Abstract

In this report, we utilized three methods: Monte Carlo Simulation, Binomial Method and Finite Difference Method (FDM), to price options in the Heng Seng Index market. Constant elasticity of variance (CEV) model, an alternative model to the classical BSM model, and non-uniform grid discretization is used in the implementation of FDM. In addition, we analyzed the results including convergence of prices in the FDM method and implied volatility of the CEV model.

Introduction

A variety of numerical methods have been derived to price financial options since the seminal work of Black and Scholes. The well-known Black-Scholes formula provides a closed-form solution to the Black-Scholes equation, which makes it handy to use in practice. The binomial approach is pedagogical, easy to understand and still retains the economic insight of the Black-Scholes version (Feng & Kwan, 2012). The Monte Carlo method utilizes simulations to estimate an integral representing expectation of terminal payoff. The finite difference method solves the partial differential equations governing option value by approximating partial derivatives with nearby points. In this report, we utilized the above methods to price options in the Heng Seng Index market and analyzed the results to answer some prompted questions.

I. Geometric Brownian Motion Model

Stock price is commonly modelled to follow Geometric Brownian Motion (GBM) which has the following form:

$$\frac{dS_t}{S_t} = \mu dt + \sigma dW_t \tag{1}$$

Where S_t is the stock price at time t, μ is the expected rate of return and σ is the volatility of the stock. A discrete version of GBM can be written as:

$$\frac{\Delta S_t}{S_t} = \mu \Delta t + \sigma\sqrt{\Delta t}\, Z_t \tag{2}$$

Where Z_t are mutually independent standard normal random variables.

To estimate parameters μ and σ of the Heng Seng Index, apply Ito's Lemma on $\ln S_t$ and get below formula for estimation:

$$\Delta lnS_t = \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_t \tag{3}$$

$$E[\Delta lnS_t] = E\left[\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_t\right] = \left(\mu - \frac{1}{2}\sigma^2\right)\Delta t \tag{4}$$

$$Var[\Delta lnS_t] = Var\left[\left(\mu - \frac{1}{2}\sigma^2\right)\Delta t + \sigma\Delta W_t\right] = \sigma^2\Delta t \tag{5}$$

Hence, the parameters can be estimated through:

$$\hat{u} = \frac{E[\Delta lnS_t]}{\Delta t} + \frac{1}{2}\hat{\sigma}^2 = \frac{\bar{R}}{\Delta t} + \frac{1}{2}\hat{\sigma}^2 \tag{6}$$

$$\hat{\sigma} = \frac{\sqrt{\sum_{i=1}^{N} \frac{(R_i - \bar{R})^2}{N}}}{\sqrt{\Delta t}} \tag{7}$$

Where $R_i = lnS_i - lnS_{i-1}$ is the daily log-return, N is sample size.

We obtained the daily historical HSI closing levels between 2021/01/04 and 2022/02/28 from Yahoo Finance. The closing level during the period is shown below:

Figure 1: Heng Seng Index Closing Level From 2021/01/04 to 2022/02/28

Since the closing levels are daily, we assume **250** trading days in a year and set $\Delta t = \frac{1}{250}$:

$$\hat{\mu} = \bar{R} * 250 + \frac{1}{2}\hat{\sigma}^2 \tag{8}$$

$$\hat{\sigma} = \sqrt{Var(R_i)} * \sqrt{250} \tag{9}$$

And we obtain the estimations: $\hat{\mu} = \mathbf{-14.6396}\%$, $\hat{\sigma} = \mathbf{20.5355}\%$.

II. Generate Future Stock Prices Using Monte Carlo Simulation

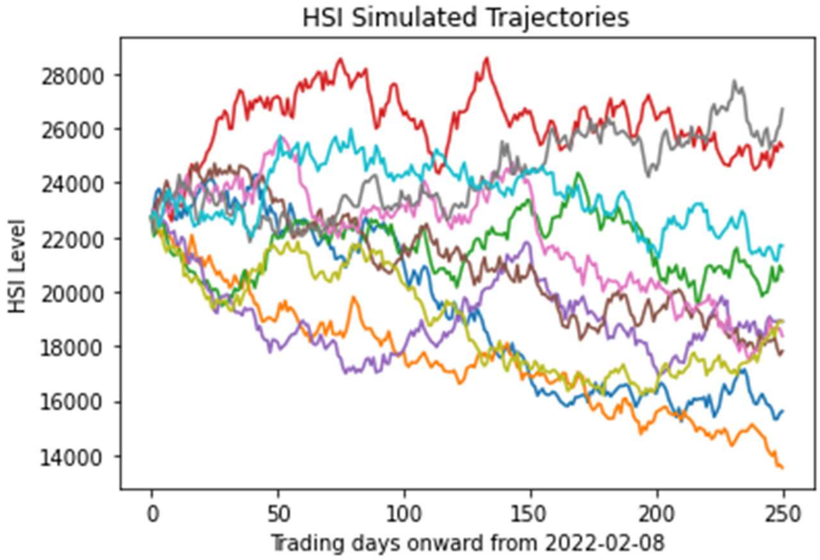
To generate future HSI index level based on the estimate $\hat{\mu}$ & $\hat{\sigma}$, we utilized the formula linking S_{t+1} and S_t:

$$\mathrm{S_{t+1}} = S_t e^{\left(\mu - \frac{1}{2}\sigma^2\right)\Delta\mathrm{t} + \sigma\sqrt{\Delta\mathrm{t}}\, Z_{t+1}} \tag{10}$$

On each new day, we generate an independent normal random variable Z_{t+1}, and calculate the calculate the stock price on that day by equation (10).

Closing index level of Heng Seng on 2022/02/28 is 22713.01953, using this starting value we generated 1000 trajectories for a year in the future (250 days). The first 10 trajectories are displayed below:

Figure 2: Simulated HSI Trajectories Starting 2022/02/28



Compared with the actual HSI path before 2022-02-08 in Figure (1), they look pretty similar in trend and volatility. In addition, we calculate the maximum level, minimum level, average price, average return, average log-return, standard deviation of return and standard deviation of log-return for each trajectory during the one-year period. The statistics are defined as follows[1]:

$$S_{i,max} = Max(S_{i,1}, S_{i,2}, \dots, S_{i,250}) \tag{11}$$

$$S_{i,min} = Min(S_{i,1}, S_{i,2}, \dots, S_{i,250}) \tag{12}$$

$$\overline{S_i} = \sum_{k=1}^{250} \frac{S_{i,k}}{250} \tag{13}$$

$$\widehat{\mu_i} = \sum_{k=1}^{250} \ln\left(\frac{S_{i,k}}{S_{i,k-1}}\right) \frac{1}{250} \tag{14}$$

$$\widetilde{\mu_i} = \sum_{k=1}^{250} \left(\frac{S_{i,k}}{S_{i,k-1}} - 1\right) \frac{1}{250} \tag{15}$$

[1] $S_{i,k}$ is the k-th price of the i-th trajectory

$$\widehat{\sigma_i} = \sqrt{\sum_{k=1}^{250}\left(\ln\left(\frac{S_{i,k}}{S_{i,k-1}}\right) - \widehat{\mu_i}\right)^2 \frac{1}{250}} \tag{16}$$

$$\widetilde{\sigma_i} = \sqrt{\sum_{k=1}^{250}\left(\left(\frac{S_{i,k}}{S_{i,k-1}} - 1\right) - \widetilde{\mu_i}\right)^2 \frac{1}{250}} \tag{17}$$

Distributions of the above statistics are displayed below:

