COMP6212 Computational Finance Assignment (Part I (2 & 3 (merged) of 3 from MN))

Zheng Cong Koh 28194578

Question 1

a1) Derivative of $\mathcal{N}(x)$

Given that $\mathcal{N}(x)$ is the standard normal cumulative distribution function (CDF), as given in equation 1, the derivative of $\mathcal{N}(x)$ is essentially the standard normal probability distribution function (PDF). This is given in equation 2.

$$\mathcal{N}(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{x^2}{2}} \, dy \quad (1) \qquad \mathcal{N}'(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} = n(x) \quad (2)$$

This is because the CDF of a continuous random variable $F_X(x)$ can be expressed as the integral of its PDF f_X, as shown in equation 3.

$$F_X(x) = \int_{-\infty}^{x} f_X(y) \, dy \tag{3}$$

a2) Show that $S\mathcal{N}'(d_1) = K \exp(-r(T-t))\mathcal{N}'(d_2)$

Starting with the term on the right hand side, we first find $\mathcal{N}'(d_2)$:

$$\begin{aligned}\mathcal{N}'(d_2) &= \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{(d_1 - \sigma\sqrt{(T-t)})^2}{2}\Big) = \frac{1}{\sqrt{2\pi}} \exp\Big(-\frac{d_1^2 - 2d_1\sigma\sqrt{(T-t)} + \sigma^2(T-t)}{2}\Big) \\ &= \frac{1}{\sqrt{2\pi}} \exp\big(-\frac{d_1^2}{2}\big) \exp\big(d_1\sigma\sqrt{(T-t)}\big) \exp\big(-\frac{\sigma^2}{2}(T-t)\big) = \mathcal{N}'(d_1) \exp\big(d_1\sigma\sqrt{(T-t)}\big) \exp\big(-\frac{\sigma^2}{2}(T-t)\big)\end{aligned} \tag{4}$$

Now, we expand the term $\exp\big(d_1\sigma\sqrt{(T-t)}\big)\exp\big(-\frac{\sigma^2}{2}(T-t)\big)$:

$$\begin{aligned}e^{d_1\sigma\sqrt{(T-t)}} e^{-\frac{\sigma^2}{2}(T-t)} &= \exp\big(\log(\frac{S}{K}) + (r + \sigma^2/2)(T-t)\big) \exp\big(-\frac{\sigma^2}{2}(T-t)\big) \\ &= \exp\big(\log(\frac{S}{K}) + r(T-t)\big) = \frac{S}{K} \exp(r(T-t))\end{aligned} \tag{5}$$

Therefore, $\mathcal{N}'(d_2)$ is given by:

$$\mathcal{N}'(d_2) = \mathcal{N}'(d_1) \frac{S}{K} \exp(r(T-t)) \tag{6}$$

Finally, we expand the right hand side term using equation 6:

$$\begin{aligned}K \exp(-r(T-t))\mathcal{N}'(d_2) &= \cancel{K} \cancel{\exp(-r(T-t))} \mathcal{N}'(d_1) \frac{S}{\cancel{K}} \cancel{\exp(r(T-t))} \\ &= S\mathcal{N}'(d_1) = \text{L.H.S} \blacksquare\end{aligned} \tag{7}$$

a3) Derivatives $\partial d_1/\partial S$ and $\partial d_2/\partial S$

By expanding d_1 and d_2, we obtain equation 8 and equation 9 respectively.

$$d_1 = \frac{1}{\sigma\sqrt{(T-t)}}(\log S - \log K) + \frac{1}{\sigma\sqrt{(T-t)}}(r+\sigma^2/2)(T-t) \quad (8)$$

$$d_2 = \frac{1}{\sigma\sqrt{(T-t)}}(\log S - \log K) + \frac{1}{\sigma\sqrt{(T-t)}}(r+\sigma^2/2)(T-t) \;-\sigma\sqrt{(T-t)} \quad (9)$$

As the S term is only present in the first term for both expansion of d_1 and d_2, therefore their derivatives w.r.t S is the same and is given in equation 10:

$$\frac{\partial d_1}{\partial S} = \frac{\partial d_2}{\partial S} = \frac{1}{\sigma\sqrt{(T-t)}}\frac{1}{S} \tag{10}$$

a4) Derivative of call option price with respect to time

As the call option price (c) is made up of many other functions, i.e. ($\mathcal{N}(d_1)$, $\exp()$, $\mathcal{N}(d_2)$), we first find the derivative of each of those functions w.r.t. time. We start off by finding $\partial d_1/\partial t$ and $\partial d_2/\partial t$.

