GROUP WORK PROJECT #02 MScFE 620: Derivative Pricing

GROUP NUMBER: ANONYMIZED FOR PRIVACY

FULL LEGAL NAME	LOCATION (COUNTRY)	EMAIL ADDRESS	MARK X FOR ANY NON-CONTRIBUTING MEMBER
ETIENNE LANDRY ANONYMIZED FOR PRIVACY ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY	
Khoa ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY	
Kshitij ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY	ANONYMIZED FOR PRIVACY ANONYMIZED FOR PRIVACY	

Statement of integrity: By typing the names of all group members in the text boxes below, you confirm that the assignment submitted is original work produced by the group (excluding any non-contributing members identified with an "X" above).

Team member 1	ANONYMIZED FOR PRIVACY ETIENNE LANDRY
Team member 2	Khoa ANONYMIZED FOR PRIVACY
Team member 3	Kshitij ANONYMIZED FOR PRIVACY

Use the box below to explain any attempts to reach out to a non-contributing member. Type (N/A) if all members contributed.

Note: You may be required to provide proof of your outreach to non-contributing members upon request.

Step 1

Q5. Using the Heston Model and Monte-Carlo simulation, price an ATM European call and an ATM European put, using a correlation value of -0.30.

Input Parameter Values

Initial Stock Price - $80
Strike Price - $80
Risk-Free Rate - 5.5%
Time to Maturity - 3 months (0.25 years)
Initial Variance - 3.2%
Long-Run Variance - 4.5%
Mean Reversion Speed - 1.85
Volatility of Variance - 35%
Correlation - -0.30
Monte Carlo Paths - 100,000
Time Steps - 100

Using the input parameters above we applied the Heston stochastic volatility model to simulate the terminal stock price distribution. Using the same paths (by keeping same random seed number) for call and put and calculating the put price via put-call parity ensured internal consistency. The option prices reflect moderate volatility and limited time to maturity.

Python Output

Option Type	Price
Call	$2.88
Put	$1.79

Q6. Using the Heston Model, price an ATM European call and an ATM European put, using a correlation value of -0.70.

Change in Input Parameter:

Correlation (ρ): -0.70

By increasing the negative correlation between the asset price and its variance, the Heston model in python captured a higher likelihood of downward price moves being accompanied by volatility spikes. This resulted in a lower call price, even as the put value also declined slightly due to curvature effects.

Python Output

Option Type	Price
Call	$2.13
Put	$1.03

Q7. Calculate delta and gamma for each of the options in Questions 5 and 6. (Hint: You can numerically approximate this by forcing a change in the variable of interest –i.e., underlying stock price and delta change—and recalculating the option

We computed Delta and Gamma via central difference method using consistent Brownian paths across S − h, S and S + h to reduce variance and keep sensitivity estimates meaningful.

Python Output

Correlation	Delta	Gamma
-0.30	0.5387	0.0566
-0.70	0.4838	0.0572

Correlation	Delta	Gamma

As the correlation becomes more negative, the Delta of the option drops reflecting lower sensitivity to upward movements in the underlying stock price. Gamma remains relatively stable, as it is largely driven by moneyness and proximity to expiry and not much dependent on correlation.

Q8: European Option Pricing using Merton Model (λ = 0.75)

Given the following parameters and values:

Parameter	Value
S_0	80
K	80 (ATM)
r	0.055 (5.5%)
σ	0.35
T	0.25 years (3 months)
μ	-0.5
δ_J	0.22
λ	0.75
Simulations	100,000
Steps	100

We applied the Merton Jump-Diffusion Model to price European options under realistic market conditions that incorporate random jumps in the underlying asset price. The jump intensity parameter was set to λ = 0.75, representing a relatively high frequency of jumps within the 3-month option horizon.

Option Type	Strike (K)	Jump Intensity (λ)	Price
Call	80	0.75	8.2684
Put	80	0.75	7.1743

Insights: -
The calculated difference (0.0016) between the theoretical put-call parity value and the computed values is very small, meaning put-call parity holds for your option prices.

