FQE

Financial Quants and Engineers Club

Fixed Income - Pricing Convertible Bonds Using Monte Carlo Simulation

Alexi T, Alan W, Megumi A

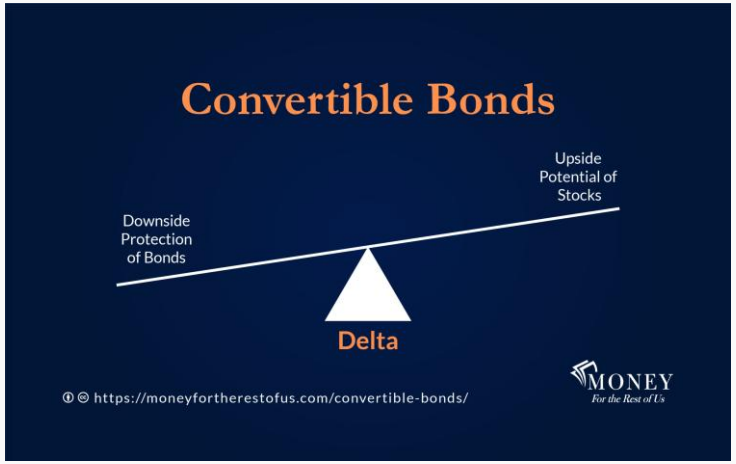
12/12/2025

What Are Convertible Bonds? + Goal

FQE
Financial Quants and Engineers Club

Definition: A convertible bond is a hybrid that combines features of a fixed-income bond and an equity-like option.

- Bond feature (downside protection): Provides fixed income in coupon payments and a face value repayment at maturity, which is what protects it from risk
- Equity feature (upside potential): It gives the bond holders the option to convert the bond into a predetermined number of shares of the issuing company's stock.



When to convert?

- If stock price > conversion price. Converting it will give you a profit because you get shares that are worth more than the bond.
- If stock price < conversion price. Do not convert because you will lose profit

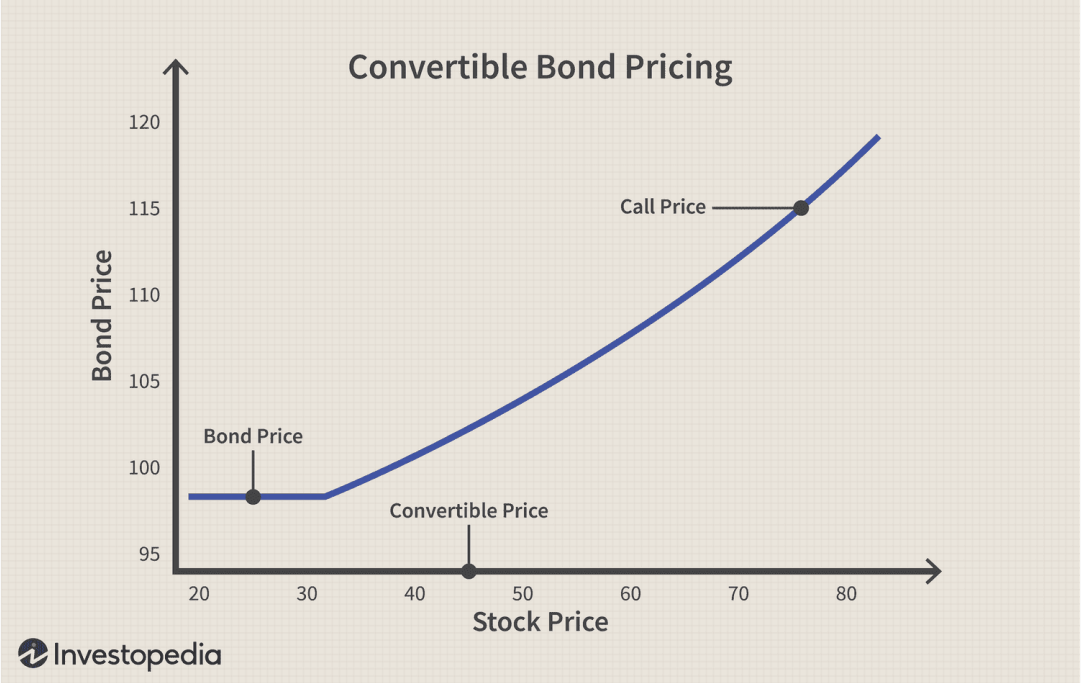
Ex: If conversion price is $50, Stock $70. You will earn $20.

