Derivative Modeling

# Introduction

## What are stocks and derivatives

*Stocks:*

A stock (or a share) is a small ownership of a company that can be freely traded (allows you to buy and sell from other participants like you). Like any other asset, the price of the stock can fluctuate based on the forces of supply and demand. Higher the demand, greater the price; and higher the supply, lower the price. If you buy a stock at, for example, $100 today, and after 3 months, its value rises to $120, you would have made a profit of $20. Likewise, if it went down to $80, you would have made a loss of $20.



*Derivatives:*

In the previous example, you have already purchased the stock, and plan to book your profit or loss after 3 months. Now obviously, you would purchase it in the hope that the price rises so that you can sell at a higher price than what you bought for, but nobody can guarantee what happens 3-months from now, right?!

Theoretically speaking, the stock price can go to 0 at the worst case and you will lose all your money. Now what if there was some way which you could use to floor your losses? This is where derivatives come in. ‘*Options, Futures and Other Derivatives’,* an extremely popular book on derivatives by John C. Hull, defines a derivative to be a financial instrument whose value depends on (or derives from) the values of other, more basic, underlying variables. Very often the variables underlying derivatives are the prices of traded assets. A stock option, for example, is a derivative whose value is dependent on the price of a stock. However, derivatives can be dependent on almost any variable, from the price of hogs to the amount of snow falling at a certain ski resort! Let’s look at some famous derivatives -

*Call Option:*

Going back to our earlier example, your goal is to place a bet that the price of the stock goes up after 3 months, and at the same time, you want to floor your losses. A call option is a contract that gives you the right, but not obligation, to buy the stock at a pre-specified price (called strike price), on a certain date in the future (3 months, in our example), irrespective of the prevailing market price on the expiry date.

Imagine you bought a call option at a price of $10. As the holder of the call option, you now have the right to buy the stock at $100, 3 months from now, even if the market price is greater than $100. If the stock price ends up <= $100, you can let your option expire without any action, and the maximum amount you can lose is the initial price that you paid to buy the option (so the loss is floored)!



*Put option:*

Just as the call option gave the buyer of the option the right to buy the stock at a predetermined price (strike) on the expiry date, a put option will give the buyer of the option the right to sell the stock at a predetermined price (strike) on the expiry date, irrespective of the prevailing market price on the expiry date.

Imagine you bought a put option at a price of $10. As the holder of the put option, you now have the right to sell the stock at $100, 3 months from now, even if the market price is < $100. If the stock price ends up >= $100, you can let your option expire without any action, and the maximum amount you can lose is the initial price that you paid to buy the option.



## 1.2 Pricing a derivative

*Expected Value:*

The expected value (EV) is an anticipated average value for an investment at some point in the future. Mathematically, EV is calculated by multiplying each of the possible outcomes by the likelihood each outcome will occur and then summing all those values. Here’s an example to demonstrate the application of EV in a financial setting:

A unique stock is currently trading at S0. An announcement has been made that the price of this stock three months from now would depend on a coin toss. If the coin toss resulted in ‘Head’ – the final price of the stock would be 1.4 times the initial value and in case of ‘Tail’ – the final price of the stock would be 0.8 times the initial value.

H 1.4S0

S0

T 0.8S0

Now, if you’re given the task to compute the expected payoff of a three-month call option with strike price ‘K’ on this stock– How would you proceed?

*Continuous Compounding:*

You probably would have heard this adage that money today is more valuable than money tomorrow. Ever thought why? Simply because you can invest the money today and earn a risk-free (i.e., FD-like) interest ‘r’, continuously compounded. To give you an insight into what continuous compounding is –

Continuous compounding is the mathematical limit that compound interest can reach if it's calculated and reinvested into an account's balance over a theoretically infinite number of periods. While this is not possible in practice, the concept of continuously compounded interest is important in finance.

The formula for continuous compounding is derived from the formula for the future value of an interest-bearing investment:

Future Value (FV) = PV x [1 + (R / N)](N x T)

where,

* PV – present value of the investment
* R – interest rate
* N – number of compounding periods
* T – time in years

Calculating the limit of this formula as N approaches infinity (per the definition of continuous compounding) results in the formula for continuously compounded interest:

FV = PV x e (R x T)

*Discounting:*

Now that the concept of continuous compounding is clear, let’s say the payoff from holding a call option that you bought at the initial time (T0) is FV at the time of maturity (Tm).

Given an opportunity to time travel to T0 and assuming a risk-free interest rate of ‘R’, how much do you think should be the call option price at time T0?

Shouldn’t it just be the present value ‘PV’ which when compounded at an interest rate of ‘R’ till the time of maturity Tm yields the payoff ‘FV’?

The process of determining the present value of a payment or a stream of payments that is to be received in the future is called discounting.

FV = PV x e (R x T)

PV = FV x e -(R x T) where T is given by (Tm -T0)

PV is obtained by discounting the payoff FV received at time Tm to time T0.

