

A FINITE-VOLUME APPROACH TO BLACK-SCHOLES FORMULA *

SHENG XU †

Key words. Nonlinear PDEs, Finite-Volumn Method, Black-Scholes Formula, European Option, Option Pricing

Abstract. Option Pricing problems during investment projects are mostly solved by the simulation-based methods, the lattice method and finite-difference method(FDM). There are also papers implies the successful application of the finite-element method(FEM), usually combined with techniques in FDM.

In this paper, we investigated the application of the finite-volume method to the Black-Scholes Model and provide a detail scheme for pratical implementations. A modified finite-volume methhod is introduced as a numerical simulation of European Option Pricing. Detailed derivation process of the method is developed and numerical results ares provided. The numerical results are compared with Matlab built-in solver and show a good performance of the method.

1. Introduction. An option is a financial derivative that represents a contract sold by one party (the option writer) to another party (the option holder). The contract offers the buyer the right, but not the obligation, to buy (call) or sell (put) a security or other financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date)[1].

Black-Scholes(BS) Model[2] is one of the most commonly employed model based on partial difference equations, which provides an exact closed form solutions for financial derivatives. The equation is stated as:

$$\frac{\partial V}{\partial t} + rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0; \quad (1.1)$$

where S is a real asset value, $0 \leq S \leq \infty$, V is the (real) option price, r is the risk-free rate, t is the time since the option was issued, $0 \leq t \leq T$, and σ is the real asset volatility. This equation is a backward moving equation, i.e. it is solved from the future to the present time.

For an European call option, the time condition becomes a final condition because its value is known at the maturity date $t = T$ and it is defined as its intrinsic value by:

$$V(S, T) = \max(S - K, 0), \forall S; \quad (1.2)$$

For an European put option, the time condition becomes a final condition because its value is known at the maturity date $t = T$ and it is defined as its intrinsic value by:

$$V(S, T) = \max(K - S, 0), \forall S; \quad (1.3)$$

This model has been mostly solved by the latice method[3, 4], the finite-difference method(FDM)[4, 5] and simulation methods[6, 7]. There are also papers applying finite-element method(FEM) to BS Model[8, 9], suggesting good convergence. It is now known that the BS formula can be transformed into heat equation[8]:

$$\frac{\partial u}{\partial t} - \frac{\partial^2 u}{\partial \chi^2} = 0; \quad (1.4)$$

*Submitted for Math 250 Final Project, April 25, 2015

†Department of Mathematics, 503 Boston Ave, Tufts University, Medford, MA 02155. *email:* sheng.xu@tufts.edu

This offers a solid theoretical foundation to solve BS formula with FDM, FEM as well as finite-volume method(FVM)[10]. The finite volume method (FVM) is a method for representing and evaluating partial differential equations. Similar to the finite difference method or finite element method, values are calculated at discrete places on a meshed geometry. "Finite volume" refers to the small volume surrounding each node point on a mesh[10, 11]. In the finite volume method, volume integrals in a partial differential equation that contain a divergence term are converted to surface integrals, using the divergence theorem. One advantage of the finite volume method is that it is easily formulated to allow for non-uniform meshes[12]. This is vital for pricing options with Black-Scholes Model. While the range of the asset price is quite wide, the values of options only change dramatically near the "current value" of asset value[1].

The finite-volume method is rarely studied with BS model. However, it is important to study other methods to numerically deal with this equation more quickly. Still, there are extreme conditions, mainly "smiles" [1], when the exact solution lose its accuracy and numerical results is extremely useful. In 2004, Wang[13] introduced a method based on a fitted finite volume spatial discretization and an implicit time stepping technique. In this paper, we investigated the application of a finite-volume method to the Black-Scholes Model and provide a detail scheme for practical implementations. A modified finite-volume method is introduced as a numerical simulation of European Option Pricing. Detailed derivation process of the method is developed and numerical results is provided. The numerical results are compared with Matlab built-in solver and show a good performance of the method.

2. Discretization of BS Model with FVM. In section 1, we introduced the Black-Scholes Model and describe the parameters. It is known that we know a strike at time $t = T$ and want to calculate the price at time $0 \leq t < T$. To deal with the problem, we define $\tau = T - t$ and transform the formula into:

$$\frac{\partial V}{\partial \tau} = rS \frac{\partial V}{\partial S} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV; \quad (2.1)$$

where S is a real asset value, $0 \leq S \leq \infty$, V is the (real) option price, r is the risk-free rate, τ is the time before the option is exercised, $0 \leq \tau \leq T$, and σ is the real asset volatility.

For the first step, we show how to transform the Black-Scholes Formula into diffusion equation[8]. Let:

$$\begin{aligned} \chi &= \ln \frac{S}{K}, \\ \gamma &= \frac{\sigma^2}{2}(T - t), \\ v(\chi, \gamma) &= \frac{1}{K} V(S, t) e^{\frac{1}{2}(k_1 - 1)\chi + \frac{1}{4}(K_1 + 1)^2 \gamma}; \end{aligned} \quad (2.2)$$

where $k_1 = \frac{2r}{\sigma^2}$. Thus, we will have: $\frac{\partial v}{\partial \gamma} - \frac{\partial^2 v}{\partial \chi^2} = 0$.