Figure 3: Distribution of Trajectories Average HSI Level

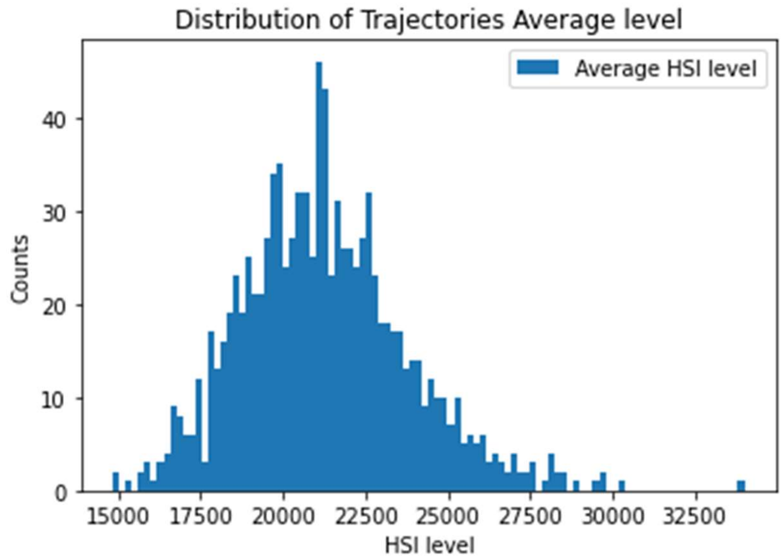


Figure 4: Distribution of Trajectories Maximum HSI Level

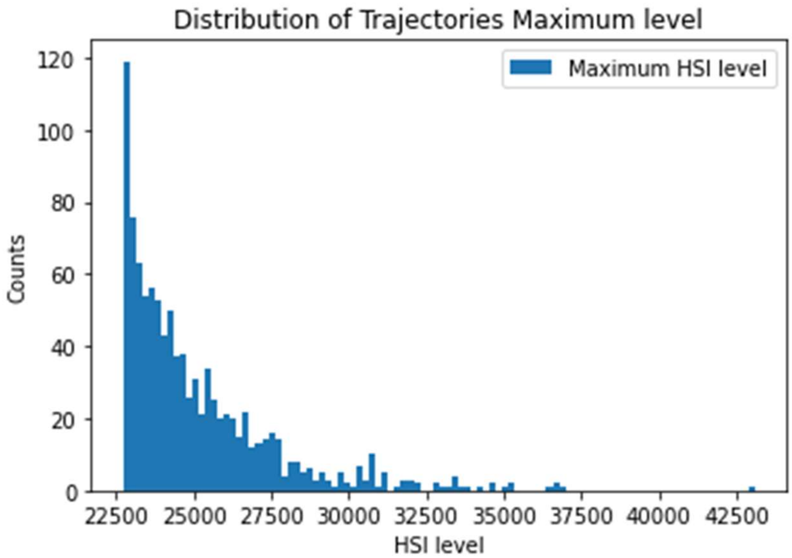


Figure 5: Distribution of Trajectories Minimum Level

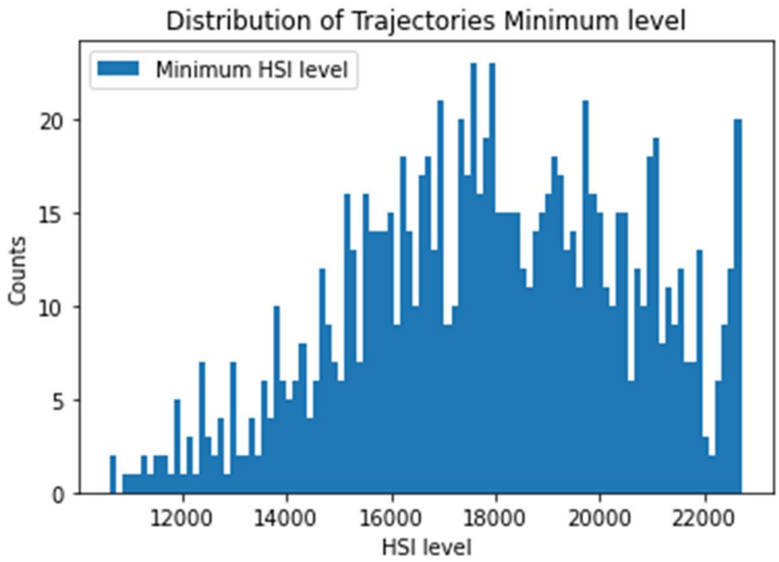


Figure 6: Distribution of Trajectories Average Daily Return

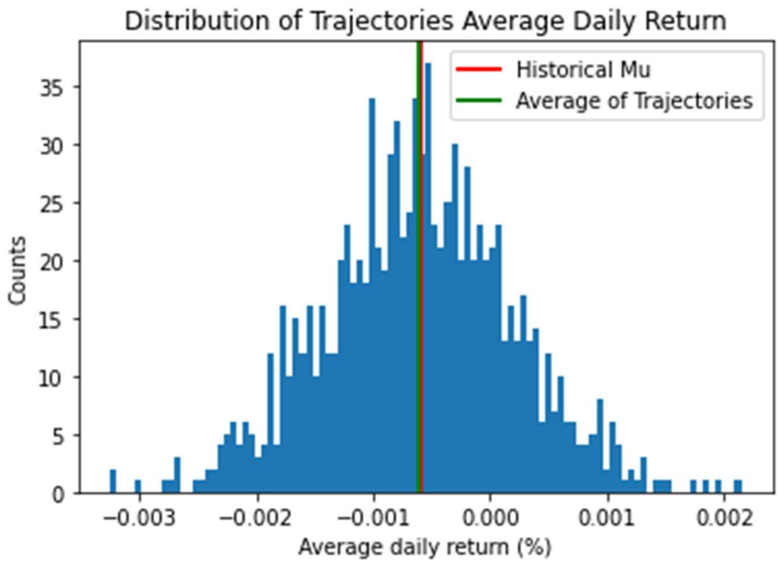
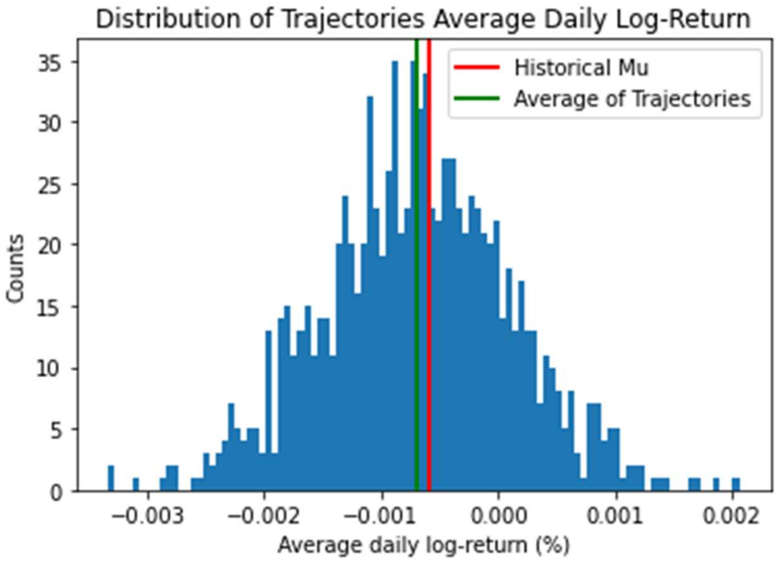


Figure 7: Distribution of Trajectories Average Daily Log-Return



As shown above, the average return of trajectories is very close to the historical $\hat{\mu}$ estimated earlier[2], but the average log-return of trajectories deviates from the estimated $\hat{\mu}$. This is because[3]:

$$E\left[\frac{S_{i,t+1}}{S_{i,t}} - 1\right] = e^{(\hat{\mu})\frac{1}{250}} - 1 = -0.05854\% \approx \frac{\hat{\mu}}{250} = -0.05856\%$$

$$E\left[\ln\left(\frac{S_{i,t+1}}{S_{i,t}}\right)\right] = \frac{\left(\hat{\mu} - \frac{1}{2}\hat{\sigma}^2\right)}{250} \approx -0.06699\% < \frac{\hat{\mu}}{250} = -0.05856\%$$

As a result, the average return of trajectories is closer to the estimated $\hat{\mu}$.

[2] $\hat{\mu}$ is converted to daily here through $\frac{\hat{\mu}}{250}$.

[3] $S_{i,k}$ is the k-th price of the i-th trajectory

Figure 8: Distribution of Trajectories Daily Return Standard Deviation

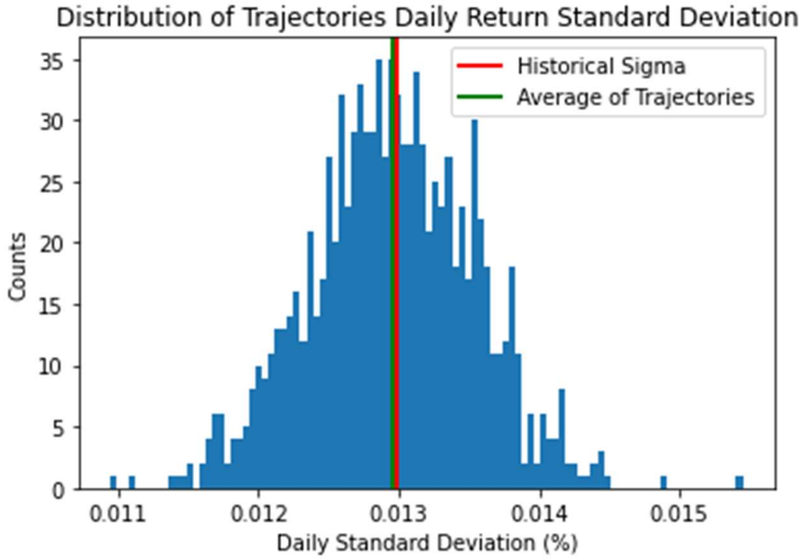
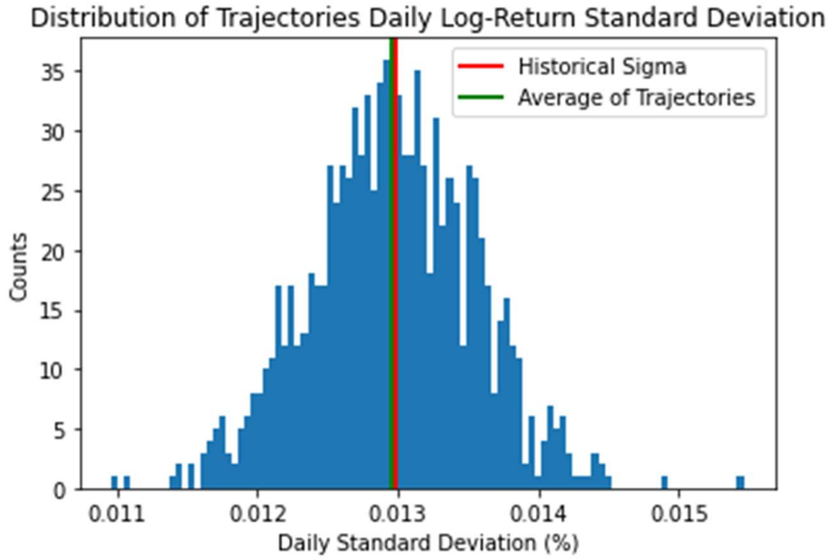


Figure 9: Distribution of Trajectories Daily Log-Return Standard Deviation



Both the standard deviation of return and the standard deviation of log-return are very close to the $\hat{\sigma}$ estimated earlier. This is because:

$$Var\left[\frac{S_{i,t+1}}{S_{i,t}} - 1\right] = \left(e^{\hat{\sigma}^2 \Delta t} - 1\right)\left(e^{2\,\hat{\mu}\Delta t}\right) \approx \ 0.01685\% < \frac{\hat{\sigma}^2}{250} = \ 0.01687\%$$

$$Var\left[\ln\left(\frac{S_{i,t+1}}{S_{i,t}}\right)\right] = \hat{\sigma}^2 \Delta t == \frac{\hat{\sigma}^2}{250}$$

Which shows that both are close to $\frac{\hat{\sigma}^2}{250}$.

