Numerical Analysis Final Project

NUMERICAL METHOD IN BASKET OPTION PRICING

May 10, 2016

Niansi Zhang nz116@scarletmail.rutgers.edu
Shuyi Wang sw669@scarletmail.rutgers.edu
Xiaolin Shen xs103@scarletmail.rutgers.edu
Yenhsiu Chang yc707@scarletmail.rutgers.edu
Shu Xie sx48@scarletmail.rutgers.edu

Introduction

Barrier basket Option is an exotic option whose underlying is a (weighted) sum or average of different assets that have been grouped together in a basket and whose existence depends upon the underlying asset's price reaching pre-set barrier level. In this project, we price the most basic barrier basket option: up-and-out european call. The pay-off function can be written as:

$$\begin{aligned} &c(T) = \max(S_1(T) + S_2(T) - K, 0) \\ &S_1(\tau) + S_2(\tau) < S_b, \forall \tau \leq T \end{aligned} \tag{1}$$

We analyze the solution in 3 different methods: (a) Monte-Carlo Simulation (b)Binomial Tree Method (c)Finite Difference method. Also, we include the analysis of its greeks by different methods.

Please Note: The input.csv is the 'test3' in our report. The Tree and Finite Difference might take more than 10 minutes to finish all the parameters estimation.

Binomial Tree Method

Assuming S_1 and S_2 follow lognormal process with no dividend

$$\begin{aligned} dS_{1t} &= rS_{1t}dt + \sigma_1 S_{1t} dW_{1t} \\ dS_{2t} &= rS_{2t}dt + \sigma_2 S_{2t} dW_{2t} \\ dW_{1t}dW_{2t} &= \rho dt \end{aligned} \tag{2}$$

We transfrom the formula above into $X_t = \log(S_t)$, we have:

$$\begin{aligned} dX_{1t} &= \mu_{1t}dt + \sigma_1 dW_{1t} \\ dX_{2t} &= \mu_{2t}dt + \sigma_2 dW_{2t} \\ dW_{1t}dW_{2t} &= \rho dt \end{aligned} \tag{3}$$

Where $\mu_i = r - \frac{1}{2}\sigma_i^2$. Then we impliment risk neutral pricing on one step to find out the risk neutral transition probability.

$$\begin{aligned} \mu_1 \delta t = E[\delta X_1] = (p_{uu} + p_{ud})\delta x_1 - (p_{du} + p_{dd})\delta x_1 \\ \mu_2 \delta t = E[\delta X_2] = (p_{uu} + p_{du})\delta x_2 - (p_{ud} + p_{dd})\delta x_2 \end{aligned} \tag{4}$$

And the second moments:

$$\begin{aligned} \sigma_1^2 \delta t = E[(\delta X_1)^2] = (p_{uu} + p_{ud})\delta x_1^2 + (p_{du} + p_{dd})\delta x_1^2 \\ \sigma_2^2 \delta t = E[(\delta X_2)^2] = (p_{uu} + p_{du})\delta x_2^2 + (p_{ud} + p_{dd})\delta x_2^2 \\ (p_{uu} - p_{ud} - p_{du+p_{dd}})\delta x_1 \delta x_2 = E[\delta X_1 \delta X2] = \rho \sigma_1 \sigma_2 \delta t \end{aligned} \tag{5}$$

Then we can solve the parameters:

$$\begin{aligned} p_{uu} &= \frac{1}{4}[1 + \sqrt{\delta t}(\frac{\mu_1}{\sigma_1} + \frac{\mu_2}{\sigma_2}) + \rho] \\ p_{ud} &= \frac{1}{4}[1 + \sqrt{\delta t}(\frac{\mu_1}{\sigma_1} - \frac{\mu_2}{\sigma_2}) - \rho] \\ p_{du} &= \frac{1}{4}[1 + \sqrt{\delta t}(\frac{\mu_1}{\sigma_1} + \frac{\mu_2}{\sigma_2}) - \rho] \\ p_{dd} &= \frac{1}{4}[1 + \sqrt{\delta t}(\frac{\mu_1}{\sigma_1} - \frac{\mu_2}{\sigma_2}) + \rho] \end{aligned} \tag{6}$$

The recursive formula for the option value V from $t + \delta t$ to t is:

$$V = \begin{cases} 0 & \text{if} \quad S_1 + S_2 > S_b \\ e^{-r\delta t}(p_{uu}V_{uu} + p_{ud}V_u d + p_{du}V_{du} + p_{dd}V_{dd}) & \text{otherwise} \end{cases}$$

