**How does the program work?**

1. User prompted to enter the name of the underlying file
2. Prompted to enter the minimum strike, then the maximum strike, then the number of estimations
3. Then asked if he wants to use a flat rate (y) or a rates file
4. If the user chooses a flat rate, the last question will be: what rate do you want to use
5. If the user chooses a file, then the last question will be: enter the file name

The csv files for underlying and rates must be in the exact same format and of the same length

Format:

Date,Data : Date must be stored as a double and Data is the stock price for underlying or the rate written as 0.01 for 1%.

**How the program is structured**

1. A class Rate that can take a flat rate or a yield curve thanks to an overload
2. A class Underlying that
   1. The get properties allow us to get one date or one underlying
   2. The read properties allow us to get all the dates, all the underlyings and all the time\_to\_maturity
3. A class CSVrow that represents one row of the CSV file and that allows to go from one row to the the other
4. A class Parser that reads all the csv file and that stocks the data of the two columns(date and underlying or date and rates)
5. A class beVolatilityComputation which is the core of our project
   1. The method PnlComputation computes the PnL
      1. With an overload if a yield curve is given
      2. With an overload with a Boolean that allows us to choose if we want to use the “Black- Scholes Robustness Formula)”
   2. The method midpoint\_algo that uses the midpoint method (dichotomy) to estimate the breakeven volatility
      1. With an overload if yield curve is given
6. Functions
   1. BS Pricer to calculate the price of a call
   2. Delta to calculate the delta of a call
   3. Gamma to calculate the gamma of a call
   4. Normal\_cdf to get the cdf of a standardised normal distribution
   5. Normal\_pdf to get the pdf of a standardised normal distribution

**Results Analysis**

We notice that the breakeven skew with rates at 0% does not seem smooth with some clear jumps but the structure seems consistent as we can notice a left skew. For K between 60 % and 150% the breakeven volatilities are between 3% and 6%. So the area that may represent an interesting arbitrage is the left side. If the implied skew raises faster than the breakeven one we will go short the call option at the relevant strike with k<80% (strike level at which the skew starts its growth)

When we use a libor3m we can notice that the skew is much smoother as it is the closest representation to a real situation. The Breakeven volatility starts to skyrocket almost linearly when k<100%. When the k<=115%, the skew stabilise between 4% and 6%. An interesting arbitrage would be to compare the slope of the skew between k=0% and k=95% for both implied vol and break even vol and then trying to take advantage of the discrepancies between both. For instance, if the gap between both slopes is rising between k=95% and k=0% we can take a position on the most liquid option with the smallest strike possible.

On this example, we realise that if we have a flat rate of 5% our code does not find a solution with a classical Black Scholes but it does find a solution if we use the “Black Scholes Robustness Formula”. We can deduce that the add of the gamma allows to smooth our PnL function and then makes convergence easier.

How to take advantage of the observed skew vs the breakeven skew? :

The breakeven skew may indicate us if the impliedvol is overvalued or undervalued. From this data, we can take suppose that if :

* Breakeven vol> Implied vol, the option is undervalued and that the realized vol should be higher than the implied vol, we will then take a long position on the option
* Breakeven vol< Implied vol, the option is overvalued and that the realized vol should be lower than the implied vol, we will then take a short position on the option