III. Valuing Option Price Using the BSM Formula

The BSM Formula (BSF) is a closed-form solution to the BS equation, which provides a handy way of pricing European options, assuming the BS model holds. The BSM formula can be formulated as follows:

$$C(S_t, t) = N(d_1)S_t - N(d_2)Ke^{-r(T-t)} \tag{18}$$

$$P(S_t, t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S_t \tag{19}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right] \tag{20}$$

$$d_2 = d_1 - \sigma\sqrt{T-t} \tag{21}$$

Where S_t is the stock price at time t, K is the strike price, T is the time to maturity

For stocks with a continuous dividend yield, the formula can be adjusted as follow ('Merton's Continuous Leakage Formula'):

$$C(S_t, t) = N(d_1)S_t e^{-\delta(T-t)} - N(d_2)Ke^{-r(T-t)} \tag{22}$$

$$P(S_t, t) = N(-d_2)Ke^{-r(T-t)} - N(-d_1)S_t e^{-\delta(T-t)} \tag{23}$$

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left[\ln\left(\frac{S_t}{K}\right) + \left(r - \delta + \frac{\sigma^2}{2}\right)(T-t)\right] \tag{24}$$

Using the above formula with continuous dividend yield adjustment, we price the below 4 European call options and 4 European put options with HSI index as the underlying asset, assuming that $S_0 = 22713$ (closing level on 2022/02/28):

Table 1: Option Pricing Using BSF

K	T	r	q	type	σ^2_{annual}	BSF price
$1.1 * S_0$	1	0.02	0	Call	20.5355%	1169.6804
$1.2 * S_0$	2	0.02	0	Call	20.5355%	1415.1951
S_0	$\frac{9}{12}$	0.02	0.1	Call	20.5355%	1676.9807
$0.8 * S_0$	$\frac{3}{12}$	0.02	0	Call	20.5355%	4642.7700
$1.1 * S_0$	1	0.02	0	Put	20.5355%	2946.2596

$1.2 * S_0$	2	0.02	0.1	Put	20.5355%	5180.3891
S_0	$\frac{9}{12}$	0.02	0.1	Put	20.5355%	1508.5383
$0.8 * S_0$	$\frac{3}{12}$	0.02	0	Put	20.5355%	9.5408

IV. Valuing Option Price Using Binomial Method

Binomial tree method is another popular method for pricing options. Compared with the BSF, it is pedagogical and easy-to-understand, which makes it a popular teaching material in option pricing courses (Feng & Kwan, 2012). In this section we use the Cox-Ross-Rubinstein Binomial Tree method which defines the parameters of binomial tree as follows[4]:

$$u = e^{\hat{\sigma}\sqrt{\Delta t}} \tag{25}$$

$$d = \frac{1}{u} = e^{-\hat{\sigma}\sqrt{\Delta t}} \tag{26}$$

$$q = \frac{e^{r\Delta t} - d}{u - d} \tag{27}$$

$$\Delta t = \frac{T}{m} \tag{28}$$

Where u is the up jump multiplicative factor, d is the down jump multiplicative factor, q is the risk-neutral probability for an up jump, m is the number of steps.

For stock with continuous dividend yield y, to make risk-neutral expectation a martingale w.r.t. to risk-free rate and y, below adjustment is made:

$$q\, S_t u + (1 + q) S_t d = S_t e^{(r-y)\Delta t}$$

$$q = \frac{e^{(r-y)\Delta t} - d}{u - d} \tag{30}$$

Using above parameters, we price the following two options with different m:

1. Call option: $\{K = S_0 = 22713,\ T = 1,\ r = 0.02,\ q = 0,\ \sigma = 20.54\%\}$
2. Put option: $\{K = 1.2S_0 = 27255.6,\ T = 2,\ r = 0.02,\ q = 0.01,\ \sigma = 20.54\%\}$

Table 2: Call Option Values Using Binomial Method

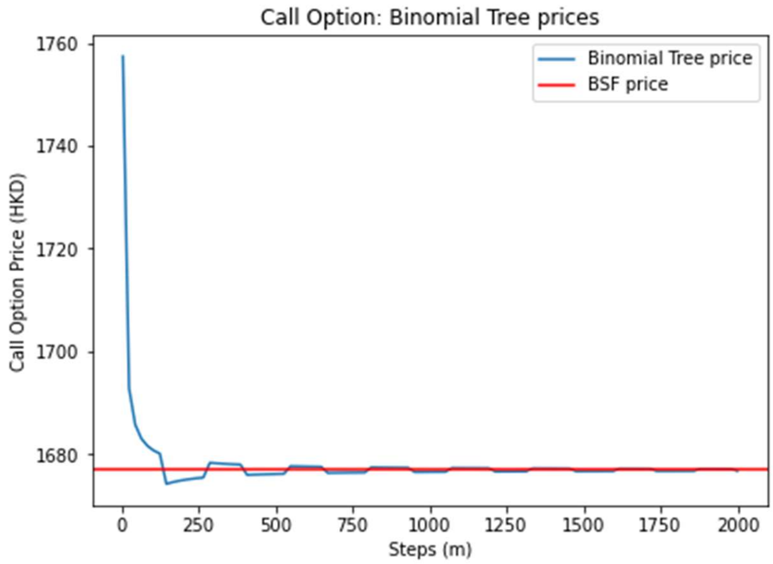
m	Binomial Method Call Price	BSF Price

[4] $\hat{\sigma}$ is the historical annual volatility estimated in section I

4	1581.3936	1676.9807
8	1628.1007	1676.9807
16	1652.3035	1676.9807
32	1664.5879	1676.9807
100	1673.0039	1676.9807
500	1676.1845	1676.9807
1000	1676.5826	1676.9807

As shown below, the binomial method price becomes closer to the BSF price as m increases.

Figure 10: Call Option Value with Increasing Steps m



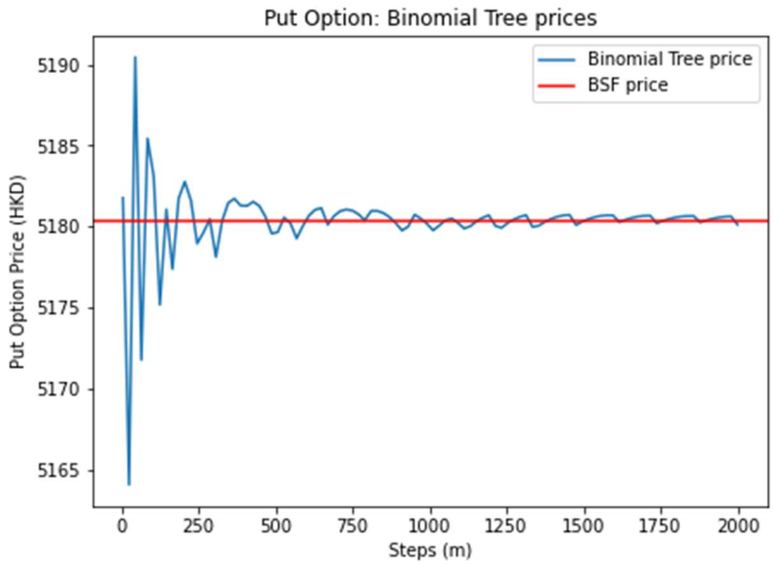
Similarly, for the put option, binomial method put price becomes closer to the BSF price as steps m increases:

Table 3: Put Option Values Using Binomial Method

m	Binomial Method Put Price	BSF Price
4	5300.2030	5180.3891
8	5158.0170	5180.3891
16	5191.0305	5180.3891
32	5185.4080	5180.3891

100	5178.9662	5180.3891
500	5179.1750	5180.3891
1000	5180.0217	5180.3891

Figure 11: Put Option Value With Increasing Steps m



IV-A Comparison of American VS European Option Price

Using the binomial tree method, we can compute American option value by setting the value at each tree node to the maximum of exercise value $S_t - K$ (if call) / $K - S_t$ (if put) and option value at the node. This is to mimic the early exercise action of the option holder since the holder will exercise early if the action gives them higher value. For comparison, we changed the option type of the above two options to American and computed their option value using binomial method with $m = 1000$. The result is shown below:

Table 4: Comparison of American and European Option Value

American Call	European Call	American Put	European Put
1676.5826	1676.5826	5350.9520	5180.0217

As shown above, the call option value remained the same, but the American put option value is higher than that of the European one.

V. Valuing Option Price Using Monte Carlo Simulation

In a risk-neutral world, derivatives are priced by their discounted expected payoff at maturity using a risk-neutral distribution:

$$P = e^{-rT} E_Q[F(S_T, T)] \tag{31}$$

$$P = e^{-rT} \int_{-\infty}^{\infty} F(S_T, T)\, f_Q(S_T) dS_T \tag{32}$$

Where $F(S_t, t)$ is the payoff function at time t, underlying price S_t.