We then integrate the equation (2.1) on a divided domain $A_i \times T_n = [S_{i-\frac{1}{2}}, S_{i+\frac{1}{2}}] \times [\tau_n, \tau_{n+1}]$, where $S_{i-\frac{1}{2}} = \frac{1}{2}(S_{i-1} + S_i)$; $S_{i+\frac{1}{2}} = \frac{1}{2}(S_i + S_{i+1})$.

Let $\|T_n\| = \Delta\tau = \tau_{n+1} - \tau_n$; $\|A_i\| = S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} = \frac{1}{2}(S_{i+1} - S_{i-1})$; $V_n^i = V(S_i, \tau_n)$. This will give

us the formula:

$$\int_{A_i} V_i^{n+1} - V_i^n = \Delta\tau \left(\int_{A_i} rS \frac{\partial V}{\partial S} + \int_{A_i} \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - \|A_i\| rV \right); \quad (2.3)$$

The left-hand side gives:

$$\int_{A_i} V_i^{n+1} - V_i^n = \|A_i\| (V_i^{n+1} - V_i^n); \quad (2.4)$$

For the right-hand side, using Approximation we have:

$$\begin{aligned} \int_{A_i} rS \frac{\partial V}{\partial S} &\approx rS_i [V_{i+\frac{1}{2}}^{n+1} - V_{i-\frac{1}{2}}^{n+1}] \\ &\approx \frac{1}{2} rS_i [V_{i+1}^{n+1} - V_{i-1}^{n+1}]; \end{aligned} \quad (2.5)$$

And:

$$\begin{aligned} \int_{A_i} \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} &\approx \frac{1}{2} \sigma^2 S_i^2 [(V_S)_{i+\frac{1}{2}} - (V_S)_{i-\frac{1}{2}}] \\ &\approx \frac{1}{2} \sigma^2 S_i^2 \left(\frac{V_{i+1}^{n+1} - V_i^{n+1}}{S_{i+1} - S_i} - \frac{V_i^{n+1} - V_{i-1}^{n+1}}{S_i - S_{i-1}} \right); \end{aligned} \quad (2.6)$$

Now let's assemble (2.4) (2.5) (2.6) into (2.3) and using the implicit scheme similar to FDM:

$$\begin{aligned} \|A_i\| (V_i^{n+1} - V_i^n) &\approx \Delta\tau \left(\frac{1}{2} rS_i [V_{i+1}^{n+1} - V_{i-1}^{n+1}] \right. \\ &\quad \left. + \frac{1}{2} \sigma^2 S_i^2 \left[\frac{V_{i+1}^{n+1} - V_i^{n+1}}{S_{i+1} - S_i} - \frac{V_i^{n+1} - V_{i-1}^{n+1}}{S_i - S_{i-1}} \right] \right. \\ &\quad \left. - \|A_i\| rV_i^{n+1} \right); \end{aligned} \quad (2.7)$$

This gives an implicit formula of V^n and V^{n+1} .

$$V_i^n \approx \alpha_{i-1} V_{i-1}^{n+1} + \alpha_i V_i^{n+1} + \alpha_{i+1} V_{i+1}^{n+1}; \quad (2.8)$$

where

$$\begin{aligned} \alpha_{i-1} &= -\frac{\sigma^2 S_i^2 \Delta\tau}{(S_{i+1} - S_{i-1})(S_i - S_{i-1})} + \frac{S_i r \Delta\tau}{S_{i+1} - S_{i-1}}, \\ \alpha_i &= 1 - \frac{\sigma^2 S_i^2 \Delta\tau}{(S_i - S_{i-1})(S_{i+1} - S_i)} + r \Delta\tau, \\ \alpha_{i+1} &= -\frac{\sigma^2 S_i^2 \Delta\tau}{(S_{i+1} - S_{i-1})(S_{i+1} - S_i)} - \frac{S_i r \Delta\tau}{S_{i+1} - S_{i-1}}; \end{aligned} \quad (2.9)$$

3. Numerical Results. In this part, we discuss our result of the implementation of the algorithm. Here, We set our parameters as:

$$\sigma = 0.4, r = 0.05, K = 100, T = 1, n = 4;$$

According to the behavior of the price of options, we input an S like this:

$$S = K/100*[0 : 20 : 60 \quad 61 : 4 : 89 \quad 90 : 0.8 : 110 \\ 111 : 4 : 139 \quad 140 : 20 : 200 \quad 300 \quad 500 \quad 1000 \quad 5000];$$

$2^{(n-1)} - 1$ points are added into each two points in S, dividing the intervals in S into 2^n subintervals. We also descritize the time into $365 * 2^{n-1} * T$ sub intervals. Note according the nature that the change of value for $S \notin [80, 100]$ is slow, so we use bigger intervals outside $[80, 100]$. The figures of $n=4$ are on Page 8 11. We briefly introduce the figures:

Figure 3.1-3.4 gives the result of a European call option; Figure 3.5-3.8 gives the result of a European put option. Here we set $n=4$. The error of European call option is 0.0022 on $S \in [80, 120]$. The error of European put option is 0.0021 on $S \in [80, 120]$.