Below is the example:

$S_1 = S_2 = 100$ K=200, $S_b = 260$ r=0.01 $\sigma_1 = 0.3$ $\sigma_2 = 0.2$ T=1 $\rho = 0.5$

PRICE	DELTA1	DELTA2	SIGMA1	SIGMA2	VEGA1	VEGA2	THETA
5.4268	0.0368	0.0368	0.0107	-0.0099	-18.5902	-8.6073	3.2009

Monte-Carlo Simulation
From Geometric Brownian Motion under risk neutral measure

$$dS_t = rS_t dt + \sigma S_t dW_t$$

and using Ito's lemma, we can obtain

$$dlogS_t = (r - \frac{1}{2}\sigma^2)dt + \sigma dW_t$$

Therefore

$$S_{t+1} = S_t e^{(r-\frac{1}{2}\sigma^2)dt+\sigma\sqrt{\delta t}\epsilon} \quad \epsilon \sim N(0,1)$$

First, we use the above formula to run a pilot simulation for stock 1 and stock 2. In order to generate two correlated stocks, we can exploit Cholesky decomposition. The covariance matrix for the epsilons of the two stocks would be

$$\begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} = \begin{bmatrix} l_{11} & 0 \\ l_{21} & l_{22} \end{bmatrix} \begin{bmatrix} l_{11} & l_{21} \\ 0 & l_{22} \end{bmatrix}$$

Hence

$$l_{11} = 1 \quad l_{21} = \rho \quad l_{22} = \sqrt{1-\rho^2}$$

$$\begin{bmatrix} \epsilon_1 \\ \epsilon_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ \rho & \sqrt{1-\rho^2} \end{bmatrix} \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \end{bmatrix} \quad \varepsilon_1, \varepsilon_2 \sim N(0,1)$$

After using the above equation to generate ϵ_1 and ϵ_2 and generating the two simulated stock paths, we can compute the value C of the up-and-out European call on a basket of two stocks for each path. Then estimate the hedge ratio c^* with stock 1

$$c^* = -cov(S_{1j}, C_j)/var(S_{1T}) \quad S_{1T} = S_{10}^2 e^{2rT}(e^{\sigma^2 T} - 1)$$

Then repeat the above steps for a full simulation and estimate the option price by computing the sample average of $C_j + c^*(S_{1j} - S_{10}e^{rT})$

Below is the example:
$S_1 = S_2 = 100$ K=200, $S_b = 260$ r=0.01 $\sigma_1 = 0.3$ $\sigma_2 = 0.2$ T=1 $\rho = 0.5$

PRICE	DELTA1	DELTA2	SIGMA1	SIGMA2	VEGA1	VEGA2	THETA
5.3746	0.0269	0.0514	-0.002	0.026	-22.6163	-14.2236	4.1891

Finite Difference Method

Compared to Binomial Tree and Mote-Carlo Simulation, the finite difference method has less Time Complexity. Suppose we have N^2 for the grid, the complexity would be $N^2 \times T$. However, the other two methods should consider the repeated time of the experiment. Firstly, we need to have the PDE of this problem.

$$\frac{\partial \phi}{\partial t} + \frac{1}{2}\sigma_1^2 S_1^2 \frac{\partial^2 \phi}{\partial S_1^2} + \frac{1}{2}\sigma_2^2 S_2^2 \frac{\partial^2 \phi}{\partial S_2^2} + rS_1 \frac{\partial \phi}{\partial S_1} + rS_2 \frac{\partial \phi}{\partial S_2} + \rho\sigma_1\sigma_2 S_1 S_2 \frac{\partial^2 \phi}{\partial S_1 \partial S_2} = r\phi \tag{7}$$

Next, we need to discrete the PDE in order to do the finite difference calculation. We show the equation of implicit method below

