Option pricer calculator

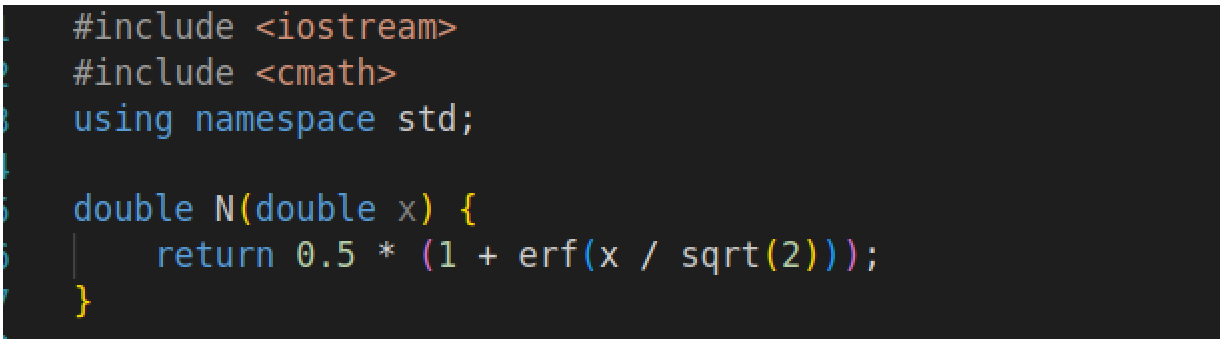
The Black-Scholes model is one of the most important concepts in modern financial theory. The mathematical equation manages to estimate the theoretical value of derivatives based on other investment instruments, taking into account the impact of time and other risk factors. It revolutionized the field by providing a systematic wat to quantify the relationship between stock price, strick price, time to expiration, risk-free interest rate and volatility for European Style options.

In this project, led and supervised by ISFA Finance, we tried to implement a simple program that calculates the price of the option.

Step 1 : Understanding the Black-Scholes model

Based on multiple resources, we exchanged with students in the group about the black-scholes model to fully grap the concept behind the equation.
Some of the resources we used but not limited to :
https://www.youtube.com/watch?v=SL8HDfYYk8Y
https://www.youtube.com/watch?v=XE7FKLfZzBA
https://dokumen.pub/basic-black-scholes-option-pricing-and-trading-6nbsped-1991155433-9781991155436.html

Step 2 : Getting started



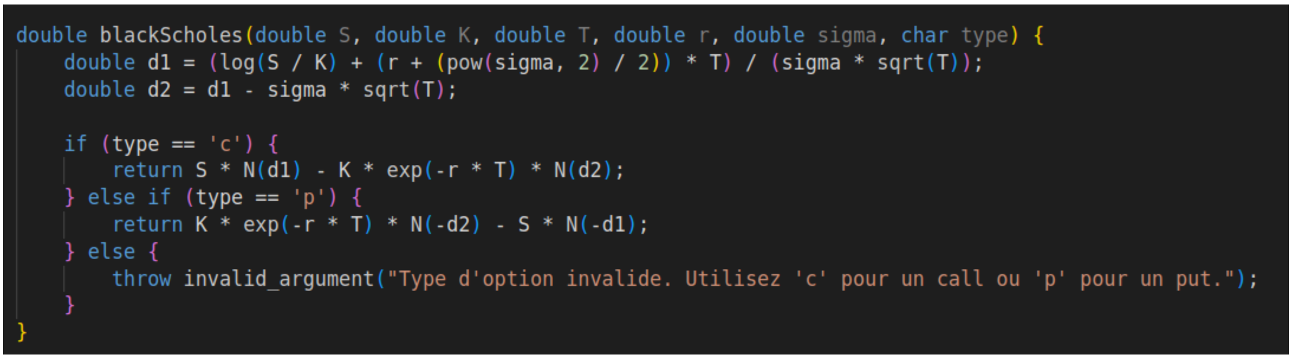
```
#include <iostream>
#include <cmath>
using namespace std;

double N(double x) {
    return 0.5 * (1 + erf(x / sqrt(2)));
}
```

Before implementing the model, we started by setting up our environment and libraries and defining a function that calculates the cumulative distribution function of the standard normal distribution which is conventionally noted as the function N(x).

Option pricer calculator

It uses the erf function to calculate the error function.

Step 3 : Implementing the Black-Scholes formula



```
double blackScholes(double S, double K, double T, double r, double sigma, char type) {
    double d1 = (log(S / K) + (r + (pow(sigma, 2) / 2)) * T) / (sigma * sqrt(T));
    double d2 = d1 - sigma * sqrt(T);

    if (type == 'c') {
        return S * N(d1) - K * exp(-r * T) * N(d2);
    } else if (type == 'p') {
        return K * exp(-r * T) * N(-d2) - S * N(-d1);
    } else {
        throw invalid_argument("Type d'option invalide. Utilisez 'c' pour un call ou 'p' pour un put.");
    }
}
```

we then coded the Black-Scholes option pricing formula, it takes as a first argument a letter c or p respectively call and put to know the type of Option it will calculate and then returns the value.