We now numerically study the relation of error and size of meshes $\sigma = 0.4; r = 0.05; K = 100; T = 1$; type="call":

$\Delta\tau$	ΔS for $S \in [80, 120]$	FVM Run-Time	Matlab Fcn Run-Time	Max Error on $S \in [80, 120]$
$\frac{1}{365}$	0.4	0.021	0.220	0.0276
$\frac{1}{365}$	0.2	0.101	0.311	0.0122
$\frac{1}{365}$	0.1	0.643	0.703	0.0073
$\frac{1}{730}$	0.4	0.038	0.202	0.0249
$\frac{1}{730}$	0.2	0.180	0.362	0.0086
$\frac{1}{730}$	0.1	1.000	0.688	0.0042
$\frac{1}{1460}$	0.4	0.077	0.222	0.0241
$\frac{1}{1460}$	0.2	0.366	0.373	0.0073
$\frac{1}{1460}$	0.1	1.744	0.658	0.0028
$\frac{1}{2920}$	0.4	0.200	0.292	0.0237
$\frac{1}{2920}$	0.2	0.679	0.380	0.0066
$\frac{1}{2920}$	0.1	4.055	0.696	0.0022

Table 3.1: Running Time and Error according to ΔS and $\Delta\tau$

We see that the running time of FVM is increasing by $\frac{1}{\Delta\tau}$ and $\frac{1}{\Delta S^2}$ respectively. We discuss the truncation error in the next section.

4. Error Estimate and Stability. In previous sections, we have:

$$\begin{aligned} \|A_i\|(V_i^{n+1} - V_i^n) &\approx \Delta\tau \left(\frac{1}{2} r S_i [V_{i+1}^{n+1} - V_{i-1}^{n+1}] \right. \\ &\quad + \frac{1}{2} \sigma^2 S_i^2 \left[\frac{V_{i+1}^{n+1} - V_i^{n+1}}{S_{i+1} - S_i} - \frac{V_i^{n+1} - V_{i-1}^{n+1}}{S_i - S_{i-1}} \right] \\ &\quad \left. - \|A_i\| r V_i^{n+1} \right); \end{aligned} \quad (2.7)$$

We modify the equation (2.7) a little bit to match the form of equation (2.1) and write ε as truncation error:

$$\begin{aligned} \varepsilon &= \frac{V_i^{n+1} - V_i^n}{\Delta\tau} - \frac{r S_i}{2 \|A_i\|} [V_{i+1}^{n+1} - V_{i-1}^{n+1}] \\ &\quad - \frac{\sigma^2 S_i^2}{2 \|A_i\|} \left[\frac{V_{i+1}^{n+1} - V_i^{n+1}}{S_{i+1} - S_i} - \frac{V_i^{n+1} - V_{i-1}^{n+1}}{S_i - S_{i-1}} \right] + r V_i^{n+1}; \end{aligned} \quad (4.1)$$

We start from here and employ the Taylor Expansion to the third power (assuming uniform meshes $\Delta\tau = \tau_{n+1} - \tau_n$, $\Delta S = \|A_i\| = S_{i+1} - S_i$):

$$\begin{aligned} \frac{V_i^{n+1} - V_i^n}{\Delta\tau} &= \frac{V_i^{n+1} - (V_i^{n+1} - V_{i_\tau}^{n+1}\Delta\tau + \frac{1}{2}V_{i_{\tau\tau}}^{n+1}\Delta\tau^2 - K_1\Delta\tau^3)}{\Delta\tau} \\ &= V_{i_\tau}^{n+1} - \frac{1}{2}V_{i_{\tau\tau}}^{n+1}\Delta\tau + K_1\Delta\tau^2; \end{aligned} \quad (4.2)$$

$$\begin{aligned} \frac{rS_i}{2\|A_i\|}[V_{i+1}^{n+1} - V_{i-1}^{n+1}] &= \frac{rS_i}{2\|A_i\|}[(V_i^{n+1} + V_{i_S}^{n+1}\Delta S + \frac{1}{2}V_{i_{SS}}^{n+1}\Delta S^2 + K_{2a}\Delta S^3) \\ &\quad - (V_i^{n+1} - V_{i_S}^{n+1}\Delta S + \frac{1}{2}V_{i_{SS}}^{n+1}\Delta S^2 - K_{2b}\Delta S^3)] \\ &= rS_iV_{i_S}^{n+1} + K_2\Delta S^2; \end{aligned} \quad (4.3)$$