The Monte Carlo Simulation method captures this idea, and numerically approximate the risk-neutral expectation integral through simulations. For call options:

$$E\left[e^{-rT} \sum_{k=1}^{N} \frac{Max\left(S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z_k} - K, 0\right)}{N}\right] \tag{33}$$

$$= \frac{e^{-r}}{N} \sum_{k=1}^{N} E\left[Max\left(S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z_k} - K, 0\right)\right]$$

$$= e^{-rT} E_Q[F(S_T, T)]$$

Where $Z_k \sim N(0,1)$, N is the sample size of simulation.

Hence, the average of simulated discounted terminal payoff is an unbiased estimator of the true discounted expected terminal payoff. In addition, the variance of this unbiased estimator decreases with the sample size N:

$$Var\left(e^{-rT} \sum_{k=1}^{N} \frac{Max\left(S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z_k} - K, 0\right)}{N}\right)$$

$$= \frac{e^{-2rT}}{N^2} \sum_{k=1}^{N} Var\left(Max\left(S_0 e^{\left(r-\frac{1}{2}\sigma^2\right)T+\sigma\sqrt{T}Z_k} - K, 0\right)\right) \; (assuming\; i.i.d. Z_k)$$

$$= \frac{e^{-2rT}}{N} Var\big(Max(S_T - K, 0)\big) \tag{34}$$

Hence, we can estimate the call and put option prices $C = e^{-rT} E_Q[Max(S_T - K, 0)]$ and $P = e^{-rT} E_Q[Max(K - S_T, 0)]$ by generating N random normal variables and take average of the simulated terminal payoffs. The algorithm is summarized as follows:

Set sum = 0;	Set sample size = N;
Set S0 = initial stock price	Set σ = annual volatility of stock;
Set r = risk-free rate	Set T = time to maturity;
Set q = dividend yield	

For i from 1 to N:

(1) Generate $Z_i \sim N(0,1)$

(2) Calculate terminal price $S_T^i = S_0 * e^{\left(r-q-\frac{1}{2}\sigma^2\right)T+\sigma*\sqrt{T}*Z_i}$

(3) Calculate discounted payoff $e^{-r}\ Max\left(S_T^i - K, 0\right)$ (if call option)

(4) $sum = sum + \frac{e^{-rT} Max\left(S_T^i - K, 0\right)}{N}$

Call option value $C = sum$

Using the Monte Carlo method, we price the below European Call option and European Put option with a range of sample size N, and observe that the estimated value becomes closer to real price from the BSF as N increases:

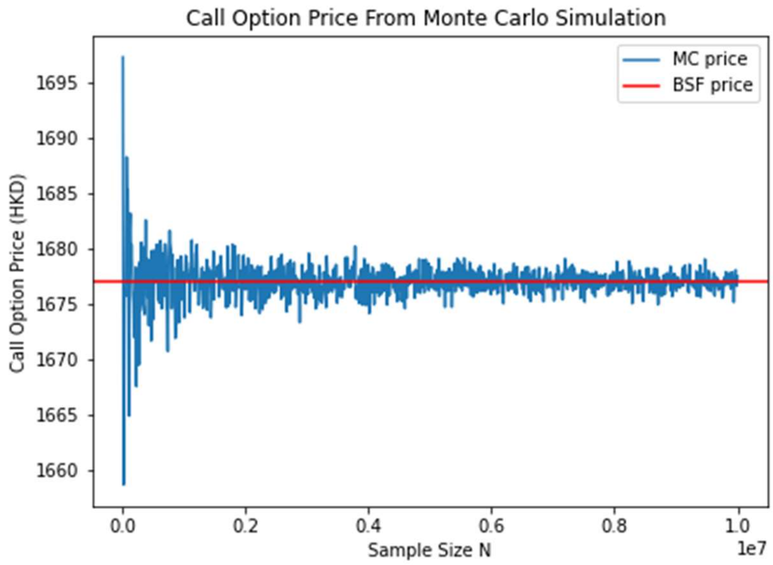
For Call option: $\{K = S_0 = 22713,\ T = 1,\ r = 0.02,\ q = 0,\ \sigma = 20.5355\%\}$

Table 5: Call Option Values Using Monte Carlo Simulation

N	BSF Price	Monte Carlo Price
100	1676.9807	1616.9456
1000	1676.9807	1741.7746
5000	1676.9807	1680.6296
10000	1676.9807	1675.4467
50000	1676.9807	1682.0854
100000	1676.9807	1664.2055
500000	1676.9807	1672.8313
1000000	1676.9807	1676.2829
1500000	1676.9807	1676.9634
5000000	1676.9807	1675.8057

10000000	1676.9807	1678.0017
15000000	1676.9807	1676.1002
30000000	1676.9807	1676.5680

Figure 12: Call Option Value with Increasing Sample Size N



As shown above, the Monte Carlo price becomes closer to the BSF price as sample size increases, and the estimation variance is lower. For the put option, its parameters and MC prices are:

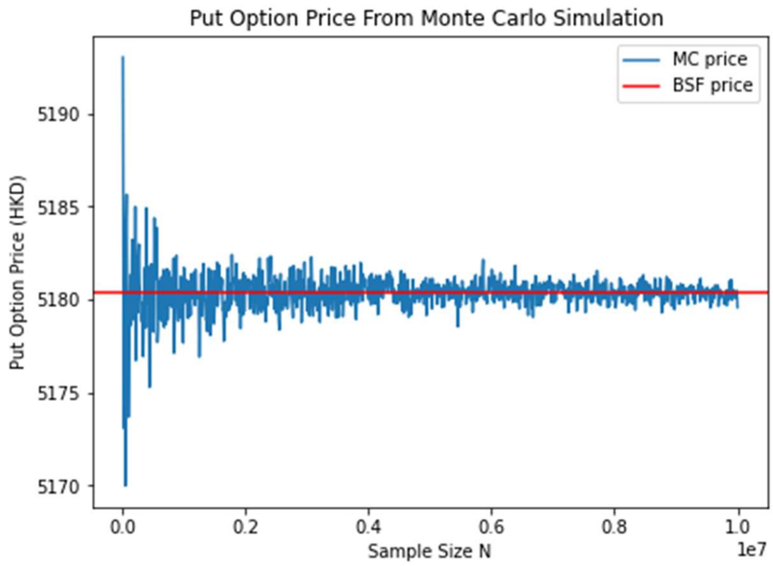
Put option: $\{K = 1.2S_0 = 27255.6,\ T = 2,\ r = 0.02,\ q = 0.01,\ \sigma = 20.5355\%\}$

Table 6: Put Option Values Using Monte Carlo Simulation

N	BSF Price	Monte Carlo Price
100	5180.3891	5165.3394
1000	5180.3891	5177.2980
5000	5180.3891	5170.0955
10000	5180.3891	5181.1168
50000	5180.3891	5182.5824
100000	5180.3891	5182.9195
500000	5180.3891	5179.9361
1000000	5180.3891	5180.1924

1500000	5180.3891	5181.5681
5000000	5180.3891	5180.7342
10000000	5180.3891	5180.5665
15000000	5180.3891	5180.1044
30000000	5180.3891	5179.9926

Figure 12: Put Option Value with Increasing Sample Size N



Same as the call option, the Monte Carlo price becomes closer to the BSF price as sample size N increases, and the estimation variance becomes lower.

VI. Option Pricing with Finite Difference Method

Finite difference methods (FDM) can be used to solve the partial differential equation governing option value by approximating the partial derivatives with nearby points (Rizk, M.M., & Hasan, M.M., 2014). In this section we used Crank-Nicolson Method with a non-uniform grid discretization to price option with a CEV model.

A stock follows constant elasticity of variance (CEV) model if the following is satisfied:

$$\frac{dS}{S} = (\mu - q)dt + \alpha S^{(1-\beta)} dW \tag{35}$$

Where W is standard Brownian motion, $\alpha > 0, \beta \geq 1$ are constants

The backward partial differential equation governing the European option value

$F(S,t)$ is:

$$\frac{\partial F}{\partial t}+\frac{1}{2}\sigma(S,t)^2S^2\frac{\partial^2 F}{\partial S^2}+(r-q)S\frac{\partial F}{\partial S}-rF=0 \tag{36}$$

With boundary conditions:

$$Call\text{:}\ \ F(S\to 0,t)=\ 0 \tag{37}$$

$$Put\text{:}\ \ F(S\to 0,t)=e^{-r(T-t)}(K-S) \tag{38}$$

$$Call\text{:}\ \ F(S\to \infty,t)=e^{-r(T-t)}(S-K) \tag{39}$$

$$Put\text{:}\ \ F(S\to \infty,t)=0 \tag{40}$$

$$Call\text{:}\ \ F(S,T)=Max(S-K,0) \tag{41}$$

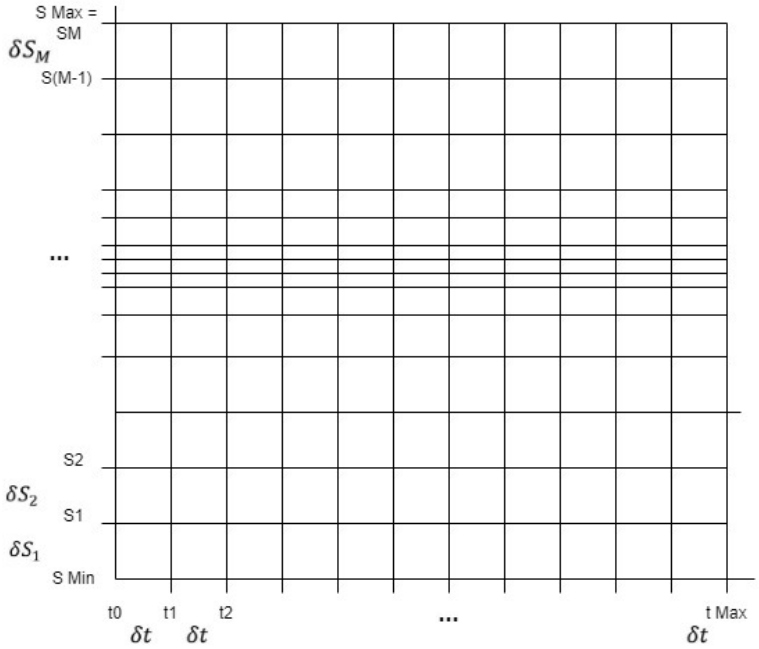
$$Put\text{:}\ \ F(S,T)=Max(K-S,0) \tag{42}$$

As stock price S at time t becomes very low ($S\to 0$), call options are deeply out-of-the-money, it is almost sure that they will not be exercised at maturity, and hence their value at t is close to 0. In contrast, put options are deeply in-the-money, it is almost sure they will be exercised at maturity, and hence their value at time t is the discounted exercised payoff. On the other hand, as stock price S becomes very high at time t ($S\to\infty$), call options are deeply in-the-money, and it is almost sure they will be exercised at maturity. Hence, their value at time t is the discounted exercised payoff. In contrast, put options are deeply out-of-the-money, and hence it is almost sure that they will not be exercised at maturity (thus 0 value as of time t). In addition, the option value at maturity T is always their payoff.

