Final Project Report

Pricing vanilla options

Takumi Li

IE 420

Spring 2020

Introduction

This project is about pricing European and American vanilla options using the binomial model. For the purpose of modelling and getting more accurate results, various parameters are considered: option type (Call or Put), strike price (K), time to maturity (T), initial stock price (S_0), volatility (σ), continuous compounding risk free interest rate (r), continuous dividend yield(q), number of time steps (N), and exercise type (American or European).

For the purpose of pricing an option, the computation speed is extremely important in financial industry, and thus C++ is chosen to implement this project. Cox-Ross-Rubinstein binomial model was used as the underlying model, and the convergence of the CRR model to the Black-Scholes model was discovered; this project also attempted to verify the relationship of change of option price and early exercise boundary for American options.

In order to reduce computational time, <u>choosing C++</u> and implementing CRR binomial model with <u>backward induction</u> were greatly helpful.

Model Formulas

1. CRR Binomial Model

Derivative payoff:

$$f_u = f(uS_0), f_d = f(dS_0)$$

$$u = e^{\sigma\sqrt{\delta}}, d = e^{-\sigma\sqrt{\delta}}, \delta = T/N$$

Risk neutral pricing formula (backward induction):

$$f_u = e^{-r\delta}(p^* f_{uu}) + (1 - p^*)f_{ud}$$

$$f_d = e^{-r\delta}(p^* f_{ud}) + (1 - p^*)f_{dd}$$

$$f_0 = e^{-r\delta}(p^* f_u) + (1 - p^*)f_d$$

the risk neutral probability: $p^* = \frac{e^{(r-q)\delta} - \mathrm{d}}{u-d}$

<u>Parameters:</u>

S_0: the initial stock price

T: the time to maturity

N: the number of time steps

r: the continuous compounding risk free interest rate

q: the continuous dividend yield

2. Black-Scholes Model

European call price:

$$c = S_0 e^{-qT} N(d_1) - K e^{-rT} N(d_2)$$

European put price:

$$p = -S_0 e^{-qT} N(-d_1) + K e^{-rT} N(-d_2)$$

where $N(x)$ is the cdf of $N(0,1)$,

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r - q + \frac{1}{2}\sigma^2\right)T}{\sigma\sqrt{T}}, d_2 = d_1 - \sigma\sqrt{T}$$

<u>Parameters:</u>

S_0: the initial stock price

T: the time to maturity

N: the number of time steps

r: the continuous compounding risk free interest rate

q: the continuous dividend yield

K: the strike price

Problems

1. Implement the CRR binomial model to price European and American puts and calls on a stock paying continuous dividend yield:

$$Binomial(Option, K, T, S_0, \sigma, r, q, N, Exercise)$$

where Option = C for calls and P for puts, K is the strike, T is the time to maturity, S0 is the initial stock price, σ is the volatility, r is the continuous compounding risk free interest rate, q is the continuous dividend yield, N is the number of time steps, and Exercise = A for American options and E for European options. Your program should output both option price and computational time.

Solution:

Implement the CRR binomial model is included in the codes “**binomial.cpp**”. Backwards induction is used in order to improve performance. The binomial program should output both option price and computational time.

For example: When inputting the following parameters, the computed option price result and computational time is shown. Detailed implementation is included in “**binomial.cpp**”.

```cpp
double question1()
{
    /* Edit parameters HERE */
    paramType params;
    params.option = P;
    params.k = 100;
    params.t = 1.0;
    params.s0 = 100;
    params.sigma = 0.25;
    params.r = 0.02;
    params.q = 0.01;
    params.n = 100;
    params.exercise = A;
    params.ifUandD = false;

    BinomialReturnType bionomialResult;
    bionomialResult = binomial(params);
    cout << "Option price = " << bionomialResult.optionPrice << endl;
    cout << "computational Time = " << bionomialResult.computationalTime << "s" << endl;
    return bionomialResult.optionPrice;
}
```

```
takumi@takumi-GE60-2PL: ~/UIUC/420/project/codes
File  Edit  View  Search  Terminal  Help
takumi@takumi-GE60-2PL:~/UIUC/420/project/codes$ ./output 1
For question 1, please edit 'question1()' in main.cpp for the arguments of Binomial model.
Option price = 9.39578, computational Time = 0.000949s
takumi@takumi-GE60-2PL:~/UIUC/420/project/codes$
```

2. Consider a 1-year European call option with strike K = 100. The current stock price is 100. Other parameters are r = 0.05, q = 0.04, σ = 0.2. Use the Black-Scholes formula to compute the price of the call option. In the binomial model, take a sequence of increasing numbers of time steps N. Verify that the binomial option prices converge to the Black-Scholes option price as N increases. Construct a table/plot to visualize the convergence.

