Quant Finance Interview

Target Audience:

1. Quant Research Roles
2. Quant Trading Roles

Types of roles

1. Front-end quant
2. Trader
3. Back-end quant
4. Dev
5. Software Engineer

Required Skill Set for Interviews

1. Stochastic Differential Equations
    a. We will cover it
2. Brain-teasers:
    a. We will cover them
3. Probabilities
    a. We will cover them
4. Linear Algebra (finding an inverse of a matrix, solving a system of a linear equations, eigenvectors, vector spaces, linear transformations):
    a. We will cover parts most commonly asked
5. Analysis:
    a. We will cover bits and pieces. It is a vast topic
6. Coding Challenges:
    a. We will cover some challenging ones in form of homework and solutions provided later
7. Fit and Behavioral Interviews:
    a. We won’t be able to cover those

Expected background

1. Mathematics (Linear Algebra, Analysis)
2. Probability theory and a bit of statistics
3. Ability to code in any programming language (C, Java, C++, Python, etc)

Some extra bits

1. We cannot explain and cover all beautiful derivations of equations in class, though I have derived them in advance and detailed derivations are included in slides. Any questions or confusion? Reach out to me by email.
2. We cannot explain simple mathematics in class. If you struggle with Taylor's Polynomial, for example, or how to do matrix operations, wrong class.
3. We will provide a lot of homework to get you ready for quant interviews.
    a. Do NOT use AI to solve them. You will learn nothing.
    b. I will provide detailed solutions for all homework.
    c. If you struggle with a particular question, please feel free to reach out, I can give you some hints to tackle it.
4. Any questions you have, ask: 1. After class 2. Via email. We have very limited time and a ton of topics to cover

Stochastic Differential Equations

1. Brownian Motion
2. Taylor’s Polynomial
3. Ito’s Lemma
4. Ito’s Lemma and its application to stochastic calculus
5. Dynamics of a stochastic process of interest
6. Black Scholes Model
7. Step 1 in Detailed Derivation of Black Scholes
8. Brain Teasers, probability questions
9. Homework

Session 2

1. Derivation of Black Scholes
2. Limitation of Black Scholes
3. Using Black Scholes to calculate European call and put options
4. Volatility Smiles
5. Stylized facts
6. Stochastic Volatility (Heston) and jump Diffusion (Merton) SDE
7. Brain teasers, probability questions
8. Solutions for homework
9. Homework

Session 3

1. Characteristic Functions
2. Fourier Transform and Inverse Fourier Transform
3. Solving SDEs with fourier methods
4. Numerical techniques
5.

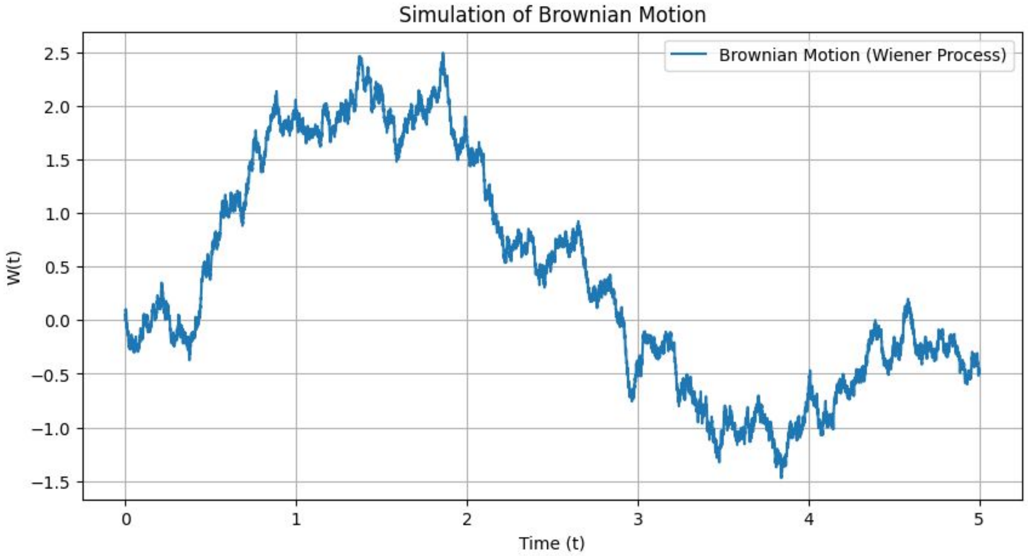
Brownian Motion

A stochastic process is a model on how X evolves over-time.