VI-A. Crank-Nicolson Method

To implement finite difference methods, a price-time grid, which separate the X-axis (t) into N intervals (N+1 points) and the Y-axis (S) into M intervals (M+1 points), is introduced. A non-uniform grid discretization is introduced such that the δS between each point in Y-axis is different, but the X-axis δt will be kept uniform. An illustration is shown below.

Figure 13: Illustration of non-uniform grid discretization



Points in the grid is defined as follows[5]:

$$S_j = S_{Min} + \sum_{k=1}^{j} \delta S_j \tag{43}$$

For j=1, 2, 3, …, M

$$S_M = S_{Max} = 2 * S_{spot} \tag{44}$$

$$S_{Min} = 0 \tag{45}$$

$$t_i = i * \delta t \tag{46}$$

$$\delta t = \frac{T}{N} \tag{47}$$

$$t_{Max} = t_N = T \tag{48}$$

A non-uniform grid discretization is chosen such that it has the highest density around $S = S_{spot}$ and gradually become coarser further away from S_{spot}. This is because option value changes more sensitively at around S_{spot}, and hence more a denser discretization in this area can improve computation efficiency (Suggested topics note).

The non-uniform discretization of Y-axis (S space) is generated from a gaussian-related kernel K through the below formula:

[5] S_{spot} is the spot stock price, we do not use S_0 here to avoid confusion with the point in the grid.

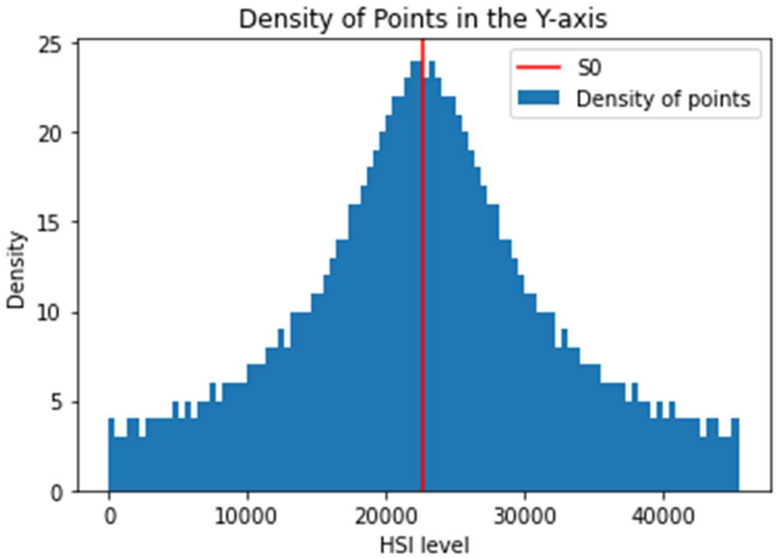
$$\delta S_j = K\left(-2 + 4\frac{j-1}{M-1}\right) \quad (49)$$
$$for\ \ j = 1,2,\dots,M$$

$$K(i) = \frac{\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{i^2}{2}}\right)^{-1}}{\sum_{j=1}^{M}\left(\frac{1}{\sqrt{2\pi}}e^{-\frac{\left(-2+4\frac{j-1}{M-1}\right)^2}{2}}\right)^{-1}} * \left(\frac{S_{max} - S_{min}}{S_{spot}}\right) \quad (50)$$

Where $N(x)$ is the c.d.f of normal distribution

The result of the above discretization of Y-axis is that there are more points (smaller δS_j) around S_{spot}, and fewer points further away from S_{spot}. The distribution of points along Y-axis is shown below:

Figure 13: Density of points along the Y-axis (M=1000)



Finite difference methods solve the partial differential equation (36) by approximating the partial derivatives using nearby points. Crank-Nicolson method is an average of implicit and explicit method (Rizk, M.M., & Hasan, M.M., 2014).

To define the method, we first define the forward, backward and central approximation of partial derivatives as follow[6]:

[6] t_i, S_j, F are defined in equation (46) , (43), (51).

$$f(i,j) = F(t_i, S_j) \tag{51}$$

Forward approximation of $\frac{\partial F}{\partial t} = F_t(t_i, S_j)$:

$$F(t + \delta t, S_j) = F(t_i, S_j) + F_t(t_i, S_j)\delta t + \frac{1}{2} F_{tt}(t_i, S_j)(\delta t)^2 + \cdots$$

$$F_t(t_i, S_j) \approx \frac{F(t + \delta t, S_j) - F(t_i, S_j)}{\delta t} = \frac{f_{i+1,j} - f_{i,j}}{\delta t} \tag{52}$$

Backward approximation of $\frac{\partial F}{\partial t} = F_t(i,j)$:

$$F(t - \delta t, S_j) = F(t_i, S_j) - F_t(t_i, S_j)\delta t + \frac{1}{2} F_{tt}(t_i, S_j)(\delta t)^2$$

$$F_t(t_i, S_j) \approx \frac{F(t_i, S_j) - F(t - \delta t, S_j)}{\delta t} = \frac{f_{i,j} - f_{i-1,j}}{\delta t} \tag{53}$$

Forward approximation of $\frac{\partial F}{\partial S} = F_S(i,j)$:

$$F(t_i, S_j + \delta S_{j+1}) = F(t_i, S_j) + F_S(t_i, S_j)\delta S_{j+1} + \frac{1}{2} F_{SS}(t_i, S_j)(\delta S_{j+1})^2$$

$$F_S(t_i, S_j) \approx \frac{F(t_i, S + \delta S_{j+1}) - F(t_i, S_j)}{\delta S_{j+1}}$$

$$F_S(t_i, S_j) \approx \frac{f_{i,j+1} - f_{i,j}}{\delta S_{j+1}} \tag{54}$$

Backward approximation of $\frac{\partial F}{\partial S} = F_S(i,j)$:

$$F(t_i, S_j - \delta S_j) = F(t_i, S_j) - F_S(t_i, S_j)\delta S_j + \frac{1}{2} F_{SS}(t_i, S_j)(\delta S_j)^2$$

$$F_S(t_i, S_j) \approx \frac{F(t_i, S_j) - F(t_i, S_j - \delta S_j)}{\delta S_j}$$

$$F_S(t_i, S_j) \approx \frac{f_{i,j} - f_{i,j-1}}{\delta S_j} \tag{55}$$

Central approximation of $\frac{\partial F}{\partial S} = F_S(i,j)$:

Subtract $F(t_i, S_j - \delta S_{j-1})$ from $F(t_i, S_j + \delta S_{j+1})$:

$$F\left(t_i, S_j + \delta S_{j+1}\right) - F\left(t_i, S_j - \delta S_j\right)$$

$$= F_S\left(t_i, S_j\right)\left(\delta S_{j+1} + \delta S_j\right)$$

$$+ \frac{1}{2} F_{SS}\left(t_i, S_j\right)\left(\left(\delta S_{j+1}\right)^2 - \left(\delta S_j\right)^2\right)$$

$$F\left(t_i, S_j + \delta S_{j+1}\right) - F\left(t_i, S_j - \delta S_j\right)$$

$$= F_S\left(t_i, S_j\right)\left(\delta S_{j+1} + \delta S_j\right) + O((\delta S)^2)$$

$$F_S\left(t_i, S_j\right) \approx \frac{F\left(t_i, S_j + \delta S_{j+1}\right) - F\left(t_i, S_j - \delta S_j\right)}{\left(\delta S_{j+1} + \delta S_j\right)} = \frac{f_{i,j+1} - f_{i,j-1}}{\left(\delta S_{j+1} + \delta S_j\right)} \tag{56}$$

Central approximation of $\frac{\partial^2 F}{\partial S^2} = F_{SS}(i, j)$:

From $F\left(t_i, S_j - \delta S_{j-1}\right)$, we have:

$$\left(F\left(t_i, S_j - \delta S_j\right) - F\left(t_i, S_j\right)\right) * \delta S_{j+1}$$

$$= -F_S\left(t_i, S_j\right)\delta S_j \delta S_{j+1} + \frac{1}{2} F_{SS}\left(t_i, S_j\right)\left(\delta S_j\right)^2 \delta S_{j+1} \tag{57}$$