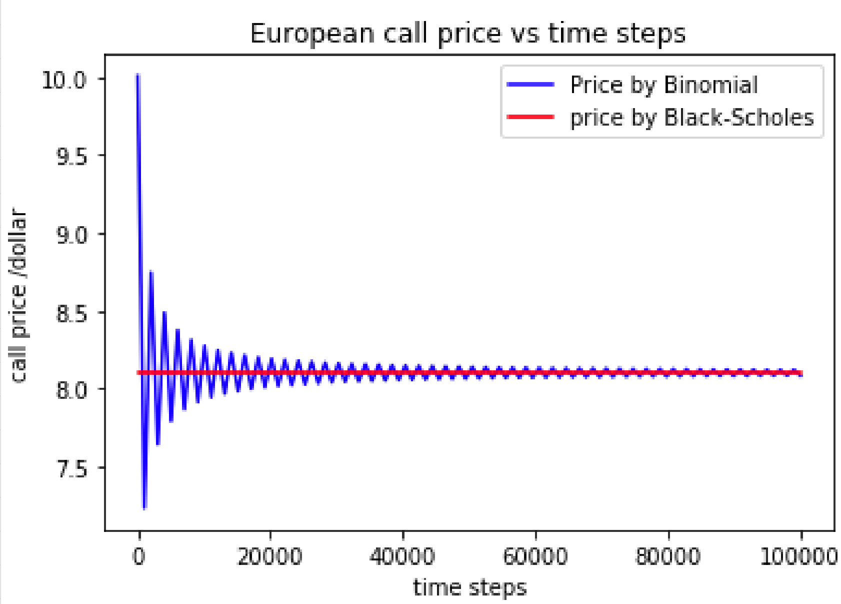
Solution:

Black-Scholes option price = <u>8.10264</u>

As shown in "**q2_result.csv**" and " **q2_result_v2.csv**".

From the table and the graph, we can clearly see the **convergence** of these two models as N grows large.

A	B
N	Price
5000	8.10226
10000	8.10245
15000	8.10252
20000	8.10255
25000	8.10257
30000	8.10258
35000	8.10259
40000	8.1026
45000	8.1026
50000	8.10261
55000	8.10261
60000	8.10261
65000	8.10261
70000	8.10262
75000	8.10262
80000	8.10262
85000	8.10262
90000	8.10262
95000	8.10262
100000	8.10262



3.

Consider American puts with $K = 100$, $\sigma = 0.2$, $r = 0.05$, $q = 0$. For time to maturity varying from 1 month to 12 months, investigate the number of time steps needed to achieve the required 10^{-3} accuracy. Calculate and plot the price of a 12-month put as a function of S_0. For time to maturity $i/12$, $i = 0, 1, \cdots, 12$, find the critical stock price $S^*(i)$ on the early exercise boundary (see notes below about how to determine $S^*(i)$). Report and plot the early exercise boundary $\{S^*(i), 0 \leq i \leq 12\}$ as a function of the time to maturity. Repeat the above for $q = 0.04$. How do put prices and the early exercise boundary change when the continuous dividend yield goes from 0 to 0.04? What is the intuition behind this dependence on the dividend yield?

Solution:

I decided to set N = 1000, which is enough to get an accuracy of 10^{-3}, with short computational time.