$$\frac{\partial d_1}{\partial t} = \frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}} \tag{11}$$

$$\frac{\partial d_2}{\partial t} = \frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}} + \frac{\sigma}{2}(T-t)^{-\frac{1}{2}} \tag{12}$$

Given the equation for the call option price c, we first write down the equation for $\partial c/\partial t$ as the derivative for the two separate term in the call option price equation.

$$\frac{\partial c}{\partial t} = \frac{\partial(S\mathcal{N}(d_1))}{\partial t} - \frac{\partial(K\exp(-r(T-t))\mathcal{N}(d_2))}{\partial t} \tag{13}$$

From equation 13, we use chain-rule and $\partial d_1/\partial t$ given in equation 11 to expand on the first term:

$$\frac{\partial(S\mathcal{N}(d_1))}{\partial t} = \frac{S\partial(\mathcal{N}(d_1))}{\partial d_1}\frac{\partial d_1}{\partial t} \quad = S\mathcal{N}'(d_1)\left(\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}}\right) \tag{14}$$

Now looking at the second term in equation 13, we apply product rule to expand the term:

$$\begin{aligned}\frac{\partial(K\exp(-r(T-t))\mathcal{N}(d_2))}{\partial t} &= K\exp(-r(T-t))\Big(\frac{\partial\mathcal{N}(d_2)}{\partial t}\Big) + \frac{\partial(K\exp(-r(T-t)))}{\partial t}\mathcal{N}(d_2) \\ &= K\exp(-r(T-t))\Big(\underline{\frac{\partial\mathcal{N}(d_2)}{\partial t}}\Big) + (rKe^{-r(T-t)})\mathcal{N}(d_2)\end{aligned} \tag{15}$$

Next, we use chain-rule and $\partial d_2/\partial t$ given in equation 12 to expand on the term $\frac{\partial\mathcal{N}(d_2)}{\partial t}$ in equation 15:

$$\begin{aligned}\frac{\partial\mathcal{N}(d_2)}{\partial t} &= \frac{\partial(\mathcal{N}(d_2))}{\partial d_2}\frac{\partial d_2}{\partial t} \\ &= \mathcal{N}'(d_2)\left(\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}} + \frac{\sigma}{2\sqrt{(T-t)}}\right)\end{aligned} \tag{16}$$

Finally, using the expansions described above we can then fully expand equation 13 to give the following expressions:

$$\begin{aligned}\frac{\partial c}{\partial t} = S\mathcal{N}'(d_1)\left(\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}}\right) - \Bigg(rKe^{-r(T-t)}\mathcal{N}(d_2) + \\ \cancel{K}\cancel{\exp(-r(T-t))}\Big(\frac{S\mathcal{N}'(d_1)}{\cancel{K}}\cancel{\exp(r(T-t))}\Big)\Big(\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}} + \frac{\sigma}{2\sqrt{(T-t)}}\Big)\Bigg)\end{aligned} \tag{17}$$

$$\frac{\partial c}{\partial t} = S\mathcal{N}'(d_1)\Bigg(\cancel{\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}}}\Bigg) - \Bigg(rKe^{-r(T-t)}\mathcal{N}(d_2) + S\mathcal{N}'(d_1)\Big(\cancel{\frac{\log(\frac{S}{K})}{2\sigma}(T-t)^{-\frac{3}{2}} - \frac{(r+\sigma^2/2)}{2\sigma}(T-t)^{-\frac{1}{2}}} + \frac{\sigma}{2\sqrt{(T-t)}}\Big)\Bigg) \tag{18}$$

$$\frac{\partial c}{\partial t} = -rKe^{-r(T-t)}\mathcal{N}(d_2) - S\mathcal{N}'(d_1)\Big(\frac{\sigma}{2\sqrt{(T-t)}}\Big) \ \blacksquare \tag{19}$$

To show that $\partial c/\partial S = \mathcal{N}(d_1)$, we differentiate call option price equation to find:

$$\begin{aligned}\frac{\partial c}{\partial S} &= \frac{\partial(S\mathcal{N}(d_1))}{\partial S} - \frac{\partial(K\exp(-r(T-t))\mathcal{N}(d_2))}{\partial S}\\ &= S\Big(\frac{\partial\mathcal{N}(d_1)}{\partial S}\Big) + \mathcal{N}(d_1) - K\exp(-r(T-t))\Big(\frac{\partial\mathcal{N}(d_2)}{\partial S}\Big)\\ &= \mathcal{N}(d_1) + \frac{1}{S\sigma\sqrt{(T-t)}}\Big(S\mathcal{N}'(d_1) - K\exp(-r(T-t))\mathcal{N}'(d_2)\Big)\end{aligned} \tag{20}$$