From $F\left(t_i, S_j + \delta S_{j+1}\right)$, we have:

$$\left(F\left(t_i, S_j + \delta S_{j+1}\right) - F\left(t_i, S_j\right)\right) * \delta S_j$$

$$= +F_S\left(t_i, S_j\right)\delta S_{j+1} \delta S_j + \frac{1}{2} F_{SS}\left(t_i, S_j\right)\left(\delta S_{j+1}\right)^2 \delta S_j \tag{58}$$

Add (57) to (58), we have:

$$F_{SS}\left(t_i, S_j\right) = 2 \frac{\left(f_{i,j+1} - f_{i,j}\right) * \delta S_j - \left(f_{i,j} - f_{i,j-1}\right) * \delta S_{j+1}}{\left(\delta S_j\right)^2 \delta S_{j+1} + \left(\delta S_{j+1}\right)^2 \delta S_j} \tag{59}$$

The Explicit method uses forward approximation for $\frac{\partial F}{\partial t} = F_t(i, j)$, and central approximation for the remaining two derivatives (Rizk, M.M., & Hasan, M.M., 2014). Substitutes equation (52), (56) and (59) into (36):

$$\frac{f_{i+1,j}-f_{i,j}}{\delta t}+(r-q)S\frac{f_{i,j+1}-f_{i,j-1}}{\delta S_j+\delta S_{j+1}}$$
$$+\sigma_j^2 S_j^2\frac{\left(f_{i,j+1}-f_{i,j}\right)*\delta S_j-\left(f_{i,j}-f_{i,j-1}\right)*\delta S_{j+1}}{\left(\delta S_j\right)^2\delta S_{j+1}+\left(\delta S_{j+1}\right)^2\delta S_j} \tag{60}$$
$$-rf_{i,j}=0$$

The implicit method uses backward approximation for $\frac{\partial F}{\partial t}=F_t(i,j)$, and central approximation for the remaining two derivatives (Rizk, M.M., & Hasan, M.M., 2014). Substitutes equation (53) and (56) (59) into (36), and set index from **(i to i+1)**:

$$\frac{f_{i+1,j}-f_{i,j}}{\delta t}+(r-q)S\frac{f_{i+1,j+1}-f_{i+1,j-1}}{\delta S_j+\delta S_{j+1}} \tag{61}$$
$$+\sigma_j^2 S^2\frac{\left(f_{i+1,j+1}-f_{i+1,j}\right)*\delta S_j-\left(f_{i+1,j}-f_{i+1,j-1}\right)*\delta S_{j+1}}{\left(\delta S_j\right)^2\delta S_{j+1}+\left(\delta S_{j+1}\right)^2\delta S_j}-rf_{i+1,j}$$
$$=0$$

Since Crank-Nicolson method is the average of implicit and explicit method, take average of equation (60) and (61):

$$\frac{f_{i+1,j}-f_{i,j}}{\delta t}+\frac{(r-q)S_j}{2(\delta S_j+\delta S_{j+1})}\left(f_{i+1,j+1}-f_{i+1,j-1}+f_{i,j+1}-f_{i,j-1}\right)$$
$$+\frac{\sigma_j^2 S_j^2}{2\left(\left(\delta S_j\right)^2\delta S_{j+1}+\left(\delta S_{j+1}\right)^2\delta S_j\right)}\Big(\big(f_{i+1,j+1}-f_{i+1,j} \tag{62}$$
$$+f_{i,j+1}-f_{i,j}\big)*\delta S_j-\left(f_{i+1,j}-f_{i+1,j-1}+f_{i,j}-f_{i,j-1}\right)$$
$$*\,\delta S_{j+1}\Big)-\frac{r}{2}\left(f_{i+1,j}+f_{i,j}\right)=0$$

Multiply equation (62) by δt, and set time index from (i to i-1), the equation becomes:

$$
\begin{aligned}
&\left[\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_{j+1}}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)}\right] f_{i, j-1} \\
&\qquad +\left[1-\frac{\delta t\ \sigma_j^2 S_j^2\left(\delta S_j+\delta S_{j+1}\right)}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-\frac{r}{2} \delta t\right] f_{i, j} \\
&\qquad +\left[\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)}+\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_j}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}\right] f_{i, j+1} \\
&\qquad =-\left[\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_{j+1}}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)}\right] f_{i-1, j-1} \\
&\qquad -\left[-\frac{\delta t\ \sigma_j^2 S_j^2\left(\delta S_j+\delta S_{j+1}\right)}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-1-\frac{r}{2} \delta t\right] f_{i-1, j} \\
&\qquad -\left[\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)}+\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_j}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}\right] f_{i-1, j+1}
\end{aligned} \tag{63}
$$

Define the following constant:

$$
a_j=\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_{j+1}}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)} \tag{64}
$$

$$
b_j=\ -\frac{\delta t\ \sigma_j^2 S_j^2\left(\delta S_j+\delta S_{j+1}\right)}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)}-\frac{r}{2} \delta t \tag{65}
$$

$$
c_j=\frac{\delta t\ (r-q) S_j}{2\left(\delta S_j+\delta S_{j+1}\right)}+\frac{\delta t\ \sigma_j^2 S_j^2 \delta S_j}{2\left(\left(\delta S_j\right)^2 \delta S_{j+1}+\left(\delta S_{j+1}\right)^2 \delta S_j\right)} \tag{66}
$$

Equation (63) can be written as:

$$
\begin{aligned}
&-a_j f_{i-1, j-1}+\left[1-b_j\right] f_{i-1, j}-c_j\ f_{i-1, j+1} \\
&\qquad =a_j f_{i, j-1}+\ \left[1+b_j\right] f_{i, j}+c_j\ f_{i, j+1}
\end{aligned} \tag{67}
$$

Since we divided time into N intervals (N+1 points, i=0, 1, …, N), price into M intervals (M+1 points, j=0, 1, …, M), equation (67) is defined for i = 1, 2, …, N and for j = 1, 2, …, M-1. Thus, for every time point i, (M-1) equations are formed with (M-1) unknown variables. The system of linear equations can be solved as:

$$BF_{i-1} = CF_i + d \tag{68}$$

Where:

$$B = \begin{bmatrix} 1-b_1 & -c_1 & 0 & \dots & 0 \\ -a_2 & 1-b_2 & -c_2 & \dots & 0 \\ 0 & -a_3 & 1-b_3 & \dots & \dots \\ \dots & \dots & \dots & \dots & -c_{M-2} \\ 0 & 0 & \dots & -a_{M-1} & 1-b_{M-1} \end{bmatrix} \tag{69}$$

$$C = \begin{bmatrix} 1+b_1 & c_1 & 0 & \dots & 0 \\ a_2 & 1+b_2 & c_2 & \dots & 0 \\ 0 & a_3 & 1+b_3 & \dots & \dots \\ \dots & \dots & \dots & \dots & c_{M-2} \\ 0 & 0 & \dots & a_{M-1} & 1+b_{M-1} \end{bmatrix} \tag{70}$$

$$F_i = \begin{bmatrix} f_{i,1} \\ f_{i,2} \\ \dots \\ \dots \\ f_{i,M-1} \end{bmatrix} \tag{71}$$

$$d = \begin{bmatrix} a_1\left(f_{i,0} + f_{i-1,0}\right) \\ 0 \\ \dots \\ 0 \\ c_{M-1}\left(f_{i,M} + f_{i-1,M}\right) \end{bmatrix} \tag{72}$$

By solving the above equations backward in time (i.e., solve for i=N, then i=N-1, N-2, … and eventually i=0), we can obtain F_0[7] which denotes the option value at $t = 0$ at different S_{spot}.

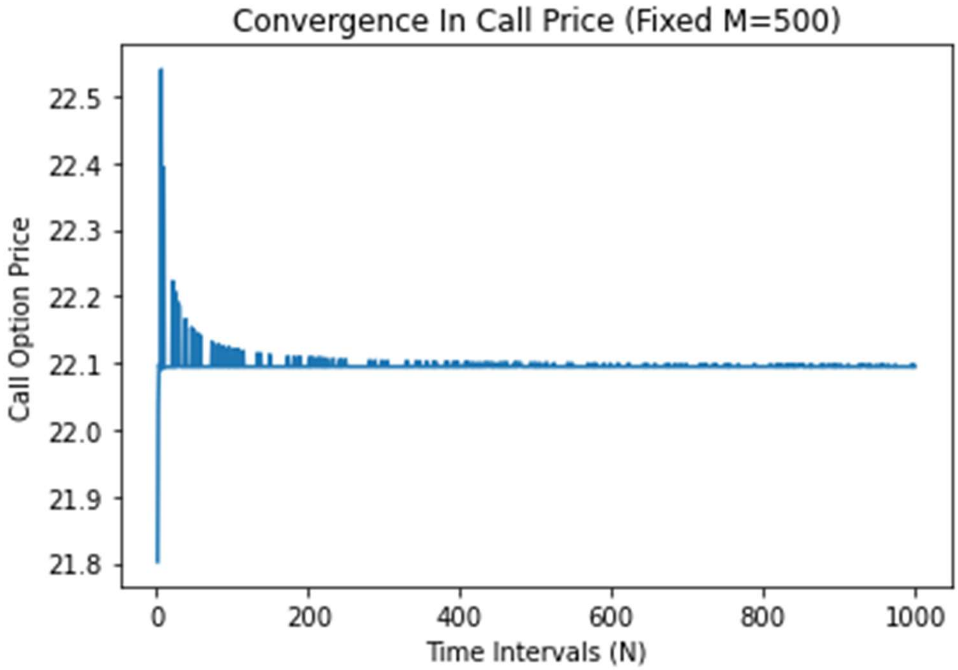
We implemented the Crank-Nicolson method and priced a European call option and a European put option with the following parameters under the CEV model:

$S_0 = 100$	$\alpha = 20$	$\beta = 2$	$r = 2\%$	$q = 1\%$
$T = 1$	$K = 0.8S_0 = 80$			

The prices for Call option with $M = 500$ and varying N converges to around 22.1HKD as shown below.