Examples include the growth of a bacterial population, an electrical current fluctuating due to thermal noise, or the movement of a gas molecule.[1][4][5] Stochastic processes have applications in: biology,[6] chemistry,[7] ecology,[8] neuroscience,[9] physics,[10] image processing, signal processing,[11] control theory,[12] information theory,[13] computer science,[14] and telecommunications.[15] Furthermore, seemingly random changes in financial markets have motivated the extensive use of stochastic processes in finance

Source: https://en.wikipedia.org/wiki/Stochastic_process

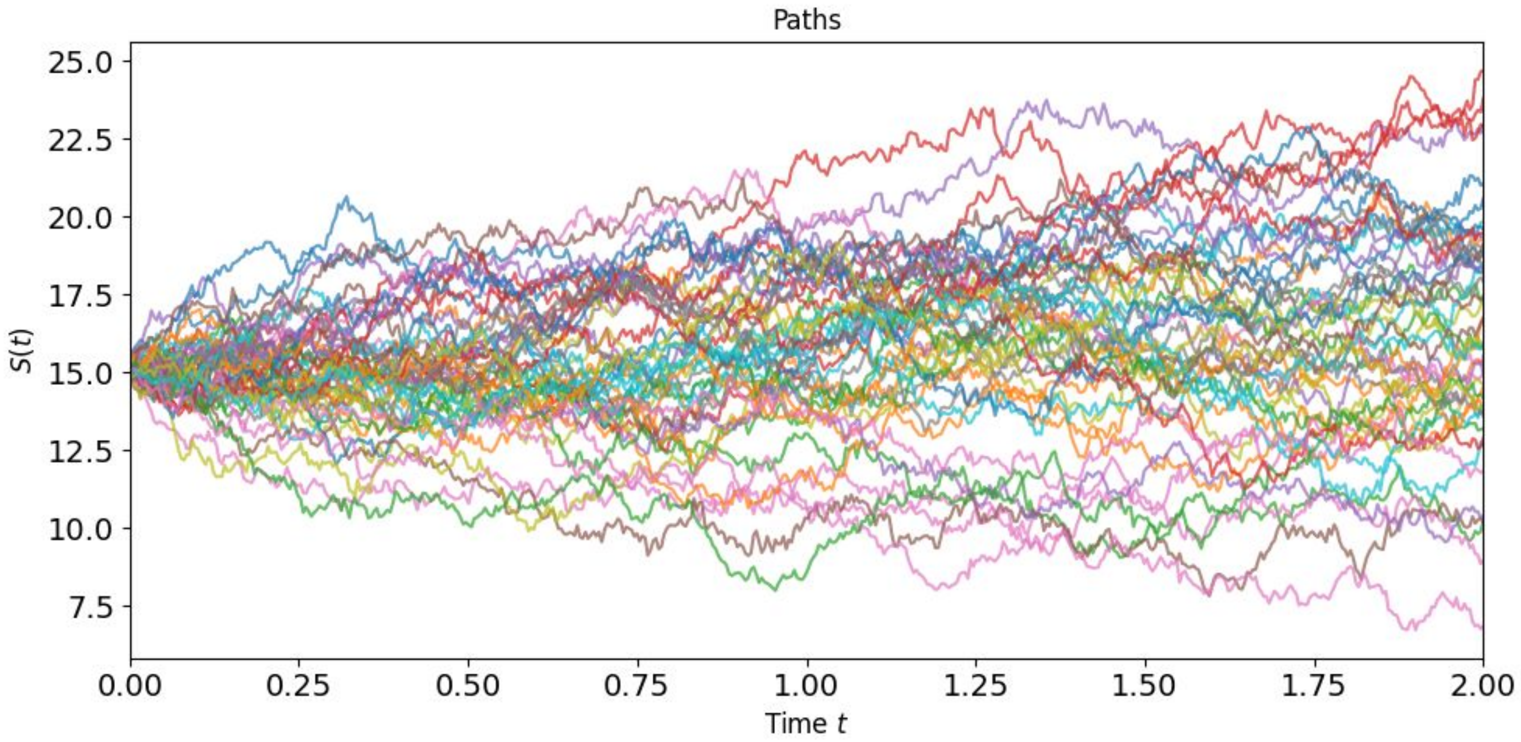
We use Brownian Motion as a way to model evolution of a random variable over time. Brownian Motion is a continuous function with independent increments N(0, dt).