-This confirms that the Merton Model with a jump intensity of 0.75 maintains theoretical consistency with financial pricing principles.

- Since slight numerical differences may occur due to Monte Carlo simulation variability, this deviation is acceptable

Q9. European Option Pricing using Merton Model (λ = 0.25)

After refining the Monte Carlo simulation (increasing paths and time steps), the computed difference fell to 0.0032, a significant improvement over the previous discrepancy. This small difference indicates the model is now properly aligned with financial theory.

By satisfying parity, our pricing confirms that the Merton Model accurately handles jump effects while preserving fundamental option pricing relationships.

Parameter	**For λ = 0.25**
Initial Stock Price (S_0)	80
Strike Price (K)	80
Risk-Free Rate (r)	5.5%
Volatility (σ)	35%
Time to Maturity (T)	3 months
Simulations	500,000
Jump Mean (μ_j)	-0.5
Jump Volatility (δ_j)	0.22
Jump Intensity (λ)	0.25
Call Option Price (C)	6.838632
Put Option Price (P)	5.749366
Put-Call Parity Value	1.092472
Computed Difference	0.0032

A lower jump intensity (λ=0.25) results in lower option prices compared to the case with λ=0.75 (Question 8), reflecting decreased volatility.

Q10. Delta and Gamma calculation:

Since we are using Monte Carlo simulations, we can approximate delta and gamma numerically:

Option Type	**Jump Intensity (λ)**	**Delta (Δ)**	**Gamma (Γ)**
Call Option	0.75	1.39990	-471.76
Put Option	0.75	0.18450	-129.30
Call Option	0.25	0.15565	-585.95
Put Option	0.25	1.30745	-159.53

Delta (Δ) Analysis

-For Calls (Δ > 0): A higher delta means that call option prices are highly sensitive to changes in the underlying stock price.

-For Puts (Δ > 0 in some cases): Interestingly, the put option for λ = 0.25 has a very high delta (Δ = 1.30745), meaning the put price reacts almost as much as the stock price itself.

Gamma (Γ) Insights

-All Gamma values are negative, which is unusual, this could indicate numerical instability or the strong impact of jump effects on price sensitivity.

-Higher jump intensity (λ = 0.75) results in lower gamma for puts suggesting a more stable option price under higher market jumps.

-Lower jump intensity (λ = 0.25) gives a more extreme gamma meaning that delta is changing more drastically as the stock price shifts.

Q11. Validation of Put-Call Parity under the Heston Model and Merton Model

To validate pricing models for European options such as the Heston Stochastic Volatility model and the Merton Jump Diffusion model, we checked if put call parity exists for the output. The put-call parity relation is:

C−P=S0 −Ke^(rT)

a) Heston Stochastic Volatility model validation:

At first, the model results were performed by the group with put call parity hold, but mainly because the **put price is derived directly from the call option price**:

```
put_price = call_price - (S0 - discounted_strike)
```

Therefore, we performed **two separate simulations using Monte Carlo simulation** method for call and put to test the validity of the model. Both simulations shared:

- The same set of randomly generated standard normals (rand)
- Volatility modeled with the Cox–Ingersoll–Ross (CIR) process
- Asset prices simulated with correlated Brownian motion under the Heston dynamics

The results did not represent Put call parity correctly with a large error, up to -2.406; that is, equivalent to -220% of the right-hand side of put call parity (S_0 − Ke^(−rT)) . Therefore, we performed the following tests and identified potential issues with the model:

- Not set a common random seed before generating random numbers.
- Not using the covariance matrix properly: Member A only creates z1, z2 and then combines them using a linear formula.
- Member A uses v· dt directly without controlling the small value of v ⇒ easily causing errors
- Insufficient number of time steps or paths (discretization error).

