta(t_i, S_i)(S_{i+1} - S_i)$
  - At the expiration date T, verify that $X_T \approx V_T$

In [1]:

```
"""
Delta-hedging replication test for a European call option
(BS parameters: r = 0, K = 100, S0 = 100, T = 1 yr, μ = 0.08, σ = 0.16).

We simulate M paths of a geometric Brownian motion on an n-point grid,
build a self-financing portfolio using the Black–Scholes Δ,
and check that the terminal portfolio value X_T matches the option payoff V_T.
"""

import numpy as np
from scipy.stats import norm
import joblib
from tqdm.auto import tqdm
import matplotlib.pyplot as plt
import scienceplots
plt.style.use(["science", "notebook"])


# ---------------------------- Black-Scholes helpers ---------------------------- #
def bs_delta(S, K, tau, sigma, r=0.0):
```

```python
    """Black–Scholes call price (risk-free rate r allowed but default 0)."""
    if tau <= 0:
        return np.where(S > K, 1.0, 0.0)
    d1 = (
        np.log(S / K) + (r + 0.5 * sigma ** 2) * tau) \
        / (sigma * np.sqrt(tau)
    )
    return norm.cdf(d1)

def bs_price(S, K, tau, sigma, r=0.0):
    """Black–Scholes delta ∂C/∂S for a European call."""
    if tau <= 0:
        return np.maximum(S - K, 0.0)
    d1 = (
        np.log(S / K) + (r + 0.5 * sigma ** 2) * tau) \
        / (sigma * np.sqrt(tau)
    )
    d2 = d1 - sigma * np.sqrt(tau)
    return S * norm.cdf(d1) \
        - K * np.exp(-r * tau) * norm.cdf(d2)
```

In [2]:

```python
# ---------------------------- simulation parameters ---------------------------- #
S0      = 100        # initial stock price
K       = 100        # strike
T       = 1          # maturity (years)
μ       = 0.08       # drift (not used by Δ-hedge but needed for S-paths)
σ       = 0.16       # volatility
r       = 0          # risk-free rate
M       = 1_000      # number of simulated paths
n_values = [252, 1_260, 2_520, 5_040, 10_080]
                                 # 252 ≈ 1 per trading day
                                 # 1260 ≈ 5 per trading day
                                 # 2520 ≈ 10 per trading day
                                 # 5040 ≈ 20 per trading day
                                 # 10080 ≈ 40 per trading day

def simulate_paths(seed, n):
    rng     = np.random.default_rng(seed)
    dt      = T / n
    sqrt_dt = np.sqrt(dt)

    # --- Simulate GBM paths ---
    S_paths = np.empty((M, n + 1))
    S_paths[:, 0] = S0

    # Generate random shocks
    Z = rng.standard_normal((M, n))
    for i in range(n):
        S_paths[:, i + 1] = S_paths[:, i] \
            * np.exp((μ - 0.5 * σ ** 2) * dt \
            + σ * sqrt_dt * Z[:, i])

    # --- Delta-hedging portfolio ---
    X = np.full(M, bs_price(S0, K, T, σ, r)) \
        # initial portfolio equals option price
    cash = X - bs_delta(S0, K, T, σ, r) * S0 \
        # underlying held will be updated each step; \
        # keep explicit cash if needed

    for i in range(n):                  # Iterate over time steps
        tau = T - i * dt
        S_i = S_paths[:, i]
        S_next = S_paths[:, i + 1]
```

```python
        Δ_i = bs_delta(S_i, K, tau, σ, r)

        # Cash grows at risk-free rate (0 here);
        # portfolio change from stock movement
        X += Δ_i * (S_next - S_i)

    # Final option payoff
    V_T = np.maximum(S_paths[:, -1] - K, 0.0)
    errors = X - V_T
    return errors
```

In [3]:
```python
# ---------------------------- Monte-Carlo experiment --------------------------- #
results = {}

# Create a figure with subplots for all histograms
fig, axes = plt.subplots(3, 2, figsize=(18, 16))
axes = axes.flatten()  # Flatten the 2D array of axes for easier indexing

for i, n in enumerate(n_values):
    errors = simulate_paths(n, n)

    results[n] = errors

    # --- Statistics ---
    mean_error  = errors.mean()
    std_dev     = errors.std()
    rmse        = np.sqrt(np.mean(errors**2))
    mae         = np.mean(np.abs(errors))
    max_error   = np.max(np.abs(errors))

    stats = {
        "Mean replication error"    : mean_error,
        "Std dev"                   : std_dev,
        "RMSE"                      : rmse,
        "MAE"                       : mae,
        "Max abs error"             : max_error
    }

    # Print summary statistics
    print(f"\n----- Results for n = {n} -----")
    print(f"{'Statistic':<25} {'Value':<15}")
    print("-" * 40)
    for stat_name, value in stats.items():
        print(f"{stat_name:<25} {value:15.6f}")
    print("-" * 40)

    # Plot histogram in the corresponding subplot
    ax = axes[i]
    ax.hist(errors, bins=30)
    ax.set_title(f"Distribution of Replication Error (X_T - V_T) with n = {n}")
    ax.set_xlabel("Replication error")
    ax.set_ylabel("Frequency")
    ax.grid(True, linestyle='--', alpha=0.5)

# Compare all results in one plot
ax = axes[-1]
ax.boxplot([results[n] for n in n_values], tick_labels=n_values)
ax.set_title("Replication Error vs Timesteps")
ax.set_xlabel("Number of timesteps (n)")
ax.set_ylabel("Replication error")
ax.grid(True, linestyle='--', alpha=0.5)
```