Project Goal

1. Grab data/parameter for a convertible bond
2. Use Geometric Brownian Motion (GBM) to stimulate stock paths
3. Price a convertible bond using Black-Scholes
4. Price the same bond with Monte-Carlo simulation
5. Compare results and compute error

2 types of convertible bonds:

1. European-Style: Conversion can only happen at maturity. This is the simplest to price which is also the main focus of our project
2. American-Style: Conversion can happen at any time before or at maturity. This makes it more complex to price.



Parameters

The parameters that we used are:

1. Geometric Brownian Motion (Wiener Process) for a Stock:

Initial Stock Price (s_nought): Starting price for the Stock

Yearly Trading Days (yearly_td): Number of trading days per year

Drift (Mu): this is the drift or risk free rate, annualized

Volatility (sigma): volatility or diffusion coefficient, annualized

Time Zero (t_nought): start date, in days

Time (T): end date or exercise time T, in days from time zero

2. Black-Scholes Model for (European/American) Convertible Bond

Risk Free Rate (mu): the rate at which an investor can invest with no risk attached

Dividend Yield (q): the yield of the dividend

Conversion Price (K): Price which conversion is allowed to occur

Bond Face Value (N): the face value of the bond

Redemption Ratio (k): the ratio to convert from bond to stock

Converstion Ratio (n): ratio to convert price to number of equity units

Interest Payments (c): the percentage of face value of coupons per year

Coupon Payment Rate (p): this is number of times per year the coupon is paid (1 for annually, 0.25 for quarterly)

3. Simulation Parameters

Number of Walks (num_walks): number of random walks

get_Parameter

Main function: GetParameter(fileName, parameterName)

- Searches the parameter.txt file
- Find the parameter name
- Return a value

```python
"""
This is a program that finds a file in the same folder by name, parses through the file, and returns the

Requirements:
-Data must have an exact match and follow this formating "name" = "value"
"""

def GetParameter(fileName, parameterName):
    foundValue = ""

    if(not isinstance(fileName, str)):
        raise TypeError("The file name must be a string")
    if(not isinstance(parameterName, str)):
        raise TypeError("The Parameter Name must be a string")

    try:
        with open(fileName, 'r') as file:
            for line in file:
                line = line.strip()
                if not line:
                    return None
                if(parameterName in line):
                    parts = line.split("=",1)
                    if len(parts)>1:
                        return float(parts[1].strip())
            return None

    except FileNotFoundError:
        print(f'Error: File {fileName} was not found')
    except Exception as e:
        print(f'An error occured: {e}')

    return foundValue
```

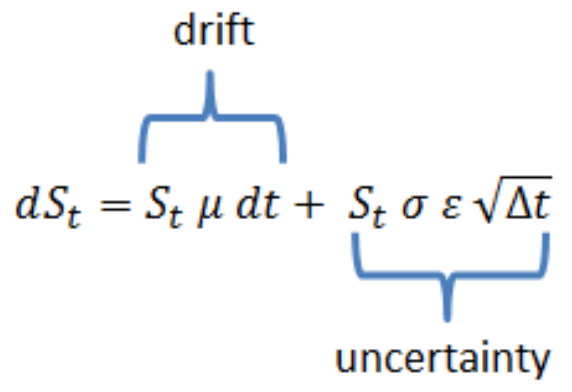
Geometric Brownian Motion (GBM)

In order to price a convertible bond, we have to model the movement of the stock price first.

GBM model is the most suitable for this because it ensures that stock prices remain positive and models their movement as stochastic, or random, process

The GBM formula is composed of two main components:

1. Drift: The predictable average growth rate
2. Volatility: The unpredictable, random fluctuation based on a Wiener process.