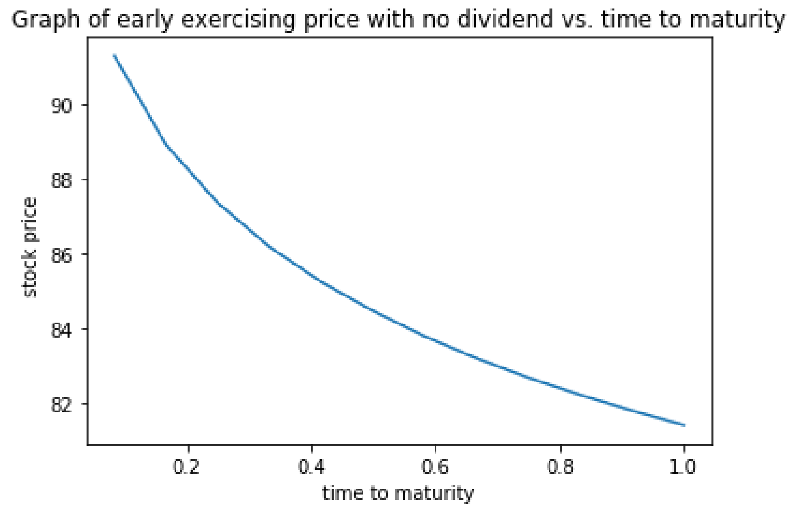
Output data are included as "**q3_000_option_prices.csv**", "**q3_004_option_prices.csv**", "**q3_000_critical.csv**", and "**q3_004_critical.csv**".

(1) q=0, The put price decreases as the S_0 increases, as shown in "**q3_000_option_prices.csv**".

(2) q=0, as shown in "**q3_000_ critical.csv**",

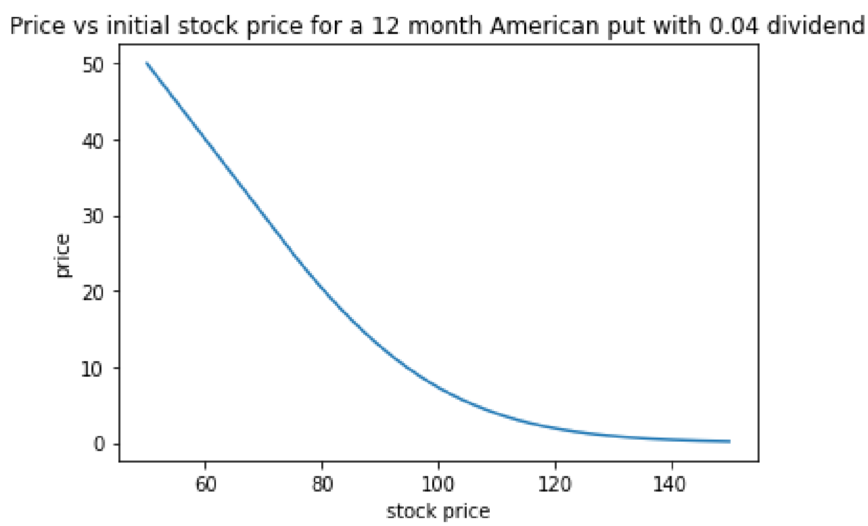
the early exercise boundary clearly had a negative relationship with time to maturity.

S_i	critical_Price
1	91.3
2	88.91
3	87.35
4	86.18
5	85.25
6	84.47
7	83.78
8	83.19
9	82.66
10	82.2
11	81.77
12	81.38



(3) q=0.04, The put price decreases as the S_0 increases, as shown in

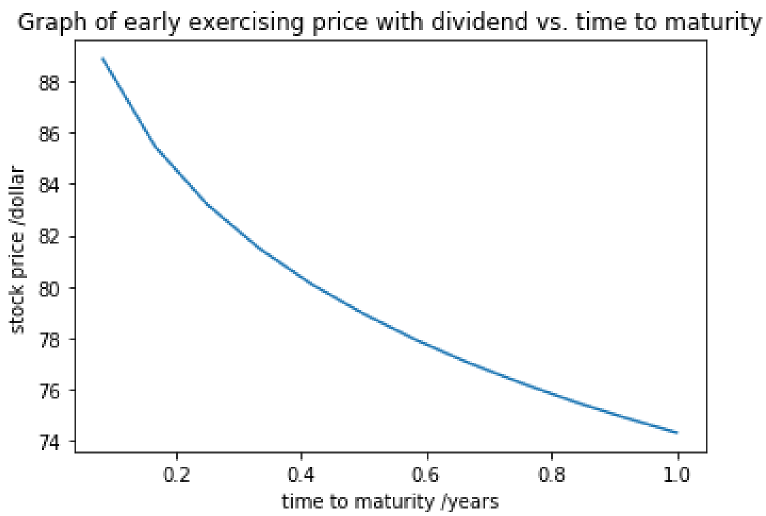
"**q3_004_option_prices.csv**".