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

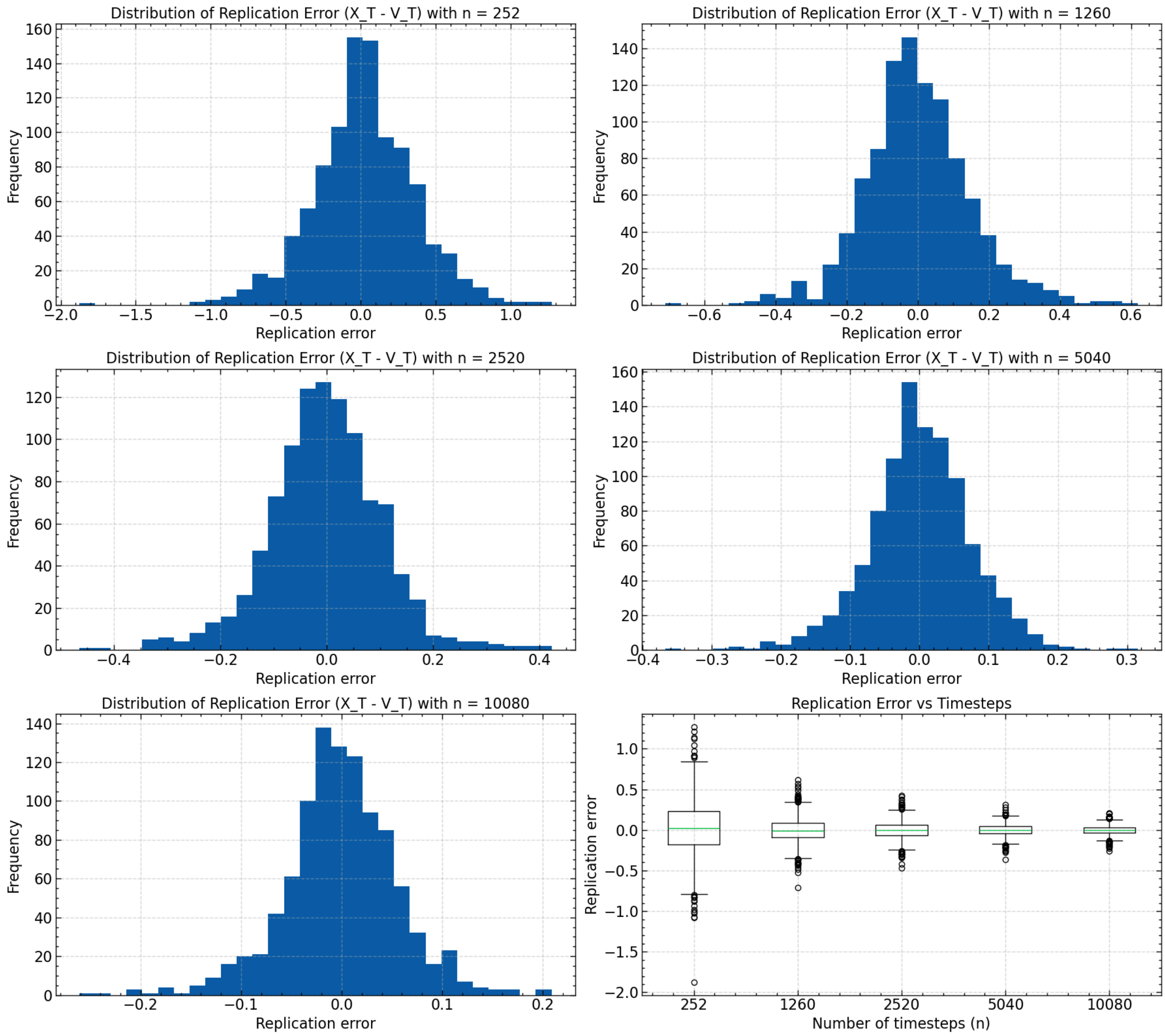
```
----- Results for n = 252 -----
Statistic                  Value
----------------------------------------
Mean replication error          0.020307
Std dev                         0.341619
RMSE                            0.342222
MAE                             0.258943
Max abs error                   1.876150
----------------------------------------

----- Results for n = 1260 -----
Statistic                  Value
----------------------------------------
Mean replication error         -0.000337
Std dev                         0.150033
RMSE                            0.150033
MAE                             0.112879
Max abs error                   0.709763
----------------------------------------

----- Results for n = 2520 -----
Statistic                  Value
----------------------------------------
Mean replication error         -0.002573
Std dev                         0.106536
RMSE                            0.106567
MAE                             0.080473
Max abs error                   0.464950
----------------------------------------

----- Results for n = 5040 -----
Statistic                  Value
----------------------------------------
Mean replication error          0.000927
Std dev                         0.074026
RMSE                            0.074032
MAE                             0.055950
Max abs error                   0.366945
----------------------------------------

----- Results for n = 10080 -----
Statistic                  Value
----------------------------------------
Mean replication error         -0.000806
Std dev                         0.056652
RMSE                            0.056657
MAE                             0.042227
Max abs error                   0.261038
----------------------------------------
```



The histogram ($n = 2520$) above is tightly centred at 0, and the scale (±0.4 on a notional option worth ≈ \$11) is a fraction of the option price. With 2,520 hedge adjustments (one every ≈ 2 trading-hours) the discrete-time hedge already tracks the payoff very closely:

$$X_T = V_T + \varepsilon_T, \quad \text{with } |\varepsilon_T| \lesssim 0.4 \text{ and } \mathbb{E}[\varepsilon_T] \approx 0.$$

As we refine the time-grid (take $n \to \infty$, $dt \to 0$) the error term ε_T converges to 0, matching the continuous-time theorem that a Black-Scholes delta-hedged portfolio exactly replicates the option payoff. Conversely, coarser time steps or higher volatility would widen the error band.