$$\begin{aligned} &\frac{\sigma^2 S_i^2}{2\|A_i\|}[\frac{V_{i+1}^{n+1} - V_i^{n+1}}{S_{i+1} - S_i} - \frac{V_i^{n+1} - V_{i-1}^{n+1}}{S_i - S_{i-1}}] \\ &= \frac{\sigma^2 S_i^2}{2\Delta S}[\frac{(V_i^{n+1} + V_{i_S}^{n+1}\Delta S + \frac{1}{2}V_{i_{SS}}^{n+1}\Delta S^2 + \frac{1}{6}V_{i_{SSS}}^{n+1}\Delta S^3 + K_{3a}\Delta S^4) - V_i^{n+1}}{\Delta S} \\ &\quad - \frac{V_i^{n+1} - (V_i^{n+1} - V_{i_S}^{n+1}\Delta S + \frac{1}{2}V_{i_{SS}}^{n+1}\Delta S^2 - \frac{1}{6}V_{i_{SSS}}^{n+1}\Delta S^3 + K_{3b}\Delta S^4)}{\Delta S}] \\ &= \frac{\sigma^2 S_i^2}{2}V_{i_{SS}}^{n+1} + K_3\Delta S^2; \end{aligned} \quad (4.4)$$

where $K_2 = \frac{rS_i}{2}[K_{2a} + K_{2b}]$; $K_3 = \frac{\sigma^2 S_i^2}{2}[K_{3a} + K_{3b}]$.

Therefore, according to (4.1) with (4.2) (4.3) (4.4), we have:

$$\varepsilon = V_{i_\tau}^{n+1} - \frac{1}{2}V_{i_{\tau\tau}}^{n+1}\Delta\tau + K_1\Delta\tau^2 - rS_iV_{i_S}^{n+1} - K_2\Delta S^2 - \frac{\sigma^2 S_i^2}{2}V_{i_{SS}}^{n+1} - K_3\Delta S^2 + rV_i^{n+1}; \quad (4.5)$$

Recall from (2.1) we have:

$$V_{i_\tau}^{n+1} = rS_iV_{i_S}^{n+1} + \frac{\sigma^2 S_i^2}{2}V_{i_{SS}}^{n+1} - rV_i^{n+1}; \quad (4.6)$$

This will help us cancel out the $V_{i_\tau}^{n+1}$, $rS_iV_{i_S}^{n+1}$, $\frac{\sigma^2 S_i^2}{2}V_{i_{SS}}^{n+1}$, rV_i^{n+1} terms, leaving:

$$\varepsilon = -\frac{1}{2}V_{i_{\tau\tau}}^{n+1}\Delta\tau + K_1\Delta\tau^2 - K_2\Delta S^2 - K_3\Delta S^2; \quad (4.7)$$

So the truncation error is $O(\Delta\tau, \Delta S^2)$.

Next, we analyze the stability. Again, recall (2.8):

$$V_k^n \approx \alpha_{k-1}V_{k-1}^{n+1} + \alpha_k V_k^{n+1} + \alpha_{k+1}V_{k+1}^{n+1}; \quad (2.8)$$

where with uniform meshes:

$$\begin{aligned}\alpha_{k-1} &= -\frac{\sigma^2 S_k^2 \Delta \tau}{2\Delta S^2} + \frac{S_k r \Delta \tau}{2\Delta S}; \\ \alpha_k &= 1 - \frac{\sigma^2 S_k^2 \Delta \tau}{\Delta S^2} + r \Delta \tau; \\ \alpha_{k+1} &= -\frac{\sigma^2 S_k^2 \Delta \tau}{2\Delta S^2} - \frac{S_k r \Delta \tau}{2\Delta S};\end{aligned}\tag{4.8}$$

Writing $V_k^n = \hat{e}_n^{(j)} e^{ijk\Delta S}$, $V_k^{n+1} = \hat{e}_{n+1}^{(j)} e^{ijk\Delta S}$, then we have:

$$\hat{e}_n^{(j)} e^{ijk\Delta S} = \alpha_{k-1} \hat{e}_{n+1}^{(j)} e^{ij(k-1)\Delta S} + \alpha_k \hat{e}_{n+1}^{(j)} e^{ijk\Delta S} + \alpha_{k+1} \hat{e}_{n+1}^{(j)} e^{ij(k+1)\Delta S};\tag{4.9}$$

$$\hat{e}_n^{(j)} = \alpha_{k-1} \hat{e}_{n+1}^{(j)} e^{-ij\Delta S} + \alpha_k \hat{e}_{n+1}^{(j)} + \alpha_{k+1} \hat{e}_{n+1}^{(j)} e^{ij\Delta S};\tag{4.10}$$

We take in all coefficients:

$$\begin{aligned}\hat{e}_n^{(j)} &= \left[-\frac{\sigma^2 S_k^2 \Delta \tau}{2\Delta S^2} + \frac{S_k r \Delta \tau}{2\Delta S}\right] \hat{e}_{n+1}^{(j)} e^{-ij\Delta S} + \left[1 - \frac{\sigma^2 S_k^2 \Delta \tau}{\Delta S^2} + r \Delta \tau\right] \hat{e}_{n+1}^{(j)} \\ &\quad + \left[-\frac{\sigma^2 S_k^2 \Delta \tau}{2\Delta S^2} - \frac{S_k r \Delta \tau}{2\Delta S}\right] \hat{e}_{n+1}^{(j)} e^{ij\Delta S};\end{aligned}\tag{4.11}$$