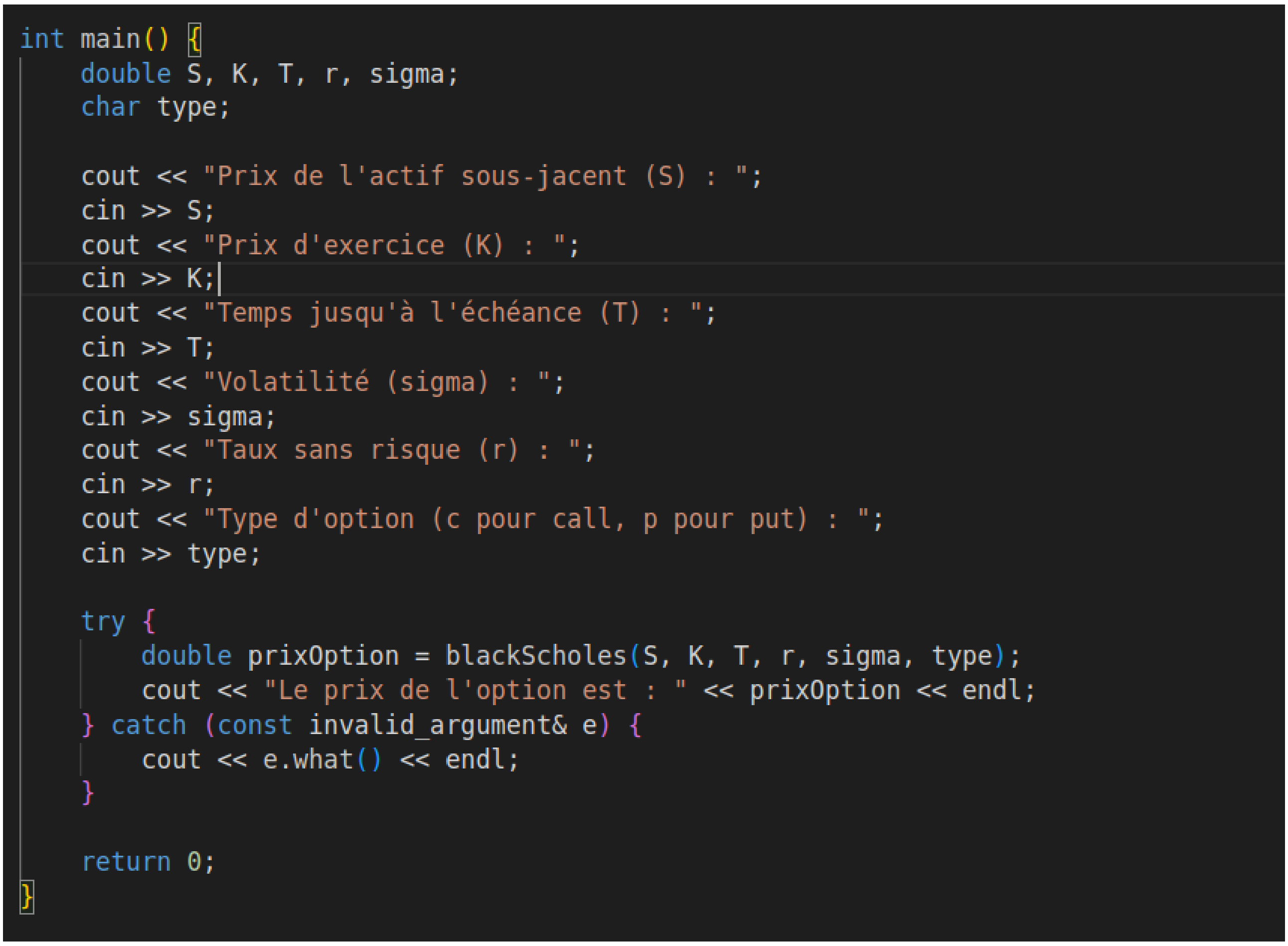
The parameters are the following :

❿ S : current stock price
❿ K : strike price
❿ T : time to expiration
❿ r : risk-free interest rate
❿ sigma : volatility

d1 represents the annualized rate of return or how often does the expected return from the stock exceeds the risk-free rate.
d2 is the same but with an adjustment only for volatility.

Option pricer calculator

Step 4 : testing out the program



```cpp
int main() {
    double S, K, T, r, sigma;
    char type;

    cout << "Prix de l'actif sous-jacent (S) : ";
    cin >> S;
    cout << "Prix d'exercice (K) : ";
    cin >> K;
    cout << "Temps jusqu'à l'échéance (T) : ";
    cin >> T;
    cout << "Volatilité (sigma) : ";
    cin >> sigma;
    cout << "Taux sans risque (r) : ";
    cin >> r;
    cout << "Type d'option (c pour call, p pour put) : ";
    cin >> type;

    try {
        double prixOption = blackScholes(S, K, T, r, sigma, type);
        cout << "Le prix de l'option est : " << prixOption << endl;
    } catch (const invalid_argument& e) {
        cout << e.what() << endl;
    }

    return 0;
}
```

we then code a simple iteration of the code, where you are asked to put in the parameters and the program calculates the price of the option using the functions defined before.

Step 5 : To go further

Option pricer calculator

some possible features to implement to make the program more complex that will be explored later would be :

- Making the program take a specific number of options and compare the prices between them
- Making an API directly gather the parameters online on website such as Yahoo Finance and calculate the price of an option.
- Making the program read the parameters from a csv file and returning option prices.