Simulating it in Python

```python
def simulate_gbm(MTC: int, T: float, N: int, S0: float, mu: float, sig: float, seed: int):
    if MTC <= 0 or T <= 0 or N <= 0 or S0 <= 0 or sig < 0:
        raise ValueError("MTC path, T, N, S0 must be positive and sig must be non-negative")
    np.random.seed(seed)

    dt = T / N
    t = np.linspace(0, T, N)

    dS = S0 * (mu * dt + sig * np.sqrt(dt) * np.random.randn(MTC, N))
    S = S0 + np.cumsum(dS, axis=1)

    plt.figure(figsize=(10, 5))
    for i in range(MTC):
        plt.plot(t, S[i, :], alpha=0.7)

    plt.xlabel("Time $t$", fontsize=12)
    plt.ylabel("$S(t)$", fontsize=12)
    plt.title("Paths", fontsize=12)
    plt.xlim([0, T])
    plt.xticks(fontsize=14)
    plt.yticks(fontsize=14)
    plt.tight_layout()
    plt.show()
```

Results with:



At the core of the simulation:

$$dS = S_0\,(\mu dt + \sigma dW_t)$$

Let's unpack GBM SDE a bit more

A risky asset, such as a stock, is expected to provide some return after sometime.

$$dS = S_0 * \mu dt$$

Here we are purposefully ignoring Brownian motion. So, how would this graph look like? (Let's draw it)

Mu here is the slope of the line.

By adding Brownian Motion, we don't change mu, on average, but we add variance to mu. Why?

Because $\Delta W_t \sim N(0, \Delta t)$

So on average there is no change to return of the asset

Δ S = S_0 * [return of asset * Δt + Δ W_t * sigma]

Where Δt is change in time and Δ W_t is change in Brownian motion and sigma is a scaling parameter

$$dS = S_0 \left(\mu dt + \sigma dW_t\right)$$

Polynomials and Vectors Space

A Polynomial of degree K would look like this:

$$P_K = \{p(x) : p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_{K-1}\, x^{K-1}\,, \quad a_i \in \mathbb{R}\}$$

With the standard basis vectors {1, x, x^2, …, x^(k-1)}

Taylor's Polynomial

$$P_n(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(n)}(a)}{n!}(x-a)^n$$

Which is, for our purposes, used for approximating complex functions.

Application of taylor’s Polynomial

$df(x) = f'(x)dx + \frac{1}{2} f''(x).(dx)^2$

Assuming $f(x) = \sin(x)$

$df(x) = \cos(x)\, dx - \frac{1}{2} (\sin(x)).(dx)^2$

Most often the change in x (dx) is very small, making the change in x squared $(dx)^2$ much smaller, hence negligible in calculus. BUT…

Taylor’s Polynomial (Itô's Lemma) in Stochastic Calculus

But … We cannot ignore the change in Brownian motion. Lets see how and why.

Itô’s Lemma for a Function f(W_t)

$$df(W_t) = \frac{\partial f}{\partial W_t} dW_t + \frac{1}{2} \frac{\partial^2 f}{\partial {W_t}^2} (dW_t)^2$$

But if we go back to the graph, we can see that for a small change in time, dW_t changes a lot. Therefore, we cannot ignore (dW_t)^2.

Derivation of Black Scholes

Lets first calculate what (dW_t)^2 is.

Recall that: W ~ N(0, dt).

So variance of the process is dt.

Now:

$$\mathrm{Var}(W_t) = E[{W_t}^2] - (E[W_t])^2$$

Which is the variance formula.

From this we can see that:

$$E[dW_t] = 0, \quad E[dW_t^{\,2}] = dt$$

Again recall that W ~ N(0, dt).

Hence, we have Var(W_t) = dt - 0 = dt where (as shown above (dW_t)^2 = dt).

Continue..

$$df(W_t) = \frac{\partial f}{\partial W_t} dW_t + \frac{1}{2} \frac{\partial^2 f}{\partial {W_t}^2} (dW_t)^2$$

$$dS = S_0 \left(\mu dt + \sigma dW_t \right)$$

We will use, f(S_t) = log(S_t) for simplicity of derivations.