Hence:

$$\hat{e}_n^{(j)} = \hat{e}_{n+1}^{(j)} \left[1 + \frac{\sigma^2 S_k^2 \Delta \tau}{\Delta S^2} (2 - 2\cos(j\Delta S)) + r \Delta \tau - i \frac{S_k r \Delta \tau}{\Delta S} \sin(j\Delta S)\right]\tag{4.12}$$

Since $|1 + \frac{\sigma^2 S_k^2 \Delta \tau}{\Delta S^2} (2 - 2\cos(j\Delta S)) + r \Delta \tau - i \frac{S_k r \Delta \tau}{\Delta S} \sin(j\Delta S)| \geq \sqrt{1 + r^2 \Delta \tau^2} > 1$, so the method is stable as $|\hat{e}_{n+1}^{(j)}| < |\hat{e}_n^{(j)}|$.

5. Conclusion. In this paper, we have described a Finite-Volume Method for option pricing using Black-Schole Model. We integrated the function on a small mesh $[S_{i-\frac{1}{2}}, S_{i+\frac{1}{2}}] \times [\tau_n, \tau_{n+1}]$ and generated the relation between the price of option at t and $t + \Delta t$ in a discrete form:

$$V_i^n \approx \alpha_{i-1} V_{i-1}^{n+1} + \alpha_i V_i^{n+1} + \alpha_{i+1} V_{i+1}^{n+1};$$

where

$$\begin{aligned}\alpha_{i-1} &= -\frac{\sigma^2 S_i^2 \Delta \tau}{(S_{i+1} - S_{i-1})(S_i - S_{i-1})} + \frac{S_i r \Delta \tau}{S_{i+1} - S_{i-1}}; \\ \alpha_i &= 1 - \frac{\sigma^2 S_i^2 \Delta \tau}{(S_i - S_{i-1})(S_{i+1} - S_i)} + r \Delta \tau; \\ \alpha_{i+1} &= -\frac{\sigma^2 S_i^2 \Delta \tau}{(S_{i+1} - S_{i-1})(S_{i+1} - S_i)} - \frac{S_i r \Delta \tau}{S_{i+1} - S_{i-1}};\end{aligned}$$

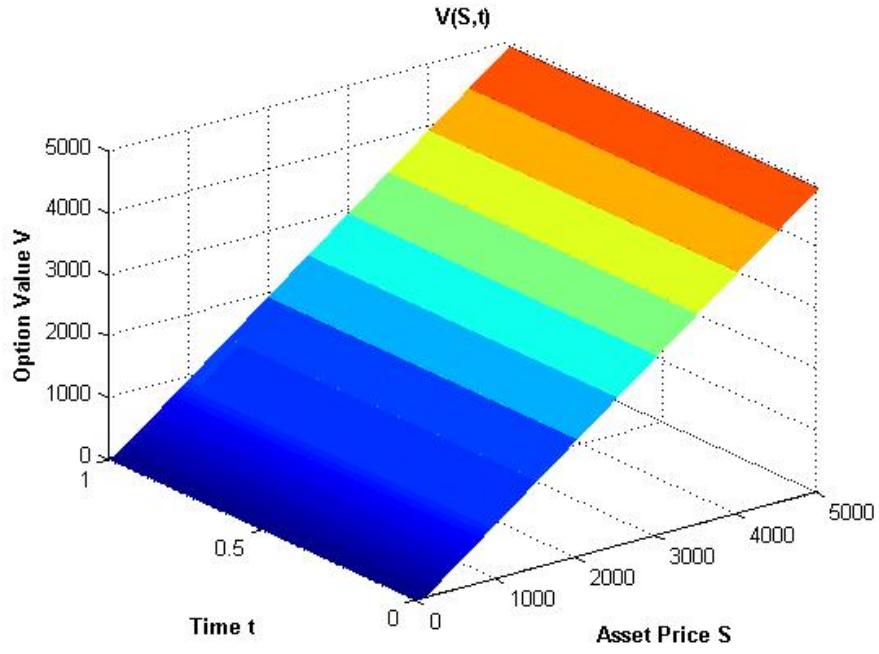
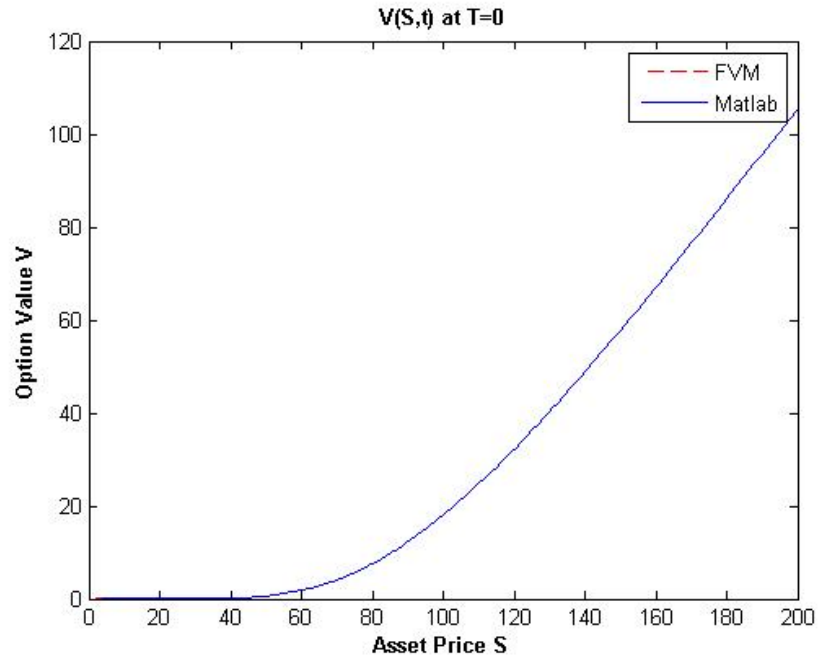
Then we applied the formula in Matlab and compare the numerical result of the method with the analytic solution of the Matlab built-in function. The result shows a high accuracy with $\Delta S = 0.1$ when $S \in [80, 120]$, $\Delta \tau = \frac{1}{365 \times 8}$ (about \$0.002 per share when $S \in [80, 120]$). The running time of

