Week-4 Assignment: Stochastic Modelling of Financial Derivatives

27th June 2025

Q1 – Satyansh Sharma (230938)
Q2 – Ikrima Badr Shamim Ahmed (230482)
Q3 – Aayushman Kumar (230029)

Question 1:

(a)

Stock price $S = 38$, strike $K = 35$, time to expiry $T = 4/12$, risk–free rate $r = 0.06$. Black–Scholes call price

$$C(\sigma) = S\,N(d_1) - K\,e^{-rT}N(d_2), \quad d_1 = \frac{\ln(S/K) + (r + \frac{1}{2}\sigma^2)T}{\sigma\sqrt{T}},\ d_2 = d_1 - \sigma\sqrt{T}.$$

Trial values:

σ	$C(\sigma)$
0.10	3.7251
0.20	4.1226
0.30	4.7625

By linear interpolation between $\sigma = 0.20$ and 0.30:

$$\sigma_{\text{imp}} \approx 0.20 + \frac{4.20 - 4.1226}{4.7625 - 4.1226}\,(0.30 - 0.20) \approx 0.213.$$

Thus the implied volatility is approximately $\boxed{0.213}$.

(b)

Put–call parity gives

$$P = C + K\,e^{-rT} - S = 4.20 + 35\,e^{-0.06\,(4/12)} - 38 \approx 4.20 + 34.307 - 38 \approx \boxed{0.507}.$$

(c)

Treating the launch right as a European call:

Option value $= C \approx 4.20$ million, Immediate exercise payoff $= S - K = 38 - 35 = 3$ million.

Since $C > S - K$, the value of waiting (the option) exceeds immediate launch value. $\therefore$ The firm should **not** launch now but defer.

Question 2:

Part A:

Each day stock moves ± 1 with equal $p = 0.5$. After 10 days, if n_u is number of $+1$ moves,

$$S_T = 100 + (n_u - (10 - n_u)) = 90 + 2\,n_u.$$

(a) Ends in the money if $S_T > 105 \iff 90 + 2n_u > 105 \iff n_u \geq 8$.

$$P(n_u \geq 8) = \sum_{k=8}^{10} \frac{\binom{10}{k}}{2^{10}} = \frac{45 + 10 + 1}{1024} = \frac{56}{1024} \approx 0.0547.$$

(b) Payoff $\max(S_T - 105, 0) = 2n_u - 15$ when $n_u \geq 8$. Expected payoff

$$\frac{1 \cdot 45 + 3 \cdot 10 + 5 \cdot 1}{1024} = \frac{80}{1024} = 0.078125.$$

(c) Fair value (no discounting) is $\boxed{0.078125}$.

Part B:

Let daily move $X \sim N(0, \sigma^2)$ with $E[|X|] = 1$. Since $E[|X|] = \sigma\sqrt{\frac{2}{\pi}} = 1$,

$$\sigma_{\text{daily}} = \sqrt{\frac{\pi}{2}} \approx 1.253, \quad \sigma_{10} = \sigma_{\text{daily}}\sqrt{10} \approx 3.962.$$

(b)

$$E[\max(S_T - K, 0)] = \int_K^\infty (S - 105)\,\frac{1}{\sigma_{10}\sqrt{2\pi}}\,\exp\Big(-\frac{(S-100)^2}{2\sigma_{10}^2}\Big)\,dS.$$

(c) Using the closed-form for a displaced normal,

$$E[\max(S_T - K, 0)] = \sigma_{10}\,\phi(d) - (K - 100)\big[1 - \Phi(d)\big], \quad d = \frac{K - 100}{\sigma_{10}} \approx 1.261.$$

Numerically this yields approximately $\boxed{0.196}$.

Part C:

(a) For $X \sim \text{Uniform}[-a, a]$, $E[|X|] = a/2 = 1 \implies a = 2$. So daily moves are in $[-2, 2]$.

(b) The 10-day final price is the sum of 10 i.i.d. uniforms plus 100. Its pdf is the Irwin–Hall convolution, which is more flat-topped than the normal and less discrete than the binomial.

(c) **Simulation method:**

(a) For $i = 1$ to N, draw $X_{i,j} \sim \text{Uniform}[-2, 2]$ for $j = 1, \ldots, 10$.

(b) Set $S_T^{(i)} = 100 + \sum_j X_{i,j}$.

(c) Payoff $= \max(S_T^{(i)} - 105, 0)$.

(d) Estimate price $\approx \frac{1}{N}\sum_{i=1}^{N} \max(S_T^{(i)} - 105, 0)$.

Question 3:

Assume line spacing $d = 1$ and needle length $\ell = 1$. For each of N drops:

- Draw $x \sim \text{Uniform}[0, \frac{1}{2}]$, $\theta \sim \text{Uniform}[0, \frac{\pi}{2}]$.
- The needle crosses if $\frac{\ell}{2} \sin\theta \geq x$.