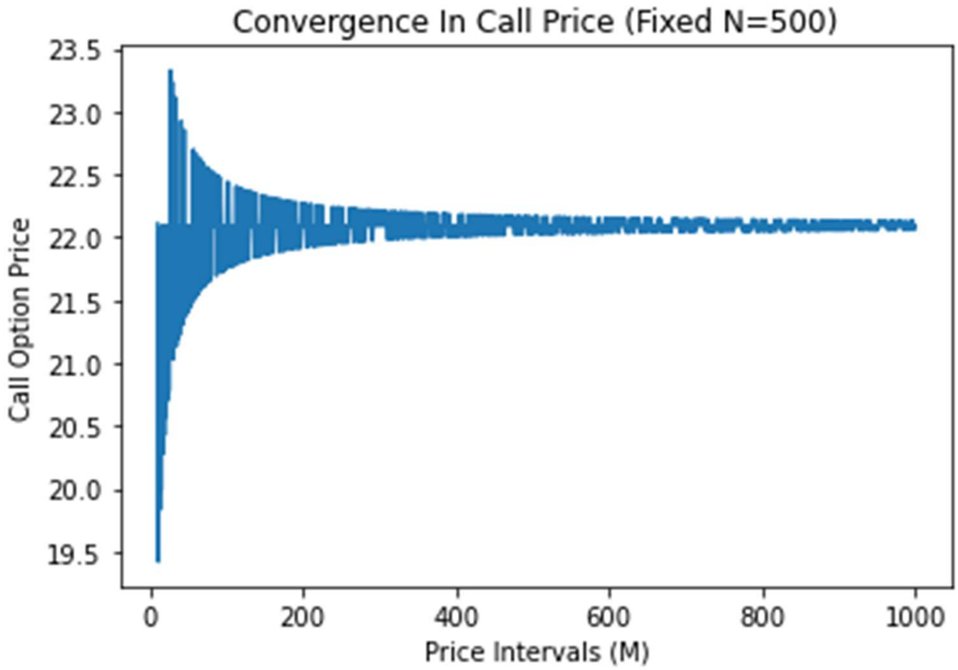
[7] F_0 here refers to the vector in equation (71), not the payoff function

Figure 14: Convergence in Call value with N (Fixed M=500)



By fixing $N = 500$ and varying M, the call option value also converges to around 22.1HKD as shown below:

Figure 15: Convergence in Call value with M (Fixed N=500)



Similarly, the put option value converges to around 1.5 HKD as shown below:

Figure 16: Convergence in Call value with N (Fixed M=500)

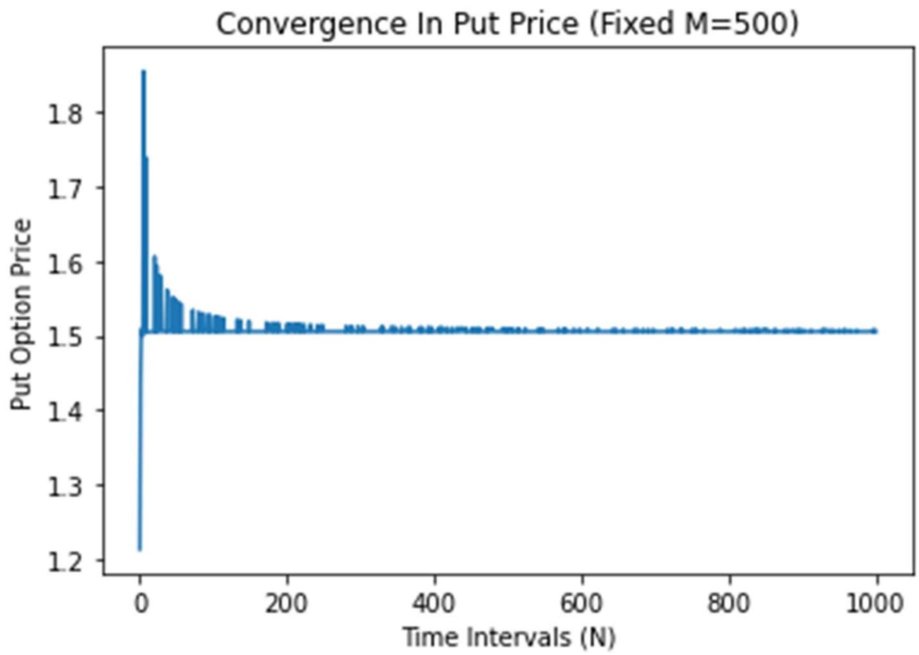
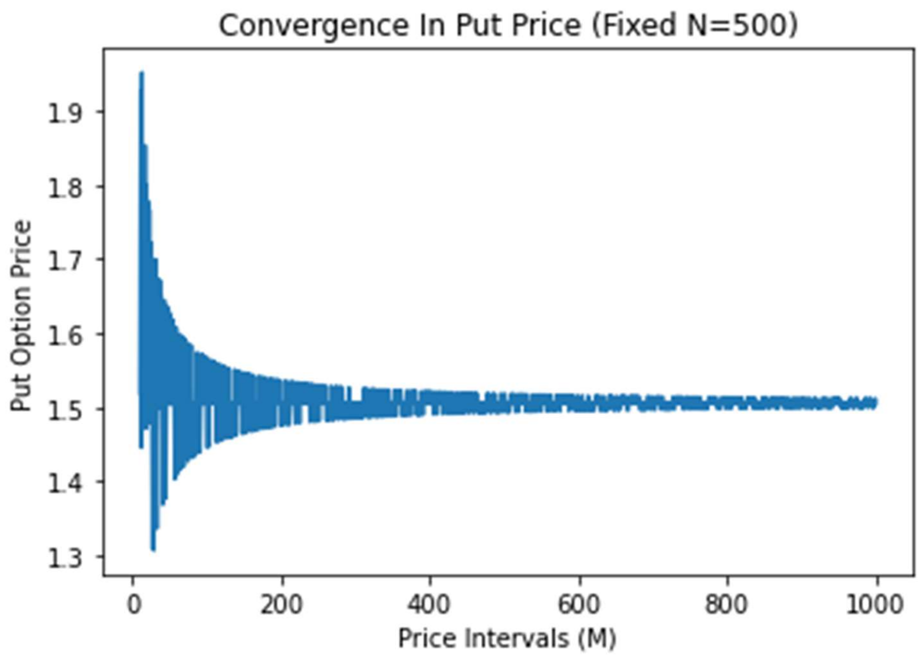
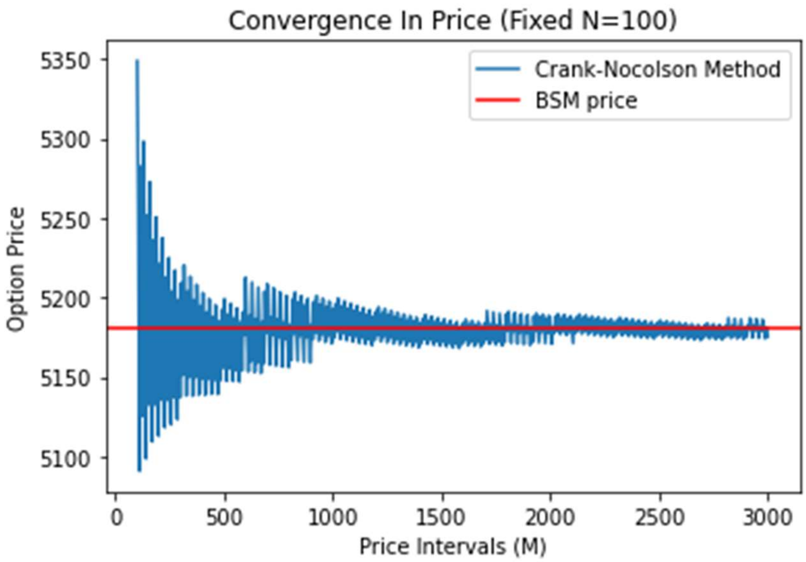


Figure 17: Convergence in Call value with M (Fixed N=500)



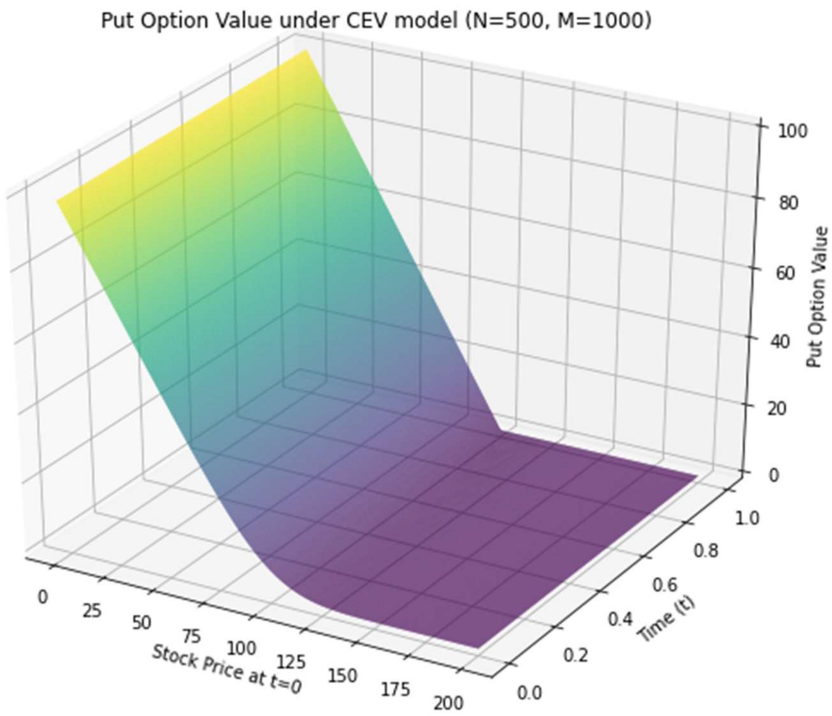
To double check that our program work, we select the call option in section V, changed the parameters of CEV model to $\alpha = \hat{\sigma}, \beta = 1$ so that it becomes the BSM model, and verify that the Crank-Nicolson Method gives a price that converges to the BSF price:

Figure 18: Convergence in Price to BSF Price



In addition, the matrix of an at-the-money put[8] value at different S_{spot} and t are computed and displayed below:

Figure 19: At-the-Money Put Option Value Under CEV Model



As shown in Figure 18, as underlying spot price S_0 increase, the put option value decreases. When the stock price is above or equal to strike $K = 100$, as time t increases option value decreases and eventually becomes 0 at $t = T$. When the stock price is below strike 100, option value converges to become intrinsic value at $t = T$.