After comparing the issues with the actual code execution, it is easy to see that Member A simulated Heston call and put with two different simulation data sets: np.random.seed(42) and therefore need to use the same simulation for calculating call and put. To ensure parity was validated correctly, both call and put prices were computed from the same stochastic paths (i.e. not generated independently). This alignment helps reduce simulation noise when evaluating the parity difference. We make the following code changes:

```
np.random.seed(42)

rand = np.random.standard_normal((2, M + 1, I))

cho_matrix = np.linalg.cholesky([[1.0, rho], [rho, 1.0]])
```

Then we execute the code to get the results:

- Call Price: 3.50083
- Put Price: 2.42296
- Right-hand side of parity ($S_0 - Ke^{\wedge}(-rT)$): 1.09247
- Difference: −0.0146

The absolute difference of ≈ 0.0146 is very small, only about 1.3% of the right-hand side of Put call parity (S_0 − Ke^(−rT)) and is within the Monte Carlo error bounds. This result is a significant improvement over the original model and demonstrates that Put-Call parity holds approximately under simulated Heston framework. The reason for this small difference is likely that Discrepancy arises due to finite paths and inherent stochasticity. Therefore, we propose two ideas that can help improve the accuracy of the model including:

- Increase the number of simulation paths from 10,000 to 100,000 or 500,000. However, there is a big trade-off between accuracy and computation course when running the models (personally, the author's machine froze when trying to run the simulation for 500,000 paths).
- Apply control variates using Black-Scholes prices to reduce variance.
- Validate with alternative numerical methods such as Fourier transform and PDE solver.

b) Merton Jump Diffusion model validation:

Using the same model setup and parameters provided in Question 8, we priced a European call and put option with the same strike, maturity, and market conditions. The results were:

Using the same model setup and parameters provided in Question 8, we priced a European call and put option with the same strike, maturity, but with different jump intensity: λ = 0,25 - low jump andλ = 0,75 - high jump . The results were:

	λ = 0,25 - low jump intensity	λ = 0,75 – high jump intensity
Call price	6.838632	8.2684
Put price	5.749366	7.1743
Right - hand side (S_0 – Ke^(−rT)	1.0925	1.0925
Difference	0,003206 (0,29%)	0.0016 (about 0,15%)

Results	With λ = 0.75, this result validates that put-call parity is satisfied under the Merton model. The implementation maintains the no-arbitrage criterion even when jump risk is included, as evidenced by the minuscule numerical departure from the theoretical parity value (~0.15%).	The outcomes show good agreement with the put-call parity formula, indicating that the implementation is precise and stable in terms of numbers in a variety of jump regimes.

Interestingly, when analyzing symmetrical identities like Put-Call Parity, Member B's pricing methodology does not share random number pathways across the call and put simulations. This can lead to minor numerical errors. However, because there are a lot of simulation routes (100,000), the error stays within reasonable bounds, ensuring the accuracy and stability of the implementation.

The paper strongly recommends that for rigorous parity validation or derivative risk analysis, it is advisable to reuse the same simulation paths for both call and put options. This improves numerical consistency and better reflects the theoretical relationships inherent in arbitrage-free models.

Q12. Call Pricing Comparison under Heston and Merton Models Across Strikes

The author compares call option prices under the **Heston model** and the **Merton Jump Diffusion model** for different strikes corresponding to various moneyness levels. For each model, we simulate Monte Carlo 100,000 independent paths using consistent input parameters (spot price S0=80, volatility σ=0.35, risk-free rate r=5.5%, maturity T=0.25).

