**Q1.** (Code File: q1.py)

**1.1**

Call Option Price: 2.9380121169138036

Put Option Price: 2.6886360765479225

**1.2**

Call Option Price: 0.3870694028577839

Put Option Price: 10.08781815441872

**1.3**

Call Option Price: 4.216659345054804

Put Option Price: 3.7191510325132064

**1.4**

Call Option Price: 4.338822781168002

Put Option Price: 4.089446740802121

**1.5**

Call Option Price: 3.060327056727921

Put Option Price: 2.5628187441863233

Comment on how increase in each parameter affects call and put values (look at appendix for graphs) (Code File: q1\_comments.py):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Option | Strike Price 🡩 | Maturity 🡩 | Volatility 🡩 | Risk Free Rate 🡩 |
| Call | 🡫 | 🡩 | 🡩 | 🡩 |
| Put | 🡩 | 🡩 | 🡩 | 🡫 |

**Q2.**

**2.1**

**2.2** (Code file: q2.py)

Sample Size = 200

Mean\_X: 0.06 Mean\_Y: 0.06 Var\_X: 0.0 Var\_Y: 0.0 Mean\_Z: 0.08 Var Z: 0.0

The value of the correlation coefficient p(X,Z)= 0.48

This value is slightly different from the theoretical value of 0.5. This is because the variables arent a perfect standard normal owing to the small sample size, X has a mean of 0.06 and a variance of 0.0, while Y has a mean of 0.06 and a variance of 0.0

If the number of samples are increased, the distribution will be almost approximate to a standard normal and yield a correlation equal to the theoretical value of 0.5 (as seen below with a much larger sample size)

Sample Size = 20000000

Mean\_X: 0.0 Mean\_Y: 0.0 Var\_X: 0.0 Var\_Y: 0.0 Mean\_Z: 0.0 Var Z: 0.0

The value of the correlation coefficient p(X,Z)= 0.5

**Q3.**

**3.1** Calculating Implied Volatility (Code file: q3-1.py):

Following algorithm has been used

def get\_implied\_volatility(S,K,T,t,r,q,Ctrue,optiontype):

    # Use Newton's method to calculate Implied Volatility

    #starting value

    sigmahat = sqrt(2\*abs( (log(S/K) + (r-q)\*(T-t))/(T-t) ) )

    tol = 1e-8; # Tolerance

    nmax = 1000  # Number of Iterations

    sigmadiff=1

    n=1

    sigma=sigmahat

    bs=BlackScholes() # Initialize an Option object

    while (sigmadiff>=tol and nmax>n):

        if optiontype=='call':

            C=bs.euro\_dividend\_and\_borrowing\_cost(S,K,T,t,sigma,r,q,'call')

        else:

            C=bs.euro\_dividend\_and\_borrowing\_cost(S,K,T,t,sigma,r,q,'put')

        d1=bs.d1\_d2(S,K,T,t,sigma,r,q)[0]

        Cvega=bs.vega(S,K,T,t,r,sigma,q)

        increment= (C-Ctrue)/Cvega

        sigma=sigma-increment

        n=n+1

        sigmadiff=abs(increment)

    return sigma

**3.2** (Code file: 3-2.py)

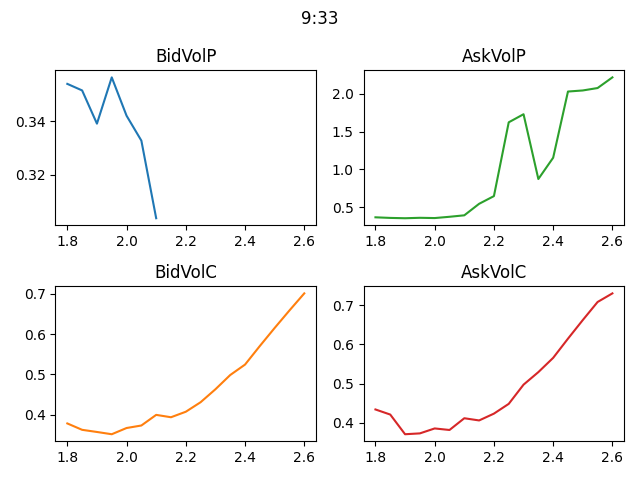
**3.2.1** The bid and ask implied volatilities have been calculated and the results can be found in 31.csv, 32, and 33 csv files as has been instructed.

**3.2.2** Plots for implied volatilities at 3 different timestampsChart, line chart, histogram

Description automatically generated

Chart, line chart, histogram

Description automatically generated



**3.3** Arbitrage Opportunities (Code file: q3-arbitrage.py, Data and Results file: arbitrage\_opportunities.csv)

The data was investigated for the following arbitrage opportunities:

* Put- Call Parity
* Call Option Bound
* Put Option Bound
* Call Option Vertical Spread
* Call Option Non-Negative Butterfly Spread

The results of all arbitrage opportunities can be found in the file arbitrage\_opportunities.csv

Here is a summary:

* A total of 521 arbitrage opportunities have been found in the data
* All the opportunities exist due to violation of Put-Call parity
* 499 of these can be exploited by creating the following portfolio: Short-sell call, Short-sell bond, Buy Put, Buy Underlying
* 22 of these can be exploited by creating the following portfolio: Short-sell put, Short-sell underlying, Buy Call, Buy Risk Free Bond
* The number of arbitrage opportunities including transaction cost of 3.3RMB per option unit: 425
* The total arbitrage profit that can be made after including transaction cost: 4020.73RMB
* The total arbitrage profit for cases without transaction cost: 140.17RMB

Appendix: Plots for Option Price vs Parameters

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart

Description automatically generatedChart

Description automatically generated

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated

Chart, line chart

Description automatically generatedChart

Description automatically generated