By substituting our findings from equation 4 and 5 into equation 20, we obtain the following:

$$\begin{aligned}\frac{\partial c}{\partial S} &= \mathcal{N}(d_1) + \frac{1}{S\sigma\sqrt{(T-t)}}\Big(S\mathcal{N}'(d_1) - \cancel{K}\cancel{\exp(-r(T-t))}\mathcal{N}'(d_1)\frac{S}{\cancel{K}}\cancel{\exp(r(T-t))}\Big)\\ &= \mathcal{N}(d_1) + \frac{1}{S\sigma\sqrt{(T-t)}}\Big(S\mathcal{N}'(d_1) - S\mathcal{N}'(d_1)\Big) = \mathcal{N}(d_1) \ \blacksquare\end{aligned} \tag{21}$$

a5) Solution to the Black-Scholes differential equation

For the European call option on an underlying stock paying no dividends, the Black-Scholes differential equation is given by:

$$\frac{\partial c}{\partial t} + \frac{1}{2}\sigma^2S^2\frac{\partial^2 c}{\partial S^2} + rS\frac{\partial c}{\partial S} - rc = 0 \tag{22}$$

Therefore by substituting the expressions obtained in the previous sections, the following expression is obtained:

$$\begin{aligned}\text{Eq. (22)} &= -rKe^{-r(T-t)}\mathcal{N}(d_2) - S\mathcal{N}'(d_1)\Big(\frac{\sigma}{2\sqrt{(T-t)}}\Big) + \frac{1}{2}\sigma^2S^2\mathcal{N}'(d_1)\frac{1}{S\sigma\sqrt{(T-t)}} + rS\mathcal{N}(d_1) - rc\\ &= -rKe^{-r(T-t)}\mathcal{N}(d_2) + rS\mathcal{N}(d_1) - rc\\ &= -rKe^{-r(T-t)}\mathcal{N}(d_2) + rS\mathcal{N}(d_1) - r\Big(S\mathcal{N}(d_1) - Ke^{(-r(T-t))}\mathcal{N}(d_2)\Big)\\ &= -rKe^{-r(T-t)}\mathcal{N}(d_2) + rKe^{-r(T-t)}\mathcal{N}(d_2) + rS\mathcal{N}(d_1) - rS\mathcal{N}(d_1) = 0 \ \blacksquare\end{aligned} \tag{23}$$

This shows that the call option price is indeed the solution to the Black-Scholes differential equation because the above expressions and expansions are based on the call option price expression.

b) Option pricing using Black-Scholes model

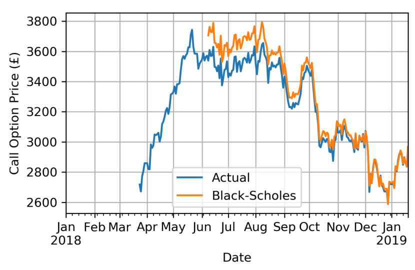
To evaluate how well the option prices in the given data satisfy the Black-Scholes model, we first need to calculate the price of the call options using the Black-Scholes formula:

$$c = S\mathcal{N}(d_1) - K\exp(-r(T-t))\mathcal{N}(d_2) \tag{24}$$

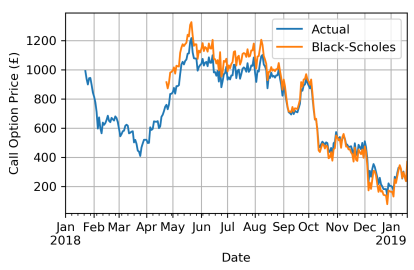
For the price of a corresponding put option it is calculated based on the put-call parity:

$$p = K\exp(-r(T-t)) - S + c \tag{25}$$

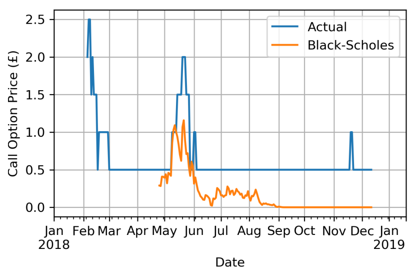
The volatility for the FTSE 100 index can be estimated using the historical data, first by finding the percentage change in the daily index values and then finding the standard deviation of that percentage change. To present this volatility in annualised terms, the daily standard deviation is multiplied by $\sqrt{252}$ (assuming 252 trading days per year). The UK 10-Year Government Bond Yield from the dataset is used as the risk-free interest rate, r for the calculation.