*Geometric Brownian Motion:*

A slight bummer - the price PV we just derived above is not actually what you’ll see in the real world. You see, real-life markets react to events (news) in and around them, which contributes to the price of stocks going up and down. We call this stock price movement uncertainty as “volatility”. [Black-Scholes-Merton](https://www.investopedia.com/terms/b/blackscholes.asp%20%20) (BSM) is a model that provides a closed form solution by factoring in the volatility of the stock’s price into its pricing of call and put options. The stock price dynamics in BSM model are governed by [Geometric Brownian Motion](https://en.wikipedia.org/wiki/Geometric_Brownian_motion#:~:text=A%20geometric%20Brownian%20motion%20(GBM,a%20Wiener%20process)%20with%20drift) (GBM).

What is GBM? – To understand GBM, let’s first understand what a stochastic process is. Any variable whose value changes over time in an uncertain way is said to follow a stochastic process. GBM is a continuous-time stochastic process in which the logarithm of the randomly varying quantity follows a Brownian motion (also called a Wiener process) with drift. Brownian motion here is a phenomenon that we borrow from the world of Physics that describes the random motion of particles in a liquid or a gas.

Why GBM? – While analyzing the graph of the price of a stock, if we try to zoom in as much as possible, we will figure out that we cannot find a point in which the function is all smooth. Whereas in case of deterministic functions, we could always zoom in enough to find a smooth surface eventually. This difference arises due to the randomness embedded in the stock price that makes the graph act like a fractal. This fractal nature of stock prices means that each zoomed in portion is self-similar and using GBM to model stock prices, we could factor in this random component.

The stock price process St, governed by GBM, follows the below stochastic differential equation. Here, dWt is a Weiner process with drift rate of zero and variance rate of 1.0. A generalized Weiner process for variable St can be defined in terms of dWt as

dSt = µStdt + σStdWt

where µ and σ are the stock’s expected rate of return and volatility of the stock price respectively. It turns out that the above stochastic differential equation can be solved explicitly yielding the below unique solution.

St = S0 exp( σWt + ( µ – σ2/2 ) t )

*Derivative Pricing using Simulation:*

Although, BSM is used in pricing of vanilla options such as call and put options, it doesn’t quite prove to be useful in the valuation of options with multiple sources of uncertainty or with complicated features. To overcome this limitation in pricing of exotic derivatives, we resort to a simulation technique known as [Monte Carlo Method](https://www.investopedia.com/terms/m/montecarlosimulation.asp).

Monte Carlo is a probabilistic numerical technique used to estimate the outcome of a given, uncertain process. This means it’s a method for simulating events that cannot be modelled implicitly. Here’s an example employing Monte Carlo simulation technique:

You wish to compute the expected price of a stock three months from now given that the price in three months follows a uniform distribution ~ U [85, 95] dollars. This can be solved analytically as we know that the average of uniform variable is equal to the (higher bound + lower bound)/2 which in this case yields to be 90.

Using Monte Carlo,

You could simulate the price and take the mean of the simulated prices to be the expected price. Greater the number of simulations, more accurate would be the resulting value. The coded solution for this example is given below. You could see that the average price gets closer to the true value 90 as we increase the number of simulations from 100 to 1000.

*Code snippet:*

*>>> import numpy as np*

*>>> simulated\_prices = np.random.uniform(low=85, high=95, size=100)*

*>>> average\_price = simulated\_prices.mean()*

*>>> average\_price*

*90.320322413031079*

*>>> simulated\_prices = np.random.uniform(low=85, high=95, size=1000)*

*>>> average\_price = simulated\_prices.mean()*

*>>> average\_price*

*89.997101194797509*

# Problem statement

## 2.1 Barrier Options

A barrier option is a type of path-dependent option where the payoff is determined by whether or not the price of the stock crosses a certain level **Sb** during its life.

There are two general types of barrier options, **‘in’** and **‘out’** options. In knock-out options, the contract is cancelled if the barrier is crossed throughout the whole life. Knock-in options on the other hand are activated only if the barrier is crossed. The relationship between the barrier **Sb** and the current asset price **S0** indicates whether the option is an up or down option. If **Sb > S0**, we have an up option; if **Sb < S0**, we have a down option.

Combining these features with the payoffs of call and put options, we can define an array of barrier options. For example, a down-and-out put option is a put option that becomes worthless if the asset price falls below the barrier Sb. Therefore, the risk for the option writer is reduced. It is reasonable to expect that a down-and-out put option is cheaper than a vanilla one, since it may expire worthless if the barrier is hit while the vanilla option would have paid off.