this mesh was about 8 time of that of the matlab built-in function. However, one could choose a bigger mesh and still got a good accuracy (less than \$0.01 per share) and reduce the running time close to or even less than that of the matlab built-in function.

We further studied the truncation error and stability of this method. The accuracy turned out to be $O(\Delta\tau, \Delta S^2)$ and the method is unconditionally stable. Both show the good feasibility of the method. For future study, we can try to apply numerical methods to solve equation(2.7) more efficiently.

REFERENCES

- [1] John Hull. *Options Futures and Other Derivatives*. Pearson Education, 2006.
- [2] Fischer Black and Myron Scholes. The pricing of options and corporate liabilities. *Journal of political economy*, 81(3):637–654, May 1973.
- [3] Gonzalo Cortazar. Simulation and numerical methods in real options valuation. *EFMA 2000 Athens*, December 2000.
- [4] Robert Geske and Kuldeep Shastri. Valuation by approximation: a comparison of alternative option valuation techniques. *Journal of Financial and Quantitative Analysis*, 20(1):45–71, 1985.
- [5] Domingo Tavella and Curt Randall. *Pricing financial instruments: The finite difference method*, volume 13. John Wiley & Sons, 2000.
- [6] Francis A Longstaff and Eduardo S Schwartz. Valuing american options by simulation: a simple least-squares approach. *Review of Financial studies*, 14(1):113–147, 2001.
- [7] Leonard CG Rogers. Monte carlo valuation of american options. *Mathematical Finance*, 12(3):271–286, 2002.
- [8] A Andalaft-Chacur, M Montaz Ali, and J González Salazar. Real options pricing by the finite element method. *Computers & Mathematics with Applications*, 61(9):2863–2873, 2011.
- [9] A Golbabai, LV Ballestra, and D Ahmadian. Superconvergence of the finite element solutions of the black–scholes equation. *Finance Research Letters*, 10(1):17–26, 2013.
- [10] SV Patankar. *Numerical Heat Transfer and Fluid Flow*. McGraw Hill, 1980.
- [11] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.
- [12] Henk Kaarle Versteeg and Weeratunge Malalasekera. *An introduction to computational fluid dynamics: the finite volume method*. Pearson Education, 2007.
- [13] Song Wang. A novel fitted finite volume method for the black–scholes equation governing option pricing. *IMA Journal of Numerical Analysis*, 24(4):699–720, 2004.

Fig. 3.1: European Call Option: $S \times T = [0,5000] \times [0,1]$ Fig. 3.2: European Call Option: $S \times T = [0,200] \times [0,1]$