(a) K = 4000 ("in the money")

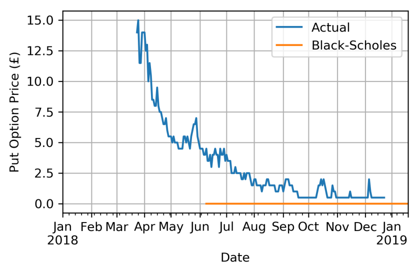


(b) K = 6600

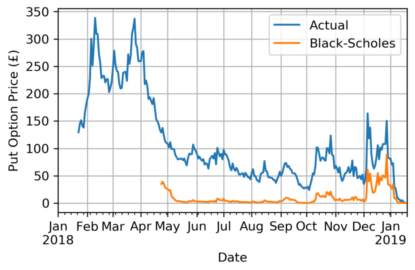


(c) K = 9600 ("out of the money")

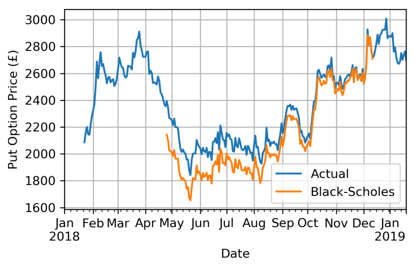
Figure 1: Comparing actual call option prices with that from Black-Scholes formula for different strike prices.



(a) K = 4000 ("out of the money")



(b) K = 6600



(c) K = 9600 ("in the money")

Figure 2: Comparing actual put option prices with that from Black-Scholes formula for different strike prices.

Looking at the results from figures 1 and 2, it is very obvious that the systematic differences is not constant for all strike prices. More specifically, the computed option price is most accurate for "in the money"(ITM) options, as observed in figures 1a and 2c. On the other hand, for "out of the money"(OTM) options, as observed in figures 1c and 2a, there tends to be larger differences between the actual price and the computed option price.

c) Implied volatilities and volatility smile

Given that we have the historical data for the option prices (c/p), as well as the other values used to find the option prices in the previous question, we can compute the implied volatility (σ_{imp}). Unfortunately, there is no closed-form inverse for the Black-Scholes model to compute σ_{imp}, but because it has a closed-form vega ($\nu(\sigma) = \frac{\partial C}{\partial \sigma}$), as shown in equation 26, and it is non-negative, therefore we can use the Newton-Raphson method to obtain σ_{imp}.

$$\frac{\partial C}{\partial \sigma} = S\mathcal{N}'(d_1)\sqrt{(T-t)} \qquad (26) \qquad\qquad \sigma_{n+1} = \sigma_n - \frac{BS(\sigma_n) - P}{\nu(\sigma_n)} \qquad (27)$$

The starting value σ_0 used in the Newton's method is arbitrarily chosen to 1. Then, equation 27 is used to iterate until a solution of sufficient accuracy is reached (where $BS(\sigma_n) - P < 10^{-7}$), where $BS(\sigma_n)$ is the option price computed from the Black-Scholes formula and P is the actual option price from historical data.

Using a random set of 30 working days (14/08/2018 - 25/09/2018), the computed implied volatility from the call option is plotted against the corresponding volatilities estimated from data, as shown in figures 3, 4 and 5. From the figures it can be observed that for options with strike price that are far from the index's price, there is a greater difference between the implied and the estimated volatility.

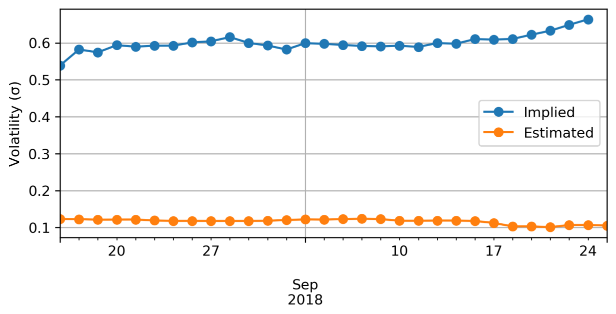


Figure 3: Implied against estimated volatility, $K = 5200$.

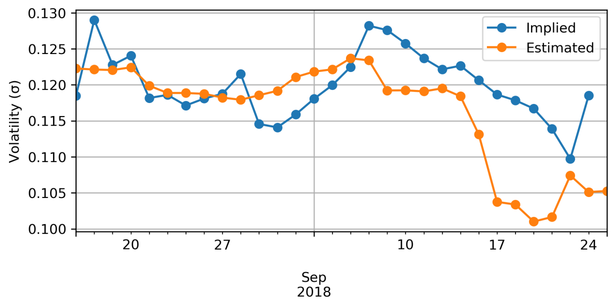


Figure 4: Implied against estimated volatility, $K = 7375$.