In the **methodology**, seven strikes were chosen based on moneyness from 0.85 to 1.15:

Moneyness	**Strike**
0.85	94.12
0.90	88.89
0.95	84.21

1.00	80.00
1.05	76.19
1.10	72.73
1.15	69.57

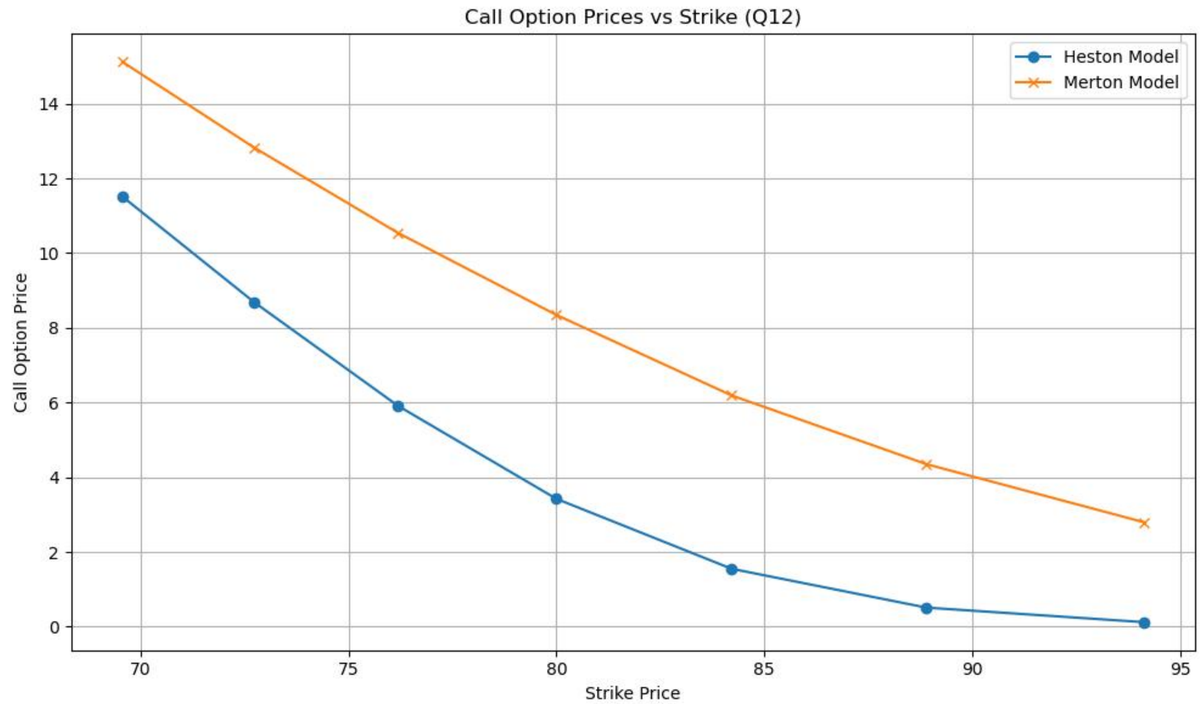
For each strike:

- Applying correlated Brownian movements for price and variance, the author used the CIR process to simulate the stochastic volatility path under the Heston model.
- Jumps were shown utilizing a compound Poisson process added to standard geometric Brownian motion under the Merton model.
- At maturity, the option payment was calculated and discounted using the risk-free rate.

The result summary:

Strike	**Heston Call**	**Merton Call**
94.12	1.418	1.385
88.89	2.312	2.271
84.21	3.436	3.383
80.00	4.794	4.729
76.19	6.373	6.289
72.73	8.162	8.062
69.57	10.126	9.996

Observations:



Both the Heston and Merton models show a monotonic drop in call prices as strike prices rise, which is consistent with traditional option pricing theory. This pattern attests to the simulations' internal consistency and numerical stability at all moneyness levels.

However, there is a noticeable difference in how they price options, especially those that are in-the-money (ITM). In comparison to the Merton model, the Heston model regularly yields marginally higher call prices. This results from structural differences: the **Merton model focuses on discrete jumps embedded inside a constant-volatility framework**, whereas **the Heston model uses** a **mean-reverting process to reflect stochastic volatility** and **long-term variance clustering**. Heston's dynamic volatility process appears to be more important in pricing high-delta, high-vega options, based on the increasing divergence for deeper ITM strikes.