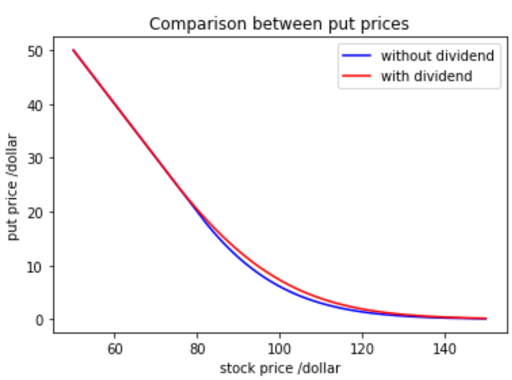
(4) q=0.04, as shown in “**q3_004_ critical.csv**”,

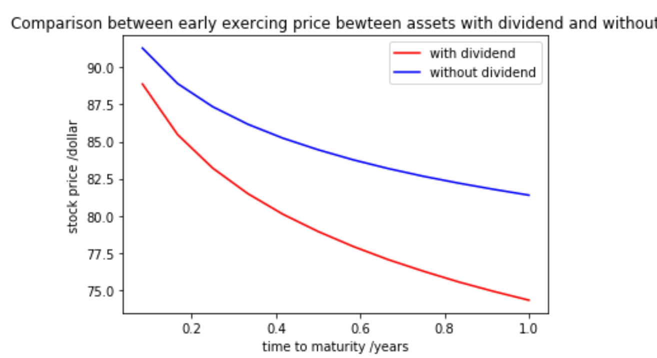
the early exercise boundary clearly had a negative relationship with time to maturity.

S_i	critical_Price
1	88.88
2	85.47
3	83.21
4	81.5
5	80.11
6	78.94
7	77.93
8	77.05
9	76.26
10	75.56
11	74.92
12	74.33



(5) Putting them together, we found that the put price with dividend is slightly more expensive and the early exercise boundary gets lower. The intuition behind this would be when the continuous dividend yield exists (or is larger), the option holder is facing less risk.





4.

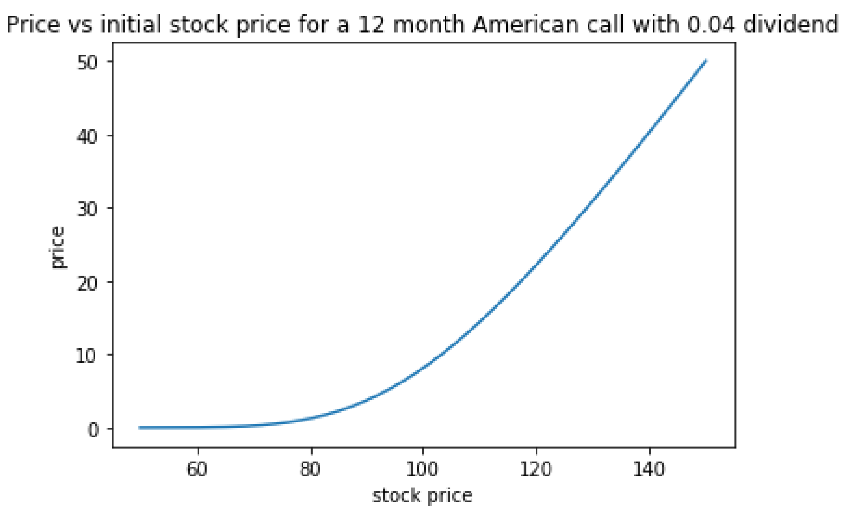
Consider American calls with $K = 100$, $\sigma = 0.2$, $r = 0.05$, $q = 0.04$. Investigate the number of time steps needed to achieve 10^{-3} accuracy for 1-month to 12-month calls. Calculate and plot the price of a 12-month call as a function of S_0. For time to maturity $i/12$, $i = 0, 1, \cdots, 12$, find the critical stock price $S^*(i)$ on the early exercise boundary. Report and plot the early exercise boundary $\{S^*(i), 0 \le i \le 12\}$ as a function of the time to maturity. Repeat the above for $q = 0.08$. How do call prices and the early exercise boundary change when the continuous dividend yield goes from 0.04 to 0.08? What is the intuition behind this dependence on the dividend yield?