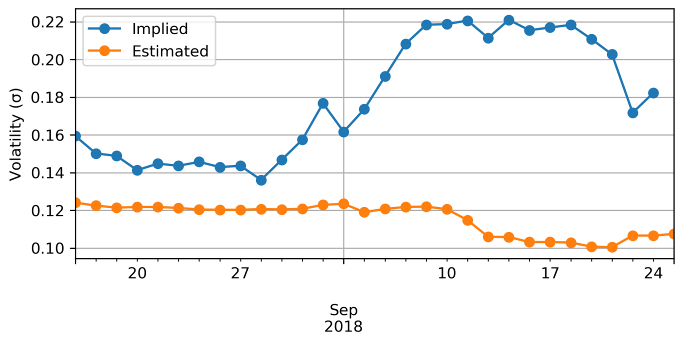


Figure 5: Implied against estimated volatility, $K = 8000$.

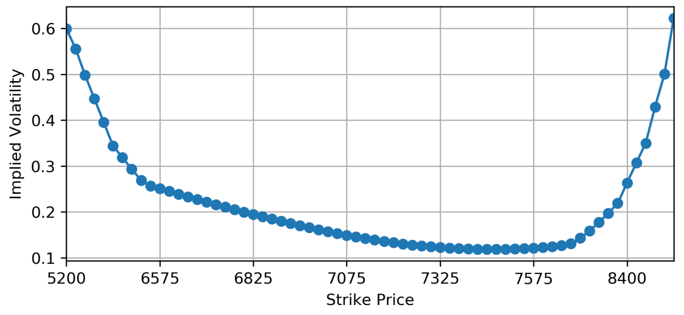


Figure 6: Implied volatility against strike price.

To observe this systematic variation more clearly, the relationship between the strike price and the implied volatility can be plotted, as shown in figure 6. This is done by taking plotting the mean from the computed volatility of the 30 days set for all the options with the different strike prices (from 5200 to 10400). Now we can observe the "Volatility Smile" whereby the volatility increases as the option becomes increasingly in the money or out of the money and is the lowest when the option is at the money. Looking at our data, this is true as it is lowest where $K = 7450$ and the 30 day average for the actual index price, $S_{avg} = 7444.74$.

d) Comparing Black-Scholes and Binomial lattice methods

To compare the pricing for call option with European style exercise for both Black-Scholes and Binomial lattice method, a random set of values for the parameters is chosen, where $K = 7300$, time to maturity $T = 0.389$ (current time - 06/07/18), $r = 0.01267$ and $S = 7617.7$. Therefore, the estimated volatility σ is 0.114 and the Black-Scholes computed price is 433.44.

Different values of the call option price are obtained by using the Binomial lattice method with different values of step time ∂t. The step time is varied from 0.008 to 0.09 and the absolute difference between the values from both methods are plotted as shown in figure 7.

From figure 7, it can be observed that as the step time decreases, the absolute difference decreases, which means that the Binomial lattice method approximates the Black-Scholes method. This is because the binomial model assumes that movements in the price follow a binomial distribution, and for many trials, this binomial distribution approaches the log normal distribution, which is assumed by the Black-Scholes method.

The steps involved in the part of the code listed in question 1d is used to work backwards for all nodes in all time layers in the binomial lattice and calculate the put option payoff. As this is an American style option where the option may be exercised at any time before the expiration date, this means that our option price will need to take into account for the possibility of exercising the option when the option is in the money to obtain an immediate

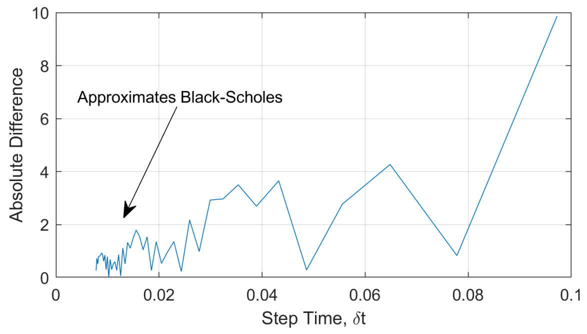


Figure 7: δt against absolute difference.

```
[...]
for tau=1:N
    for i= (tau+1):2:(2*N+1-tau)
        hold = p_u*PVals(i+1) + p_d*PVals
            (i-1);
        PVals(i) = max(hold, SVals(i)-K);
    end
end
[...]
```

Listing 1: Updated snippet for pricing a call option (with American style exercise).

profit.