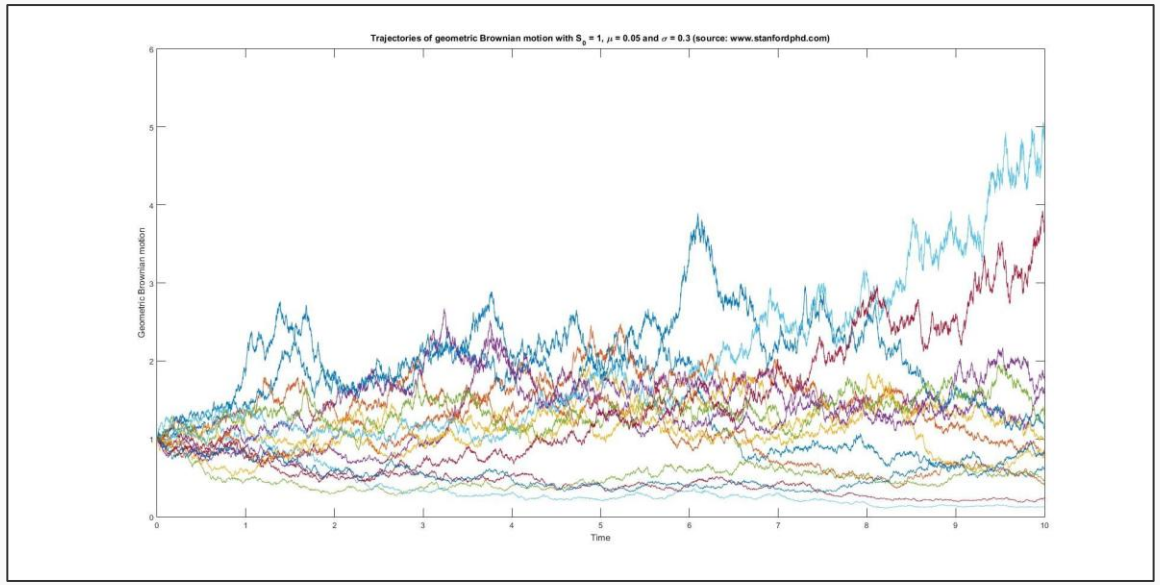


Implementation of Geometric Brownian Motion

FQE
Financial Quants and Engineers Club

Brownian_Motion.py: Implements the mathematical equation for a single step of the stock price movement.

BrownianMotionVisualized.py: This file collects those steps to generate thousands of stock price paths (or "random walks") over time, demonstrating the volatility and the range of possible outcomes. The plots can show the simulated paths and a final price distribution.



Pricing Method 1: Black-Scholes

Formula:

$$C = N(d_1)S_t - N(d_2)Ke^{-rt}$$

$$\text{where } d_1 = \frac{\ln \frac{S_t}{K} + (r + \frac{\sigma^2}{2})t}{\sigma\sqrt{t}}$$

$$\text{and } d_2 = d_1 - \sigma\sqrt{t}$$

Black-Scholes.py: Uses the equation from the research paper

It gives us our **benchmark** which will be compared to our Monte Carlo Price

The Black-Scholes formula is used to close value for the European CB

For a European CB, the price can be broken down into two parts:
CB = Straight Bond Value + Conversion Option Value

The Straight Bond Value is the discounted value of the redemption and any coupons. The Conversion Option Value is equivalent to a European Call Option on the stock

Pricing Method 2: Monte Carlo Simulation

FQE
Financial Quants and Engineers Club

Monte Carlos uses simulation to approximate the expected value instead of going through math equations. The principle used for Monte Carlos is the **Law of Large Numbers**: the average of a large number of random samples will converge to the true expected value.

How to price bond with MC:
1. Simulate Paths with a loop: Simulate thousands of possible stock price paths from today to Maturity using GBM.

2. Calculate payoff: For each path's final stock price, we calculate the payoff at maturity, the maximum of the bond's redemption value or the value if converted to stock.

3. We discount all the payoffs back to today and take the average. This average is our Monte Carlo price estimate.

Code

FQE
Financial Quants and Engineers Club

```python
#Monte Carlo price
def cb_mc(S0, r, sigma, q, T, N, K, n, B, paths=10000, steps=252):
    dt = T / steps
    disc = m.exp(-r * T)
    payoffs = []

    for _ in range(paths):
        S = S0
        for _ in range(steps):
            z = np.random.normal()
            S *= m.exp((r - q - 0.5 * sigma**2) * dt + sigma * m.sqrt(dt) * z)
        # straight bond + conversion option
        payoff = B + n * max(S - K, 0)
        payoffs.append(payoff)

    return disc * np.mean(payoffs)
```

Testing results

FQE
Financial Quants and Engineers Club

For our test run, we used a short maturity (10 days) to quickly test the model.

We used the following as our parameter:

- Initial Stock price: $25.00
- Face Value: $1000.00
- Annual volatility: 5%
- Number of simulation paths: 10,000

We conclude that out model does work since the output will show a low relative error (MC price – BS price)/(BS Price). This validates the Monte Carlo model for a European Convertible Bond

Conclusion

FQE
Financial Quants and Engineers Club

- We successfully implemented the Geometric Brownian Motion model for stock price simulation.
- Used the Black-Scholes model to correctly price the convertible bond
- We validated the use of the Regular Monte Carlo method for pricing the simpler European convertible bond

Future steps:

- Use a real convertible bond data
- Try to use Monte Carlo for American Convertible Bonds

Questions?