Options Part 3 - Black-Scholes Formula and Classical Delta

Understanding the Black-Scholes Formula Intuitively

The Black-Scholes formula is a cornerstone in options pricing, used to calculate the theoretical value of European-style options. Developed by Fischer Black, Myron Scholes, and Robert Merton, it allows us to price options based on certain assumptions about the market and the asset's price behavior.

Key Intuitions Behind the Formula

The formula consists of two main components:

1. **The Expected Asset Price (Adjusted for Probability)**: This is represented by $S \cdot N(d_1)$, which estimates the probability-adjusted value of the asset if the option were exercised.
2. **The Present Value of the Strike Price**: This is represented by $K \cdot e^{-rT} \cdot N(d_2)$, which adjusts the strike price for the risk-free rate and the probability of exercise.

Black-Scholes Formula for European Call and Put Options

For a European call option:

$$C = S \cdot N(d_1) - K \cdot e^{-rT} \cdot N(d_2)$$

For a European put option:

$$P = K \cdot e^{-rT} \cdot N(-d_2) - S \cdot N(-d_1)$$

where:

- S: Current stock price
- K: Strike price of the option
- T: Time to expiration (in years)
- r: Risk-free interest rate
- σ: Volatility of the stock
- $N(\cdot)$: Cumulative distribution function of the standard normal distribution
- $d_1 = \frac{\ln(S/K)+(r+0.5\sigma^2)T}{\sigma\sqrt{T}}$
- $d_2 = d_1 - \sigma\sqrt{T}$

Breaking Down d_1 and d_2

- d_1 represents the probability-adjusted expected return of the asset relative to the strike price, incorporating volatility and the time to expiration.
- d_2 is related to the probability that the option will expire in the money, taking into account the time and volatility.

Why $N(d_1)$ and $N(d_2)$?

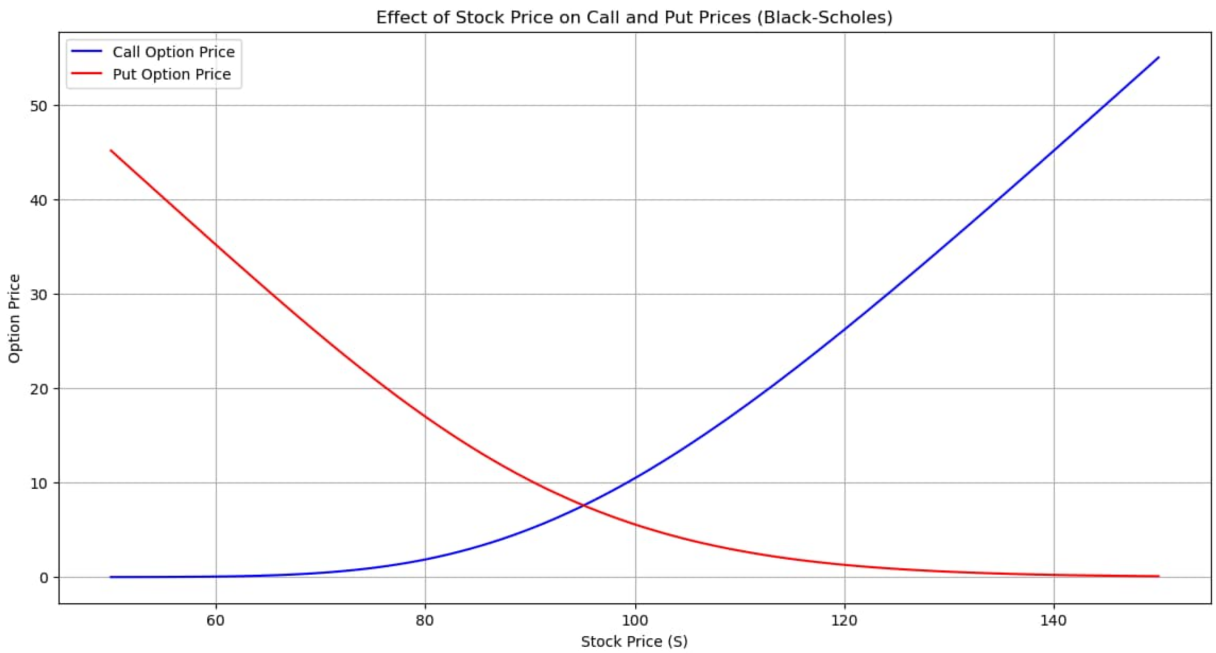
- $N(d_1)$: Gives the probability (in a risk-neutral context) that the option will be exercised, i.e., that the asset price will exceed the strike price.
- $N(d_2)$: Reflects the adjusted probability of payoff at the strike price, factoring in the time value of money.