Therefore, the `hold` variable in line 4 of the code listed in question 1d is used to compute the asset value where we continue and keep the option. This value which is the continuation value is used to compare with the intrinsic value ($K - S$ for put option) to find the option value in each node of the lattice. This is shown in line 5, where the payoff is `max(hold, K-SVals(i))` instead of `max(0, K-Svals(i))` which is used for European exercise options.

To change the code for pricing a call option (with American style exercise), we would need to change line 5 to what is shown in listing 1, where `max(hold, SVals(i)-K)` is used instead. Besides that, there is a part which is above the code snippet that is also required to be changed accordingly for the call option pricing to work correctly. This is because for a call option to have any positive payoff value, the spot price needs to be greater than the strike price, which is why `max(hold, SVals(i)-K)` is used.

Question 2

The paper by Hutchinson *et al.* propose a data-driven method for option pricing, where a RBF model is trained using historical data to price options "accurately". The following describes the work done to reproduce Fig. 4 and Fig. 5 shown in the paper.

To construct the dataset used to trained the RBF model, call option prices are generated using the Black-Scholes formula as shown in equation 24 where the input parameters are taken from the data used in Question 1. A total of 12,999 samples are generated for training the model and evaluating its performance. The generated samples normalised by strike price are plotted versus stock price and time to maturity, as shown in figure 8.

The samples are randomly split in a ratio of 70/30 for training and testing splits respectively. This gives a total of 9,099 samples for the training split and 3,900 samples for the testing split.

The RBF model used is shown in equation 28 with the nonlinear function $\phi_j(\boldsymbol{x})$ is shown in equation 29 where $\boldsymbol{x} = [S/K\ (T-t)]^T$ and $J = 4$. The model thus predicts the call option price C normalised by the strike price K.

$$C/K = \sum_{j=1}^{J} \lambda_j \phi_j(\boldsymbol{x}) + \boldsymbol{w}^T\boldsymbol{x} + w_0 \qquad (28)$$

$$\phi_j(\boldsymbol{x}) = [(\boldsymbol{x} - \boldsymbol{m_j})^T \boldsymbol{\Sigma_j} (\boldsymbol{x} - \boldsymbol{m_j})]^{1/2} \qquad (29)$$

To find the parameters $\boldsymbol{m}_j$ and $\boldsymbol{\Sigma}_j$ used to compute equation 29, the `MATLAB` function `fitgmdist` is used on the training split to fit a Gaussian mixture distribution model with 4 components, as $J = 4$. The option `'SharedCovariance'` is set to `true` so that all values of $\boldsymbol{\Sigma}_j$ are the same, which is similar to what is used in the paper. This is necessary for the fitting to converge with 4 components. Table 1 shows the extracted values for each component that will be used in the RBF model.

Parameters	Values
$\boldsymbol{\Sigma} = \boldsymbol{\Sigma}_j$	$\begin{bmatrix} 0.0122 & -0.0006 \\ -0.0006 & 0.0079 \end{bmatrix}$
$\boldsymbol{m}_1$	$[0.9883 \;\; 0.1529]^T$
$\boldsymbol{m}_2$	$[1.0706 \;\; 0.3876]^T$
$\boldsymbol{m}_3$	$[1.5589 \;\; 0.3105]^T$
$\boldsymbol{m}_4$	$[0.9902 \;\; 0.1556]^T$

Table 1: Computed values for parameters $\boldsymbol{m}_j$ and $\boldsymbol{\Sigma}_j$.

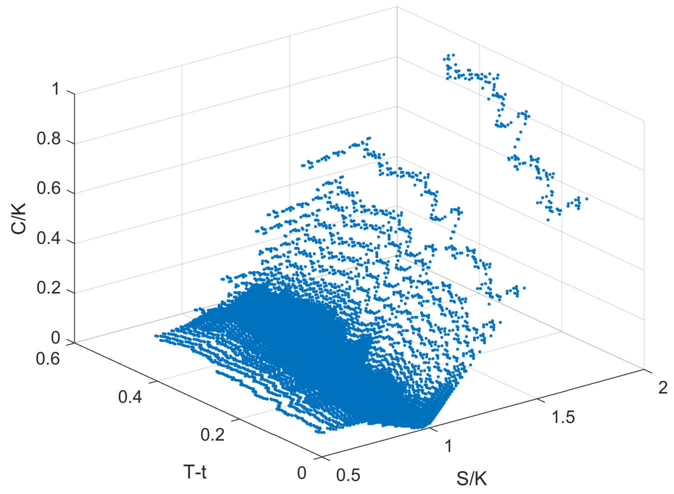


Figure 8: Generated call option prices normalised by strike price against S/K and $T-t$.