The discretized form is:

$$\begin{aligned}
\phi_{i,j}^{k+1} =& (1 + \delta t\sigma_1^2 (i-1)^2 + \delta t\sigma_2^2 (j-1)^2 + \delta t r)\phi_{i,j}^k \\
&- (\frac{1}{2}\sigma_1^2 (i-1)^2 \delta t + \frac{1}{2}(i-1) r\delta t)\phi_{i+1,j}^k \\
&- (\frac{1}{2}\sigma_1^2 (i-1)^2 \delta t - \frac{1}{2}(i-1) r\delta t)\phi_{i-1,j}^k \\
&- (\frac{1}{2}\sigma_2^2 (j-1)^2 \delta t + \frac{1}{2}(j-1) r\delta t)\phi_{i,j+1}^k \\
&- (\frac{1}{2}\sigma_2^2 (j-1)^2 \delta t - \frac{1}{2}(j-1) r\delta t)\phi_{i,j-1}^k \\
&- \frac{1}{4}\rho\sigma_1\sigma_2 (i-1)(j-1)\delta t\phi_{i+1,j+1}^k \\
&+ \frac{1}{4}\rho\sigma_1\sigma_2 (i-1)(j-1)\delta t\phi_{i+1,j-1}^k \\
&+ \frac{1}{4}\rho\sigma_1\sigma_2 (i-1)(j-1)\delta t\phi_{i-1,j+1}^k \\
&- \frac{1}{4}\rho\sigma_1\sigma_2 (i-1)(j-1)\delta t\phi_{i-1,j-1}^k
\end{aligned} \tag{8}$$

From the equation above, we want to build the above equation into a matrix form that can be calculate recursively. Suppose we divide both S_1 and S_2 into N grids, we would have a $(N^2, 1)$ dimension vector as the price matrix at time k.

$$\phi^k = \begin{bmatrix} \phi_1^k \\ \phi_2^k \\ \phi_3^k \\ \vdots \\ \phi_{N-1}^k \\ \phi_N^k \end{bmatrix} = \begin{bmatrix} \phi_{1,1}^k \\ \phi_{1,2}^k \\ \vdots \\ \phi_{1,N}^k \\ \phi_{2,1}^k \\ \vdots \\ \phi_{N,N-1}^k \\ \phi_{N,N}^k \end{bmatrix}$$

We want to write down the recursion in the form:

$$U\phi^k = \phi^{k+1} \tag{9}$$

Thus, with the usage of the Neumann bondary condition, we can write down the block tridiagonal matrix as the form:

$$U = \begin{bmatrix} 2A(1)+B(1) & C(1)-A(1) & & & \\ A(2) & B(2) & C(2) & & \\ & A(3) & B(3) & C(3) & \\ & \ddots & \ddots & \ddots & \\ & & A(N-1) & B(N-1) & C(N-1) \\ & & & A(N)-C(N) & B(N)+2C(N) \end{bmatrix}$$

The U is $N^2 \times N^2$ dimension matrix. The matrix A, B and C are $N \times N$ shown below:

$$A(j) = \begin{bmatrix} 2a_9(1)+a_5(1) & a_7(1)-a_9(1) & & & \\ a_9(2) & a_5(2) & a_7(2) & & \\ & a_9(3) & a_5(3) & a_7(3) & \\ & \ddots & \ddots & \ddots & \\ & & a_9(N-1) & a_5(N-1) & a_7(N-1) \\ & & & a_9(N)-a_7(N) & a_5(N)+2a_9(N) \end{bmatrix}(j)$$

$$B(j) = \begin{bmatrix} 2a_3(1)+a_1(1) & a_2(1)-a_3(1) & & & \\ a_3(2) & a_1(2) & a_2(2) & & \\ & a_3(3) & a_1(3) & a_2(3) & \\ & \ddots & \ddots & \ddots & \\ & & a_3(N-1) & a_1(N-1) & a_2(N-1) \\ & & & a_3(N)-a_2(N) & a_1(N)+2a_3(N) \end{bmatrix}(j)$$

$$C(j) = \begin{bmatrix} 2a_8(1)+a_4(1) & a_6(1)-a_8(1) & & & \\ a_8(2) & a_4(2) & a_6(2) & & \\ & a_8(3) & a_4(3) & a_6(3) & \\ & \ddots & \ddots & \ddots & \\ & & a_8(N-1) & a_4(N-1) & a_6(N-1) \\ & & & a_8(N)-a_6(N) & a_4(N)+2a_9(N) \end{bmatrix}(j)$$

where:

$$\begin{aligned} a_1(i,j) &= 1 + \delta t \sigma_1^2 (i-1)^2 + \delta t \sigma_2^2 (j-1)^2 + \delta t r \\ a_2(i,j) &= -(\frac{1}{2}\sigma_1^2 (i-1)^2 \delta t + \frac{1}{2}(i-1) r \delta t) \\ a_3(i,j) &= -(\frac{1}{2}\sigma_1^2 (i-1)^2 \delta t - \frac{1}{2}(i-1) r \delta t) \\ a_4(i,j) &= -(\frac{1}{2}\sigma_2^2 (j-1)^2 \delta t + \frac{1}{2}(j-1) r \delta t) \\ a_5(i,j) &= -(\frac{1}{2}\sigma_2^2 (j-1)^2 \delta t - \frac{1}{2}(j-1) r \delta t) \\ a_6(i,j) &= a_9(i,j) = -\frac{1}{4}\rho \sigma_1 \sigma_2 (i-1)(j-1)\delta t \\ a_7(i,j) &= a_8(i,j) = \frac{1}{4}\rho \sigma_1 \sigma_2 (i-1)(j-1)\delta t \end{aligned} \tag{10}$$

From the observation of two images above, we can see that when the volatility different, it shows asymmetry in the graph.

Below is the example:
$S_1 = S_2 = 100$ K=200, $S_b = 260$ r=0.01 $\sigma_1 = 0.3$ $\sigma_2 = 0.2$ T=1 $\rho = 0.5$

PRICE	DELTA1	DELTA2	SIGMA1	SIGMA2	VEGA1	VEGA2	THETA
5.355378	0.052661	0.028421	-0.003654	-0.004158	-19.061378	-12.819503	3.837125

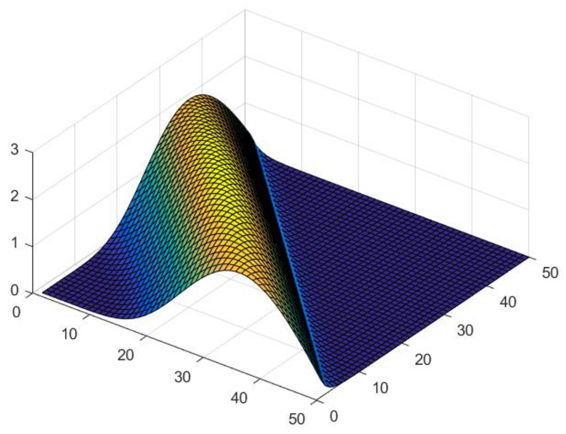

Figure 1: $\sigma_1 = \sigma_2 = 0.4$

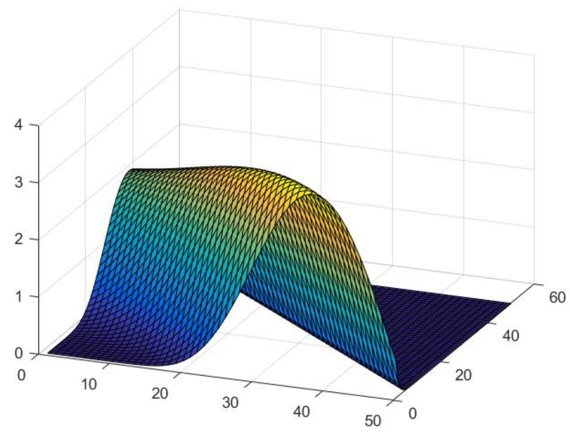

Figure 2: $\sigma_1 = 0.4 \sigma_2 = 0.3$

Figure 3: price at T=0 vs S_1 S_2

Observations and Analysis

Based on the results, the trend of option prices in the three methods is very consistent. Comparing test 1 with test 2, we can see that when the stock barrier decreases, option price will also decrease. In test 2 and test 3, decreased strike price can make the option price higher. And for test 3 and test 4, higher volatility will lower the option price. This may be due to that higher volatility will enhance the chance of cancellation of the barrier option. When it comes to test 4 and test 5, higher correlation of the two stocks will cause the option price to decrease. This is because higher correlation between the two stocks would make the sum of the two stocks have a higher chance to reach the barrier, thereby reducing the option price. For test 1 and test 6, increasing stock 2 price can make the option price higher, but we think that is because the stock barrier is relatively high, so increased stock price could still provide benefits in this case.
Regarding greeks, we observe that the consistency of the three methods is not very good. The reason is that using discretized scheme to estimate greeks may not be able to obtain greeks with reasonable accuracy, especially when we just use a displacement of one percent relating to original variables. Hence, this may be a direction for further improvement.
Results