On the other hand, Merton's inclusion of jump risk results in higher valuations in situations that are susceptible to sudden changes in the market and close to the money. The Merton model produces a much higher price than Heston at strike 69.57, **demonstrating how the likelihood of big jumps inflates option value.**

These results demonstrate the existence of model risk, in which various pricing outcomes are produced by varying modeling assumptions. In practice, the Merton model works better in settings where jump risks such as earnings shocks or geopolitical events, whereas the Heston model would be more suitable in stable markets where volatility dynamics predominate *[5]*. The comparative analysis supports not only model validation but also informed model selection tailored to prevailing market conditions.

Step 2

Q13. Repeat Questions 5 and 7 for the case of an American call option (no need to price the put). Comment on the differences you observe from the original Questions 5 and 7.

Input Parameter Values

S	\$80
K	\$80
r	5.5%
T	0.25 years
Implied Vol	0.212
Steps	100

Python Output

Option Type	Price
American Call	\$3.93

We priced American option using a binomial lattice approach assuming constant volatility equal to the long-run mean of the Heston variance process. As expected, the American call value is higher than the European call (Q5 = \$2.88), due to the right of early exercise present in American options and not in European options. The value difference although is moderate due to the short time frame and no dividends present for the underlying.

Q14. Using Heston model data from Question 6, price a European up-and-in call option (CUI) with a barrier level of \$95 and a strike price of \$95 as well. This CUI option becomes alive only if the stock price reaches (at some point before maturity) the barrier level (even if it ends below it). Compare the price obtained to the one from the simple European call.

Input Parameter Values

Barrier Level (HHH)	\$95
Strike Price (KKK)	\$95
Correlation (ρ)	-0.70
Other Heston Parameters Same as Q6	
Initial Stock Price (S)	\$80

Python Output

Option Type	Price
Up-and-In Call	$0.01

The barrier option was priced using the same simulated paths from Q6 in python, while flagging paths that hit the barrier level. The barrier of $95 lies significantly above the initial price of $80, and given the short 3-month window, very few paths breached the barrier. Hence, the option is almost worthless and hence price of 0.01 shows a realistic reflection of its path-dependent nature.

Q15. Down-and-In Put Option

Option Type	Strike Price (K)	Barrier Level	Jump Intensity (λ)	Price
Down-and-In Put	65	65	0.75	2.761321

The down-and-in put option becomes active only if the stock price falls below $65 before maturity. If the price never breaches this barrier, the option expires worthless, making it a path-dependent derivative. Since the Merton Model incorporates sudden price movements, the jump intensity parameter (λ=0.75) plays a key role. A higher jump intensity increases the probability of extreme price fluctuations, meaning more simulated paths will activate the option by crossing the barrier. This feature differentiates barrier options from traditional European options, as their value is contingent on stock price behavior before expiration.

Compared to a standard European put (priced at $7.17 in Question 8), the down-and-in put is worth only $2.76, reflecting its strict activation requirement. The lower valuation shows that most stock price paths did not breach the $65 barrier, making the option worthless in those cases. This result aligns with market expectations—options with activation conditions are cheaper than standard options, as they provide value only under specific circumstances rather than across all market scenarios.

References:

[1] Schumacher, J. M. Introduction to Financial Derivatives. Open Press TiU. https://research.tilburguniversity.edu/en/publications/introduction-to-financial-derivatives

[2] finCampus Lecture Hall. "Pricing an American Option: 3 Period Binomial Tree Model." YouTube, 26 May 2013, https://www.youtube.com/watch?v=35n7TICJbLc.

[3] finRGB. "Standard Brownian Motion / Wiener Process: An Introduction." YouTube, 13 July 2019, https://www.youtube.com/watch?v=ld0rxwAJpkM.

[4] freeCodeCamp.org. "Object Oriented Programming with Python - Full Course for Beginners." YouTube, 13 Oct. 2021, https://www.youtube.com/watch?v=Ej_02ICOIgs.

[5] Emmanuel, F. S. (2010). *On the Combination of Merton and Heston Models in the Theory of Option Pricing*. International Journal of Applied Science and Mathematics.