Once the parameters $\boldsymbol{m}_j$ and $\boldsymbol{\Sigma}_j$ are computed, the design matrix of inputs can now be constructed using the training split. The setup of the design matrix is shown in equation 30, which is in the form of $y = X\beta$ where X is the design matrix, β is the vector of weights and y is the vector of target outputs for each sample.

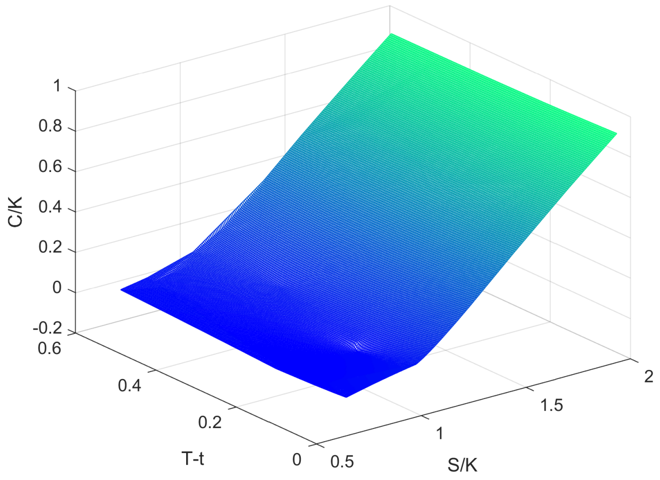
$$\begin{bmatrix} C_1/K_1 \\ C_2/K_2 \\ \vdots \\ C_{N-1}/K_{N-1} \\ C_N/K_N \end{bmatrix} = \begin{bmatrix} \phi_1(\boldsymbol{x}_1) & \phi_2(\boldsymbol{x}_1) & \phi_3(\boldsymbol{x}_1) & \phi_4(\boldsymbol{x}_1) & x_{1(1)} & x_{2(1)} & 1 \\ \phi_1(\boldsymbol{x}_2) & \phi_2(\boldsymbol{x}_2) & \phi_3(\boldsymbol{x}_2) & \phi_4(\boldsymbol{x}_2) & x_{1(2)} & x_{2(2)} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \phi_1(\boldsymbol{x}_{N-1}) & \phi_2(\boldsymbol{x}_{N-1}) & \phi_3(\boldsymbol{x}_{N-1}) & \phi_4(\boldsymbol{x}_{N-1}) & x_{1(N-1)} & x_{2(N-1)} & 1 \\ \phi_1(\boldsymbol{x}_N) & \phi_2(\boldsymbol{x}_N) & \phi_3(\boldsymbol{x}_N) & \phi_4(\boldsymbol{x}_N) & x_{1(N)} & x_{2(N)} & 1 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \\ \lambda_4 \\ w_1 \\ w_2 \\ w_0 \end{bmatrix} \tag{30}$$

Once the system of linear equations is formed, we then then estimate the weights of the model by solving the least squares problem. This can be done by first finding the (Moore-Penrose) pseudo-inverse of the the design matrix X^+ using the `MATLAB` function `pinv` on the design matrix. After finding X^+, we can then use it to find the weight estimates $\hat{\beta}$, such that $\hat{\beta} = X^+ y$, where it is the least squares solution to the problem.

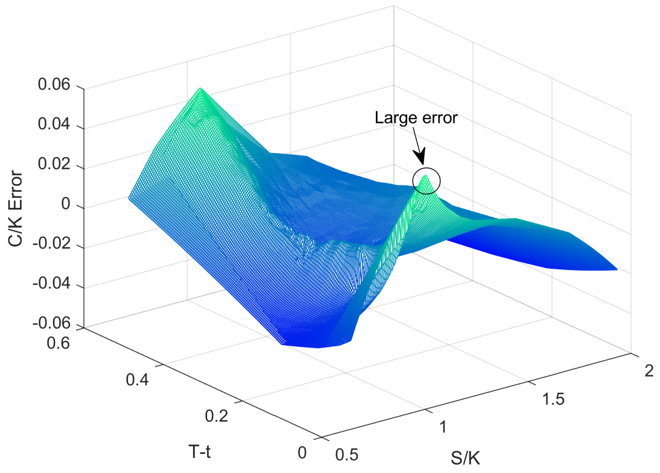
The estimated equation for the trained RBF network is