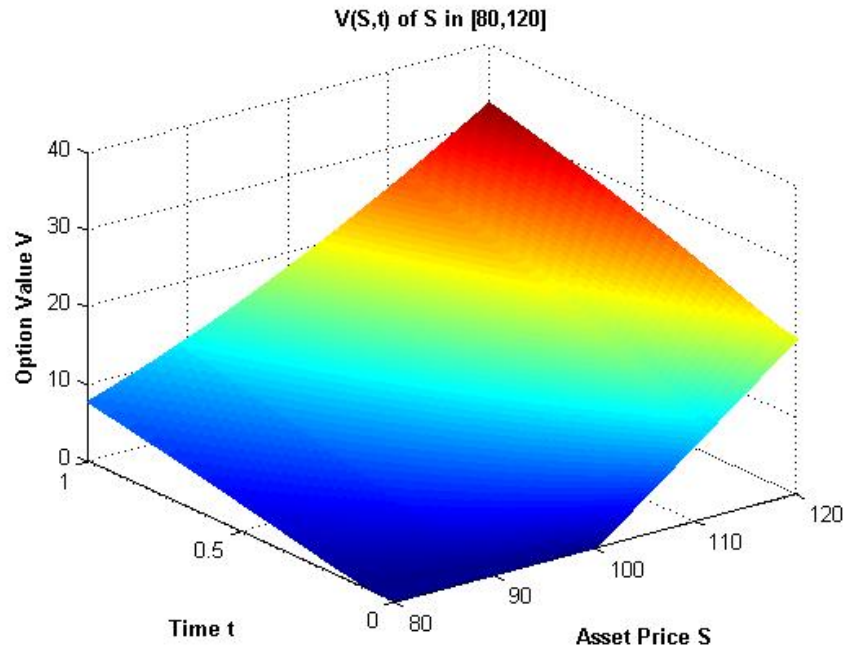


Fig. 3.3: European Call Option: $S \times T = [80,120] \times [0,1]$

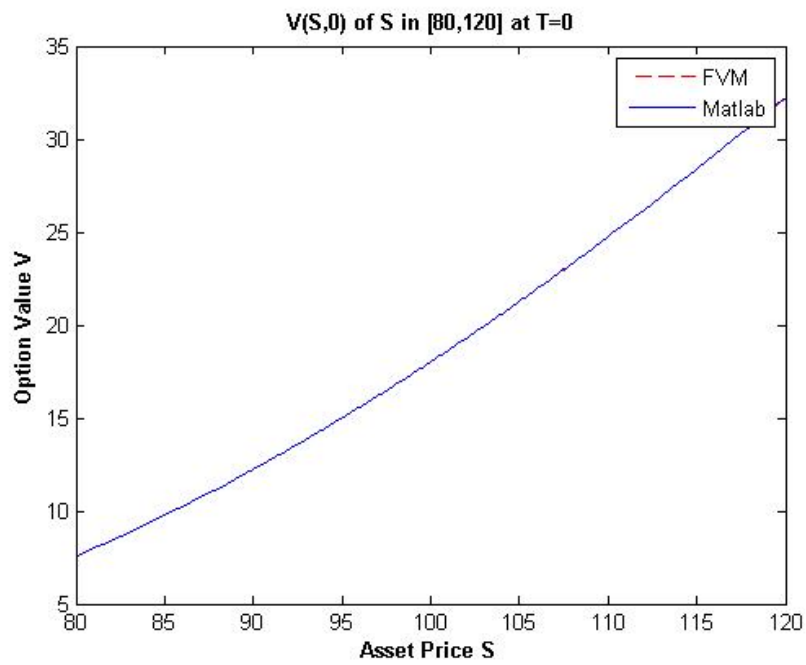
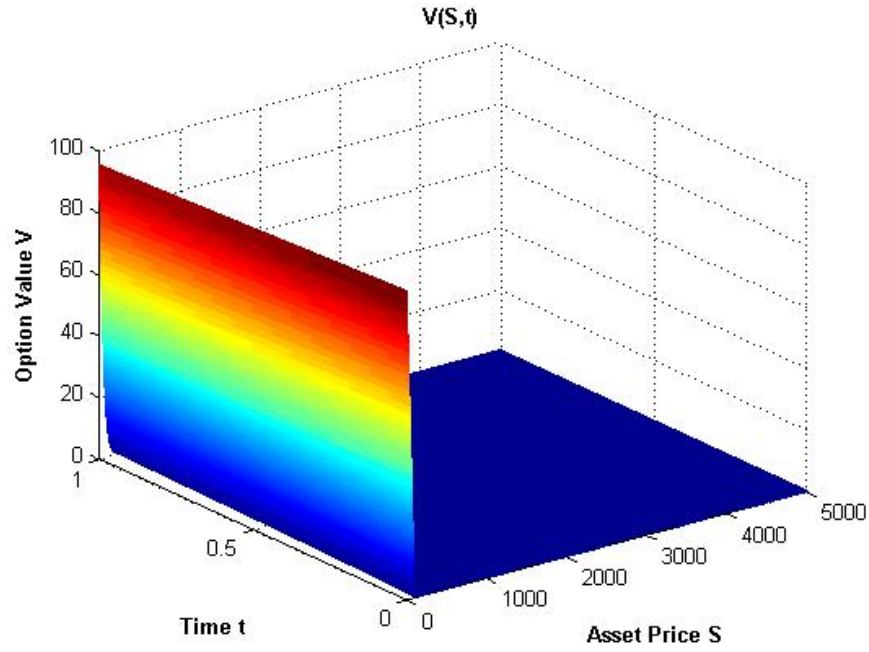
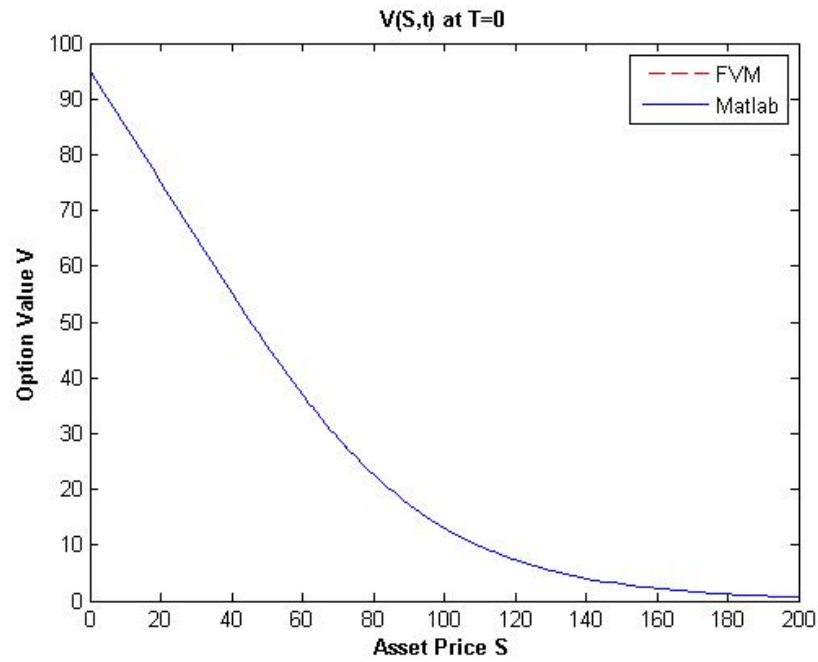