Derivation of Black Scholes

$d \log(S_t) = (\partial \log(S_t)/\partial S_t)\, dS_t + (1/2)(\partial^2 \log(S_t)/\partial S_t^2)\, (dS_t)^2$

$d \log(S_t) = (1/S_t)\, dS_t - (1/2)(1/S_t^2)\, (dS_t)^2$

Where $dS_t = S_t\, (\mu\, dt + \sigma\, dW_t)$ **(from above)**

NOW:

$d \log(S_t) = (1/S_t)\, [S_t\, (\mu\, dt + \sigma\, dW_t)] - (1/2)(1/S_t^2)\, [S_t\, (\mu\, dt + \sigma\, dW_t)]^2$

$= (\mu\, dt + \sigma\, dW_t) - (1/2)\, (\mu\, dt + \sigma\, dW_t)^2$

$= (\mu\, dt + \sigma\, dW_t) - (1/2)\, [\mu^2\, dt^2 + 2\mu\sigma\, dt\, dW_t + \sigma^2\, dW_t^2]$ **Where** $dt^2 = 0$

$= \mu\, dt + \sigma\, dW_t - \mu\sigma\, dt\, dW_t - (1/2)\, \sigma^2 dW_t^2$ **Where** $dt dW_t = 0$ **and** $dW_t^2 = dt$ **and dt**

$= (\mu - \frac{1}{2} \sigma^2)\, dt + \sigma\, dW_t$

$d \log(S_t) = (\mu - \frac{1}{2} \sigma^2)\, dt + \sigma\, dW_t$

Continue

$\int d \log(S_t)$ (from t to T) $= \int [(\mu - \frac{1}{2}\sigma^2)\, dt + \sigma\, dW_t]$ (from t to T)

$\log(S_T)) - \log(S_t) = (\mu - \frac{1}{2}\sigma^2) \int dt + \sigma \int dW_t$

$\log(S_T) / \log(S_t) = (\mu - \frac{1}{2}\sigma^2)(T - t) + \sigma (W_T - W_t)$

$\log (S_T/S_t) = (\mu - \frac{1}{2}\sigma^2)(T - t) + \sigma (W_T - W_t)$

$S_T / S_t = \exp((\mu - \frac{1}{2}\sigma^2)(T - t) + \sigma (W_T - W_t))$

$S_T = S_t * \exp((\mu - \frac{1}{2}\sigma^2)(T - t) + \sigma (W_T - W_t))$

Risk Neutral Valuations

Risk-Neutral Pricing

$$\text{price}_t = \frac{E^*[\text{Payoff}]}{r_f}$$

$$r_f = e^{r(T-t)}$$

Risk-Neutral Expectation of S_T

$$S_T = S_t e^{(r-\frac{1}{2}\sigma^2)(T-t)+\sigma(W_T-W_t)}$$

Now Since we have this property

$$W_T - W_t \sim \sqrt{T-t}N(0,1)$$

we can then substitute it in our equation and get:

$$S_T = S_t e^{(r-\frac{1}{2}\sigma^2)(T-t)+\sigma\sqrt{T-t}N(0,1)}$$

Expectation Integral Representation

The expectation is given by:

$$E^*[f(S_T)] = \int_{-\infty}^{\infty} f(S_T)p(x)dx$$

where the Probability Density Function (PDF) of the Normal Distribution is:

$$p(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$$

Since

$$W_T - W_t = \sqrt{T-t}N(0,1),$$

we substitute:

$$S_T = S_t e^{(r-\frac{1}{2}\sigma^2)(T-t)+\sigma x\sqrt{T-t}}$$

Thus, the expectation integral becomes:

$$E^*[f(S_T)] = \int_{-\infty}^{\infty} f(S_t e^{(r-\frac{1}{2}\sigma^2)(T-t)+\sigma x\sqrt{T-t}}) \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}\, dx$$

Which obviously needs to be multiplied by the discount factor:

$$\exp(-r(T-t))$$

Summary So far

This is the asset price dynamic: $dS_t = S_t (\mu \, dt + \sigma \, dW_t)$

This is the price at time T: $S_T = S_t * \exp[(\mu - \frac{1}{2} \sigma^2)(T - t) + \sigma (W_T - W_t)]$

Where $(W_T - W_t)$ is $N(0, T - t)$ or $\sqrt{T-t} * N(0,1)$

And Black scholes is simply a way to price any derivative using:

1. the asset price dynamics shown above
2. It is the product of integral of Standard Normal Distribution and the derivative function f, depending on the price evolution dynamics shown

So, we can now use a function of this underlying asset price to derive Black Scholes equation for call and put => **This will be done next session**

Time for Brain Teasers

1. Conditional Probability
2. Total Probability
3. Combinatorics
4. Bayes’ Rule
5. Linear Algebra

Requirements for today’s brain-teasers

Appendix

1. Hull, J. C. (2021). *Options, Futures, and Other Derivatives* (11th ed.). Pearson.
2. Schumacher, J. M. (2020). *Introduction to Financial Derivatives: Modeling, Pricing, and Hedging*. Open Press TiU.