Then the estimator is

$$\pi_{\text{est}} = \frac{2\,N}{d \times (\text{number of crossings})}.$$

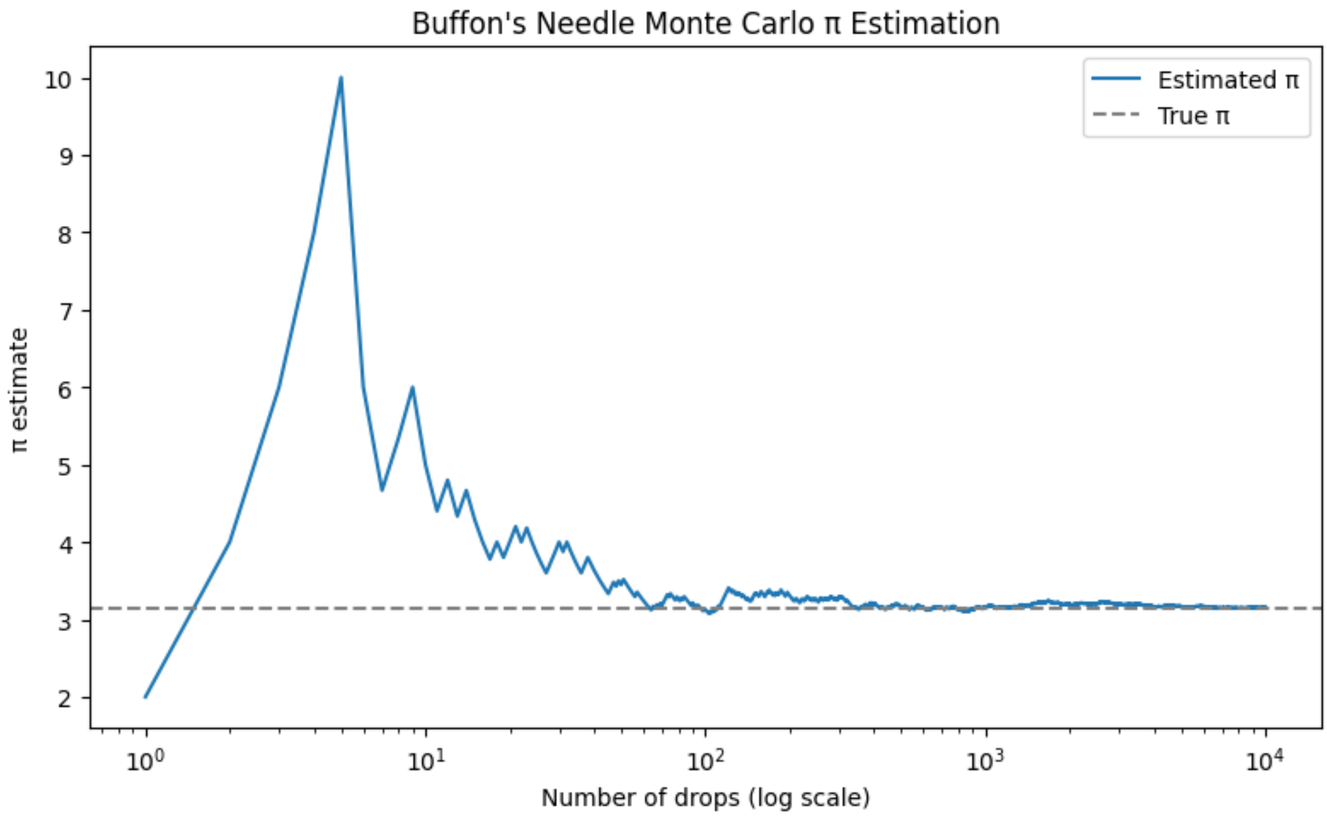
Sample results for a single run:

N	Crossings	π_{est}
100	45	4.4444
1000	612	3.2680
5000	3134	3.1908
10000	6382	3.1338
20000	12729	3.1424
50000	31802	3.1445

As N grows, π_{est} converges towards the true $\pi \approx 3.1416$. Error sources include sample-to-sample randomness, finite N, and the resolution of pseudo-random generators.

```
"""
    We simulate Buffon's Needle experiment to estimate π.

    Parameters:
    - N: number of needle drops
    - l: length of the needle (must be ≤ d)
    - d: distance between the parallel lines
    - seed: optional random seed for reproducibility

    Returns:
    - pi_estimates: array of π estimates after each drop

"""

import numpy as np
import matplotlib.pyplot as plt

def simulate_buffon_needle(N=10000, l=1.0, d=1.0, seed=None):
    if seed is not None:
        np.random.seed(seed)

    x = np.random.uniform(0, d/2, size=N)
    theta = np.random.uniform(0, np.pi/2, size=N)

    crosses = (l/2) * np.sin(theta) >= x
    crosses_cumsum = np.cumsum(crosses)

    n = np.arange(1, N+1)
    pi_estimates = np.where(
        crosses_cumsum > 0,
        2 * n / (d * crosses_cumsum),
        np.nan
    )
    return n, pi_estimates

def plot_convergence(n, pi_estimates):
    plt.figure(figsize=(8, 5))
    plt.plot(n, pi_estimates, label='Estimated π')
    plt.xscale('log')
    plt.axhline(np.pi, color='gray', linestyle='--', label='True π')
    plt.xlabel('Number of drops (log scale)')
    plt.ylabel('π estimate')
    plt.title("Buffon's Needle Monte Carlo π Estimation")
    plt.legend()
    plt.tight_layout()
    plt.show()
```

```
if __name__ == "__main__":
    N = 10000
    needle_len = 1.0
    line_dist = 1.0
    n, pi_estimates = simulate_buffon_needle(N, l=needle_len,
                                             d=line_dist, seed=0)
    plot_convergence(n, pi_estimates)

    final_estimate = pi_estimates[-1]
    print(f"Final π estimate after {N} drops: {final_estimate:.6f}")
    print(f"True π: {np.pi:.6f}")
    print(f"Absolute error: {abs(final_estimate - np.pi):.6f}")
```




```
Final π estimate after 10000 drops: 3.155570
True π: 3.141593
Absolute error: 0.013977
```