Table 1: Test Input Parameters

	test1	**test2**	**test3**	**test4**	**test5**	**test6**
Current stock price1	100	100	100	100	100	100
Current stock price2	100	100	100	100	100	120
Barrier	300	260	260	260	260	300
Strike	220	220	200	200	200	220
Risk free rate	0.01	0.01	0.01	0.01	0.01	0.01
Option expiration	1	1	1	1	1	1
Stock volatility1	0.3	0.3	0.3	0.4	0.4	0.3
Stock volatility2	0.2	0.2	0.2	0.4	0.4	0.2
Correlation	0.5	0.5	0.5	0.5	0.7	0.5
Method selection - 1: MC; 2: FD; 3: Tree	1	1	1	1	1	1
Number of time step	1000	1000	1000	1000	1000	1000
Number of paths for Monte Carlo	20000	20000	20000	20000	20000	20000

Table 2: Prices

	test1	test2	test3	test4	test5	test6
Monte Carlo(Control Variate)	5.9959	1.6136	5.3746	2.2378	1.9682	8.9266
Finite Difference	5.998614	1.55874	5.355378	2.186403	1.908255	8.380796
Tree Method	6.0828	1.6069	5.4268	2.2241	1.9274	9.0181

Table 3: Greeks

Monte Carlo(Control Variate)						
incr = 0.01	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**
delta1	0.1272	0.0077	0.0269	0.0052	0.0158	0.0937
delta2	0.1266	0.0148	0.0514	-0.0024	0.0132	0.1119
gamma1	-0.0562	-0.0037	-0.002	0.0156	0.0201	-0.0181
gamma2	-0.0507	0.0099	0.026	0.0099	0.0234	-0.0152
vega1	-9.0339	-7.5474	-22.6163	-5.3971	-2.6673	-26.9894
vega2	-10.3595	-5.0594	-14.2236	-3.4423	-1.8607	-23.82
theta	2.2686	1.2964	4.1891	1.5695	0.966	5.258
Tree Method						
incr = 0.01	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**
delta1	0.1503	0.0161	0.0368	0.0293	0.0358	0.097
delta2	0.1508	0.0121	0.0368	0.0293	0.0358	0.1129
gamma1	-0.0038	0.0064	0.0107	0.0822	0.0952	-0.0131
gamma2	-0.0082	-0.0039	-0.0099	0.0822	0.0952	-0.0185
vega1	-6.5254	-5.7148	-18.5902	-1.7434	-0.7733	-22.7194
vega2	-6.6769	-1.7083	-8.6073	-1.7434	-0.7733	-11.2528
theta	1.0712	0.8222	3.2009	-0.9547	-0.9904	5.8428
Finite Difference						
incr = 0.01	**test1**	**test2**	**test3**	**test4**	**test5**	**test6**
delta1	0.152364	0.018724	0.052661	-0.01359	-0.01382	0.108888
delta2	0.14638	0.011284	0.028421	-0.01359	-0.01382	0.080045
gamma1	0.000585	-0.00103	-0.00365	-0.0011	-0.00086	-0.002925
gamma2	-0.00037	-0.00122	-0.00416	-0.001	-0.00081	-0.003816
vega1	-5.5056	-5.82143	-19.0614	-6.28702	-5.60677	-26.11264
vega2	0.386539	-3.57728	-12.8195	-6.28702	-5.60677	-12.15071
theta	0.625278	1.128732	3.837125	2.316398	2.065498	4.758635

Table 4: Computer Memory Size: 8GB, CPU: 2.7-3.5GHz

	Monte Carlo	Finite Difference	Binomial Tree
Time	1.571184	72.1303	26.312

The table above are the elapsed time for pricing using the three methods

Appendix

$$
\begin{aligned}
&dS = S_0 * (1 + 0.01) - S_0 = S_0 * 0.01 \\
&\Delta \equiv \frac{\partial f}{\partial S} = \frac{df}{dS} = \frac{f(S_0 * (1 + 0.01), t) - f(S_0, t)}{S_0 * (1 + 0.01) - S_0} \\
&\Gamma \equiv \frac{\partial^2 f}{\partial S^2} = \frac{d\Delta}{dS} = \frac{\Delta_1 - \Delta_2}{dS} \\
&\Delta_1 = \frac{f(S_0 * (1 + 0.01), t) - f(S_0, t)}{S_0 * (1 + 0.01) - S_0} \\
&\Delta_2 = \frac{f(S_0 * (1 - 0.01), t) - f(S_0, t)}{S_0 * (1 - 0.01) - S_0}
\end{aligned}
\tag{11}
$$