$$\begin{aligned} \hat{C/K} = 55.8226 &\sqrt{\begin{bmatrix} S/K - 0.9883 \\ T-t-0.1529 \end{bmatrix}^T \begin{bmatrix} 0.0122 & -0.0006 \\ -0.0006 & 0.0079 \end{bmatrix} \begin{bmatrix} S/K - 0.9883 \\ T-t-0.1529 \end{bmatrix}} \\ &+ 2.2476\sqrt{\begin{bmatrix} S/K - 1.0706 \\ T-t-0.3876 \end{bmatrix}^T \begin{bmatrix} 0.0122 & -0.0006 \\ -0.0006 & 0.0079 \end{bmatrix} \begin{bmatrix} S/K - 1.0706 \\ T-t-0.3876 \end{bmatrix}} \\ &- 0.9672\sqrt{\begin{bmatrix} S/K - 1.5589 \\ T-t-0.3105 \end{bmatrix}^T \begin{bmatrix} 0.0122 & -0.0006 \\ -0.0006 & 0.0079 \end{bmatrix} \begin{bmatrix} S/K - 1.5589 \\ T-t-0.3105 \end{bmatrix}} \\ &- 53.1329\sqrt{\begin{bmatrix} S/K - 0.9902 \\ T-t-0.1556 \end{bmatrix}^T \begin{bmatrix} 0.0122 & -0.0006 \\ -0.0006 & 0.0079 \end{bmatrix} \begin{bmatrix} S/K - 0.9902 \\ T-t-0.1556 \end{bmatrix}} \\ &+ 0.4481\, S/K - 0.0333(T-t) - 0.4238. \end{aligned} \tag{31}$$

The performance of the trained model is then evaluated using the testing split. The generated option price are computed by multiplying the design matrix of the testing split with the trained weights of the model. The network call prices and its error are plotted against S/K and $T-t$ as shown in figures 9a and 9b.



(a) Network call price $\hat{C/K}$

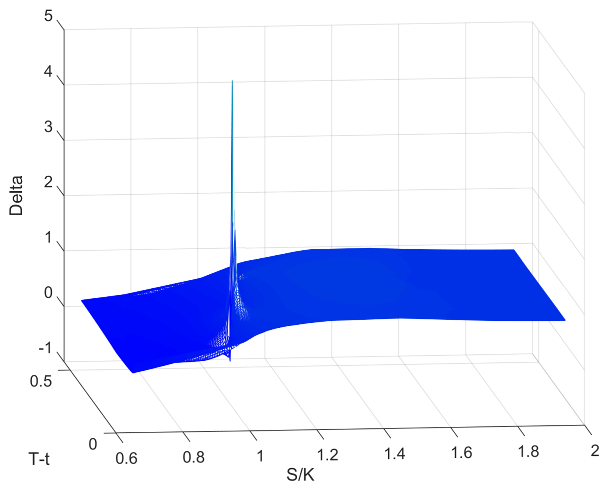


(b) Call price error $\hat{C/K} - C/K$

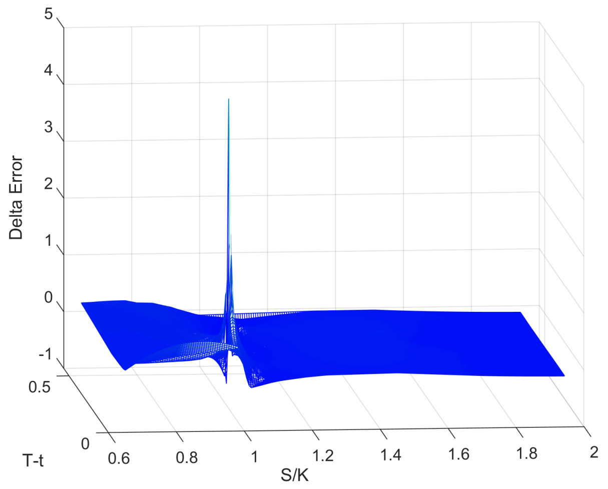
Figure 9: Performance analysis of the network for the call prices.

Looking at figure 9a, it can be seen that the network call price for the testing split approximates the dataset shown in figure 8, which suggests the model is trained correctly. Looking at figure 9b, it can be observed that there are large errors that occur at the point where the option is at the money and at expiration, which is similar to the findings in the paper.

For the performance of the network delta, it can be seen from figure 10a that the overall surface follows the behavior of what is shown in the paper except for the small peak near where $S/K = 1$. For the delta error, it can be seen from figure 10b that delta error only increases as S/K approaches 1 and $T-t$ approaches 0. This means that overall the RBF model is performing relatively well as it has low call price error and low delta error for the majority of the testing split.



(a) Network delta $\frac{\partial \hat{C}}{\partial S}$



(b) Delta error $\frac{\partial \hat{C}}{\partial S} - \frac{\partial C}{\partial S}$

Figure 10: Performance analysis of the network for the delta.