Fig. 3.4: European Call Option: $S \times T = [80,120] \times [0,1]$

Fig. 3.5: European Put Option: $S \times T = [0, 5000] \times [0, 1]$ Fig. 3.6: European Put Option Compared with Matlab: $S \times T = [0, 200] \times 1$

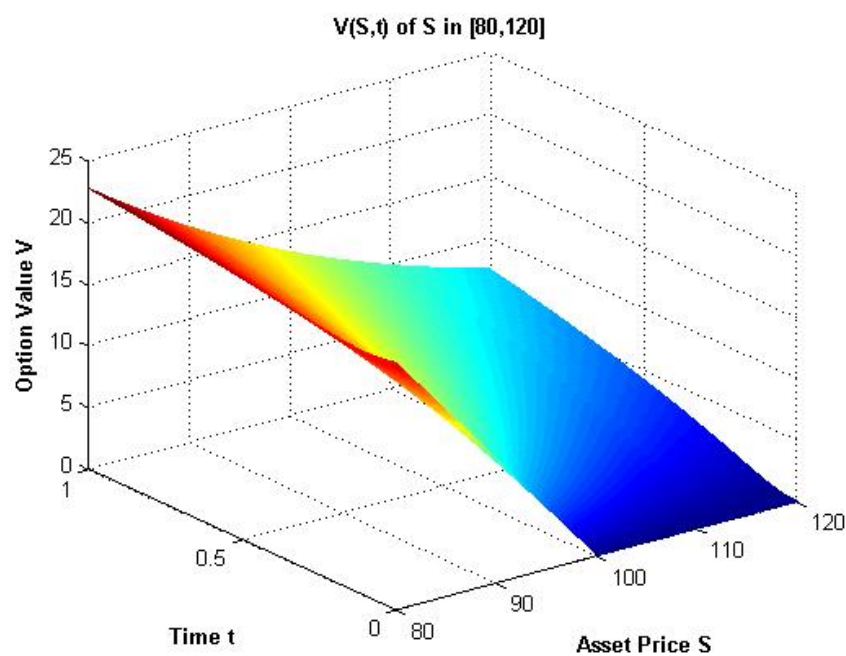


Fig. 3.7: European Put Option: $S \times T = [80,120] \times [0,1]$

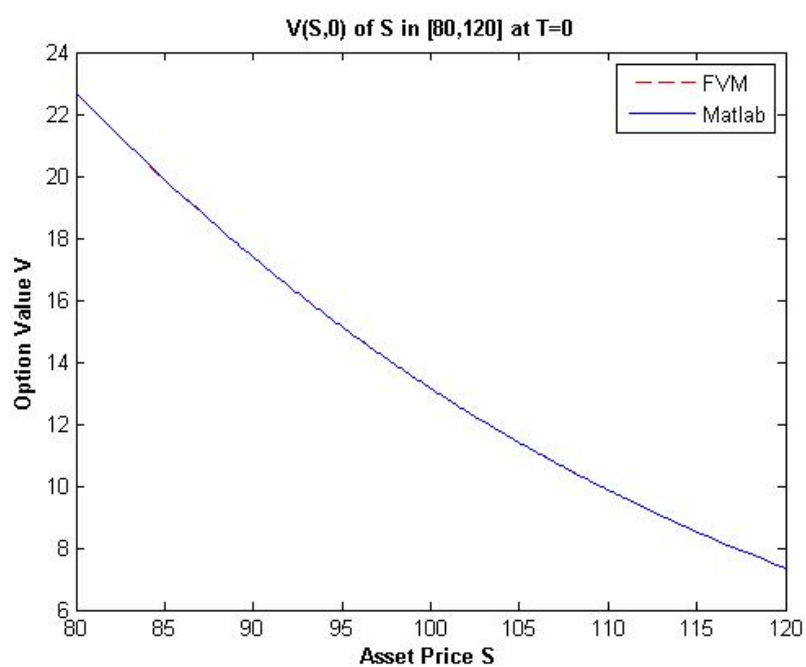


Fig. 3.8: European Put Option Compared with Matlab: $S \times T = [80,120] \times 1$