Moreover, we tabulate the put option prices with $T = 1$ and strike [0.9, 0.92, 0.94, …, 1.1] S_0[9] below:

Table 7: Put Option Values Using Crank-Nicolson Method

K	Put Value (CEV, alpha=20, beta=2)
90	3.6161
92	4.2226
94	4.8995

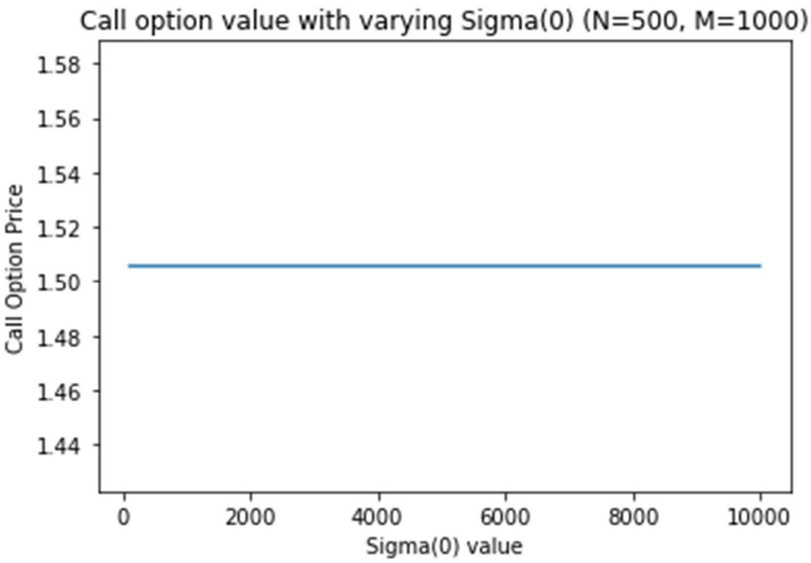
[8] Parameters for the at-the-money options are :{K=100, S_0=100, $r = 2\%$, $q = 1\%$}

[9] Prices are calculated using the Crank-Nicolson Method with $N = 200$, $M = 1000$.

96	5.6496
98	6.4750
100	7.3750
102	8.3574
104	9.4150
106	10.5496
108	11.7596
110	13.0427

Under the CEV model, $\sigma(S_t, t) = \alpha S_t^{1-2} = \frac{20}{S_t}$. At point $S_{min} = 0$, $\sigma(0, t) = \frac{\alpha}{0}$ is undefined. By setting the $\sigma(0, t)$ to arbitrarily large values, we observed the below option value:

Figure 20: Call Option Value with Arbitrarily large $\boldsymbol{\sigma(0, t)}$

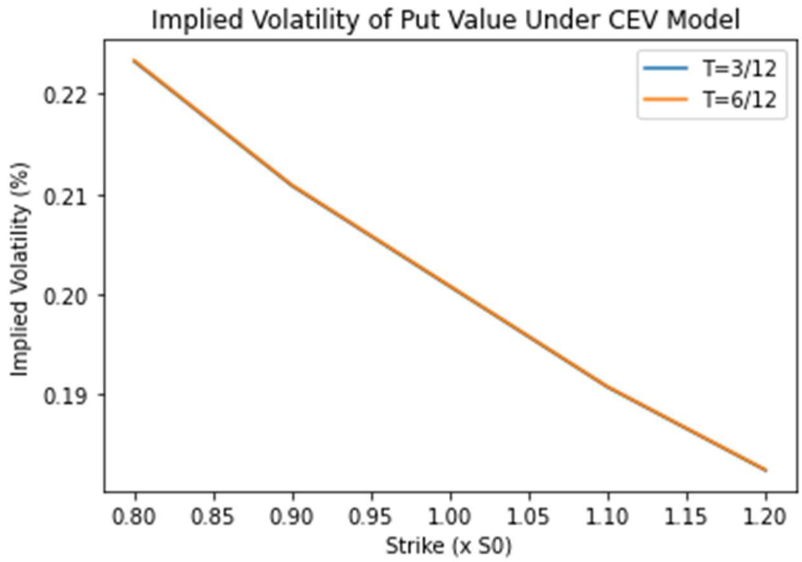


The option value under the Crank-Nicolson Method is invariant to the $\sigma(0, t)$ value, this is because the Crank-Nicolson method uses central approximation for $\frac{\partial^2 F}{\partial S^2}$. It takes $f_{i,j+1}$, $f_{i,j}$ and $f_{i,j-1}$ to approximate $\frac{\partial^2 F}{\partial S^2}(t_i, S_j)$. As a result, we can only approximate $\frac{\partial^2 F}{\partial S^2}(t_i, S_j)$ for $j = 1,2, \dots, M-1$ and define equation (67) for $j = 1,2, \dots, M-1$. And because of the boundary conditions, there are $M-1$ unknown variables, which means we can solve the matrix equation. Hence, we only need a_j, b_j, c_j for $j = 1,2, \dots, M-1$ to solve the equation. This only requires $\sigma(S_j, t)$ to be defined for $j = 1,2, \dots, M-1$. $\sigma(S_{min} = 0, t)$ is not used in the solution.

VI-B. Implied Volatility Under CEV model

We plot the implied volatility of option value under the CEV model with parameters $\alpha = 20$, $\beta = 2$ for puts with strike $[0.8, 0.9, 1.1, 1.2]S_0$ and maturity $T = [\frac{3}{12}, \frac{6}{12}]$. The result is shown below:

Figure 21: Implied Volatility Under CEV Model ($\alpha = 20$, $\beta = 2$)



Under the CEV model, implied volatility of put option is decreasing with strikes. The implied volatility is very close for the two maturities.

VI-C. Forward Parabolic PDE

Alternative to the backward parabolic PDE, a forward parabolic PDE can be written which regards option value $F(K, T, S_0, 0)$ as a function of T (Suggested Topics note). The PDE has the below form:

$$\frac{\partial F}{\partial T} + \frac{1}{2}\sigma(K,T)^2 K^2 \frac{\partial^2 F}{\partial K^2} + (r-q)K\frac{\partial F}{\partial K} - rF = 0 \tag{73}$$

With initial conditions:

$$call: F(K, 0, S_0, 0) = \max(S_0 - K) \tag{74}$$

$$put: F(K, 0, S_0, 0) = \max(K - S_0) \tag{75}$$

$$call: F(K \to 0, T, S_0, 0) = S_0 - K \tag{76}$$

$$put: F(K \to 0, T, S_0, 0) = 0 \tag{77}$$

$$call: F(K \to \infty, T, S_0, 0) = 0 \tag{78}$$

$$put: F(K \to \infty, T, S_0, 0) = K - S_0 \tag{79}$$

As $K \to 0$, $F(K \to 0, T) = 0$ for put options, $F(K \to 0, T) = S_0 - K$ for call options, since it is almost sure that the call options will be exercised and the put option will not.

As $K \to \infty$, $F(K \to \infty, T) = 0$ for call options, $F(K \to \infty, T) = K - S_0$ for put options, since it is almost sure that the put options will be exercised and that the call options will not.

If the initial values of options for many different strikes and maturities are needed, this forward parabolic PDE is computationally more efficient. Since by solving the PDE, we yield option values of different strike K and maturity T, rather than different S_0 and t in the case of backward PDE.

However, If the initial as well as future option values for a single option are needed, the backward PDE is more efficient, since it yields us option values at different t. For future option values, we can look at option value with $t > 0$.

Summary

Binomial tree is useful in pricing American Options due to its stepwise nature which allows us to consider the early exercise decision at each step. Monte Carlo Simulation is a numerical method to approximate option value, but it is prone to estimation error. Finite Difference Method is another numerical method to price option, it allows us to assume a model different than the classical BSM model and approximate option prices under that model. But, it is still an approximation, and the value is affected by parameter configuration ($M\&N$). As the parameters become extremely large, both the Monte Carlo and Finite Difference Method gives close value to the true price.

References

Feng, Yi, and Clarence C. Y. Kwan, 2012, Connecting binomial and Black-Scholes option pricing models: A spreadsheet-based illustration, Spreadsheets in Education 5.

Rizk, M.M., & Hasan, M.M., 2014, Non-uniform Finite Difference Method For European And American Put Option Using Black-Scholes Model. *Journal of Fractional Calculus and Applications* Vol. 5.