*Extra: For a given set of parameters, we can combine an in and an out option of the same type to replicate an ordinary vanilla option. This is due to the fact that when one option gets knocked out, the other is knocked in. Therefore, holding both a down-and-out and a down-and-in put option is equivalent to holding a vanilla put option. The parity relationship can be described as:* ***P = Pdi + Pdo*** *, where P is the price of the vanilla put, and* ***Pdi*** *is the price for the down-and-in option and* ***Pdo*** *is the price of the down-and-out options.*

## 2.2 Cox-Ingersoll-Ross Model

Cox-Ingersoll-Ross (CIR) Model is a stochastic process used to model key financial variables in finance. Some of the major examples are interest rate, credit spreads etc. It is also called mean reverting square root process. Mean Reverting means that on a long term the stochastic value will tend to converge towards the mean. It was introduced in 1985 by John C. Cox, Jonathan E. Ingersoll and Stephen A. Ross as an extension of the Vasicek model.

The process is governed by the following stochastic differential equation:

Here a is called the rate of mean reversion, b is called long term level and  governs the volatility of the model. When we have  , the rate would always remain positive.

When  reaches 0, the stochastic term tends to zero and the drift term tends to move the value towards the long-term level and hence it moves to the positive direction.

These properties make the CIR process ideal to model financial variables like interest rate and default intensities as the value always remain positive.

Now, in a hypothetical world, we will try to use the CIR process to model the price of a stock. This is a very simplifying assumption – generally, the interest rate volatility does not remain constant, and even the stock prices don’t revolve around a static mean. But we’ll go on with that for now.

## 2.3 Task

We aim to find the price of a discretely monitored **up and out put option** with the following parameters, where the underlying stock price is governed by a CIR model.

|  |  |  |
| --- | --- | --- |
| Parameter name | Symbol | Value |
| Spot Price (price today) | **S0** | 20 |
| Strike of put option | **K** | Varies by test case, given below |
| Barrier level | **Sb** | Varies by test case, given below |
| Volatility parameter (sigma) | **σ** | Estimate this |
| Time to expiry (in years) | **τ** | 4 |
| Discount rate | **r** | 0.1 (use continuous decay discounting) |

(Consider a monitoring frequency of **12** per year - at equal intervals in a year, for every year till expiry, i.e., monitoring at time 1/12, 2/12, …, τ)

You can use **any discretization** approach to simulate the process. Note that simulating continuous processes after discretizing them leads to **biases** in the calculations – hence there can be more accurate but complicated schemes. Here, we detail a basic scheme, where we truncate the values when the rates try to go below zero to avoid imaginary values:

### 2.3.1 But wait, what’s a, b and

While working with financial data, you need to predict the behavior of certain characteristics going forward. An example of this is trying to fit real world historical data to our theoretical math and leveraging our knowledge about theoretical concepts to predict real world characteristics. Here, we provide the real stock prices which follow a CIR model.

Generally, we’ll have to prove that a particular timeseries fits a CIR model, but here we skip that part, and you can assume that the values are indeed an output of a CIR modelling process.

You have to calibrate the real-world data and find the governing parameters using some optimization approaches. One such is **ordinary least square** fitting but there can be more complicated approaches. You can write the above discretized equation as:

Where is normal i.i.d (0, 1). This equation can be used for linear regression, but you can try more complicated optimization procedures as well.

You have been provided with an excel sheet which contains 100 paths generated from a CIR process. Use this information to get the parameters of the generating process, .

### 2.3.2 Pricing barrier options

Once you have calibrated a, b and , use the calibrated parameters to perform diffusion, and price barrier options using the following values for **Sb** & **K**, using Monte Carlo simulations to forecast underlying stock prices (much like you did in the main round of the quant challenge) –

|  |  |  |
| --- | --- | --- |
| S.N. | **Sb** | **K** |
| 1 | 140 | 145 |
| 2 | 140 | 150 |
| 3 | 137 | 137 |
| 4 | 139 | 139 |
| 5 | 150 | 200 |
| 6 | 145 | 145 |
| 7 | 132 | 132 |

# Evaluation

To summarize, your task is to calibrate the CIR process and retrieve the values of . Using this information, price the barrier option (according to all 7 configurations above) using Monte Carlo pricing – the final price of the option will be the expected output.

**Please explain your calibration process and the solution to this question in detail providing qualitative as well as quantitative justification to your approach. You’ll get 15 minutes to present your approach and solution to a panel of senior leaders in JPMorgan QR head office (please adhere to the time limit, you’ll be strictly asked to stop once your time ends). At 10 am on the day of the event, you need to submit 3 things: your cpp/python3 code in a doc file, explanation of the code in the same doc file and the ppt. This implementation can flexibly contain as many definitions as required, but it compulsorily needs to contain one definition called priceOption( S\_b, K ), which takes the barrier level (Sb) and strike (K) as inputs, and returns one floating point value as the price of the up and out put option.**

**Your solution will be judged on a combination of accuracy, implementation efficiency, conceptual clarity and the final presentation. All the best, godspeed!**

# References

1. <https://www.investopedia.com/terms/b/blackscholes.asp>
2. <https://en.wikipedia.org/wiki/Geometric_Brownian_motion#:~:text=A%20geometric%20Brownian%20motion%20(GBM,a%20Wiener%20process)%20with%20drift>
3. <https://www.investopedia.com/terms/m/montecarlosimulation.asp>