The formula essentially finds the expected payoff by adjusting the probabilities using the risk-free rate and the asset's volatility.

Let's visualize the effect of varying each of these parameters.

In [1]:
```
# Import necessary libraries
import numpy as np
from scipy.stats import norm
import matplotlib.pyplot as plt

# Define Black-Scholes Formula
def black_scholes(S, K, T, r, sigma, option_type="call"):
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    d2 = d1 - sigma * np.sqrt(T)
    if option_type == "call":
        return S * norm.cdf(d1) - K * np.exp(-r * T) * norm.cdf(d2)
    elif option_type == "put":
        return K * np.exp(-r * T) * norm.cdf(-d2) - S * norm.cdf(-d1)

# Parameters
K = 100     # Strike price
T = 1       # Time to expiration in years
r = 0.05    # Risk-free interest rate
sigma = 0.2  # Volatility

# Plotting the effect of stock price on option prices
S_range = np.linspace(50, 150, 100)
call_prices = [black_scholes(S, K, T, r, sigma, "call") for S in S_range]
put_prices = [black_scholes(S, K, T, r, sigma, "put") for S in S_range]

plt.figure(figsize=(14, 7))
plt.plot(S_range, call_prices, label="Call Option Price", color="blue")
plt.plot(S_range, put_prices, label="Put Option Price", color="red")
plt.xlabel("Stock Price (S)")
plt.ylabel("Option Price")
plt.title("Effect of Stock Price on Call and Put Prices (Black-Scholes)")
plt.legend()
plt.grid(True)
plt.show()
```

```python
# Exploring the effect of volatility on call prices
volatilities = [0.1, 0.2, 0.3, 0.4]
plt.figure(figsize=(14, 7))
for sigma in volatilities:
    call_prices_vol = [black_scholes(S, K, T, r, sigma, "call") for S in S_range]
    plt.plot(S_range, call_prices_vol, label=f"Volatility: {sigma}")

plt.xlabel("Stock Price (S)")
plt.ylabel("Call Option Price")
plt.title("Effect of Volatility on Call Option Prices")
plt.legend()
plt.grid(True)
plt.show()
```



Classical Delta Derived from Black-Scholes

Delta is a measure of an option's price sensitivity to small changes in the price of the underlying asset. It is a critical "Greek" in option trading and risk management.

Derivation of Delta for Call and Put Options

For a European call option, Delta (Δ_{call}) is derived by taking the partial derivative of the option price with respect to the stock price, S:

$$\Delta_{call} = \frac{\partial C}{\partial S} = N(d_1)$$

For a European put option, Delta (Δ_{put}) is given by:

$$\Delta_{put} = \frac{\partial P}{\partial S} = N(d_1) - 1$$

Intuitive Interpretation of Delta

- **Call Options**: Delta for a call option ranges from 0 to 1, indicating the extent to which the option's price will move with a change in the stock price. A Delta close to 1 implies a high sensitivity to the underlying price, meaning the option behaves almost like the stock itself.
- **Put Options**: Delta for a put option ranges from -1 to 0. A Delta close to -1 indicates strong sensitivity to price movements, but in the opposite direction.

Delta can also be interpreted as the probability of the option expiring in-the-money in a risk-neutral context.

Visualization of Delta Behavior

We will plot Delta for call and put options across a range of stock prices to visualize how it changes with respect to the underlying asset.

In [2]:

```
# Function to calculate Delta for call and put options
def delta(S, K, T, r, sigma, option_type="call"):
    d1 = (np.log(S / K) + (r + 0.5 * sigma ** 2) * T) / (sigma * np.sqrt(T))
    if option_type == "call":
        return norm.cdf(d1)
    elif option_type == "put":
        return norm.cdf(d1) - 1

# Calculate Delta for call and put options across a range of stock prices
delta_call = [delta(s, K, T, r, sigma, "call") for s in S_range]
delta_put = [delta(s, K, T, r, sigma, "put") for s in S_range]

# Plotting Delta for Call and Put Options
plt.figure(figsize=(14, 7))
plt.plot(S_range, delta_call, label="Call Option Delta", color="blue")
plt.plot(S_range, delta_put, label="Put Option Delta", color="red")
```

```python
plt.xlabel("Stock Price (S)")
plt.ylabel("Delta")
plt.title("Classical Delta of Call and Put Options over Stock Prices")
plt.legend()
plt.grid(True)
plt.show()
```