Solution:

I decided to set N = 1000, which is enough to get an accuracy of 10^{-3}, with short computational time.

Output data are included as "**q4_004_option_prices.csv**", "**q4_008_option_prices.csv**", "**q4_004_critical.csv**", and "**q4_008_critical.csv**".

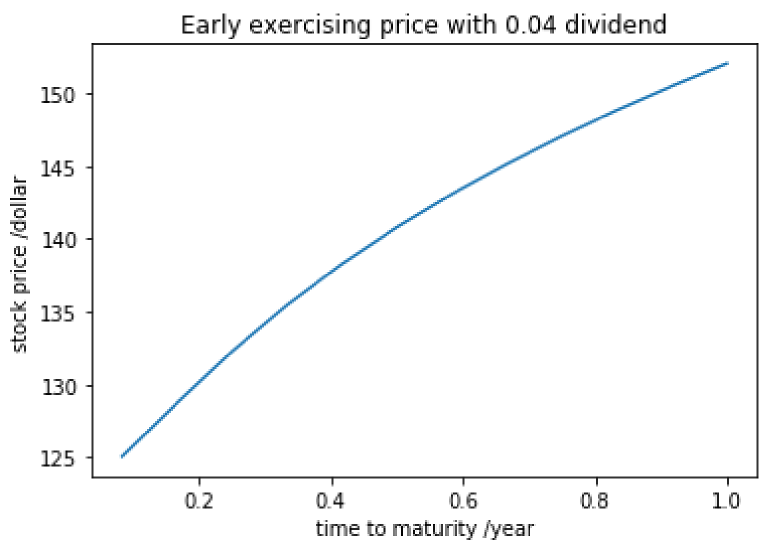
(1) q=0.04, The call price increases as the S_0 increases, as shown in "**q4_004_option_prices.csv**".



(2) q=0.04, as shown in “**q4_004_ critical.csv**”,

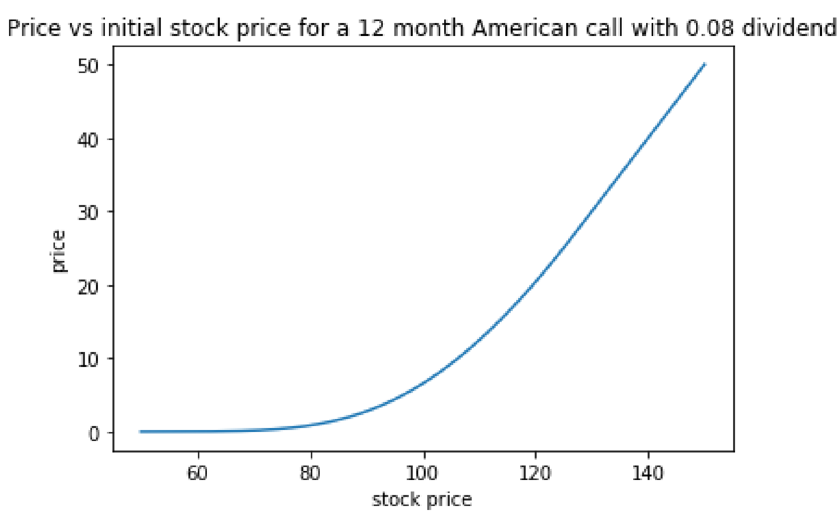
the early exercise boundary clearly had a positive relationship with time to maturity.

S_i	Crit_Price
1	125.06
2	128.73
3	132.27
4	135.46
5	138.28
6	140.79
7	143.06
8	145.13
9	147.04
10	148.82
11	150.48
12	152.02



(3) q=0.08, The call price increases as the S_0 increases, as shown in

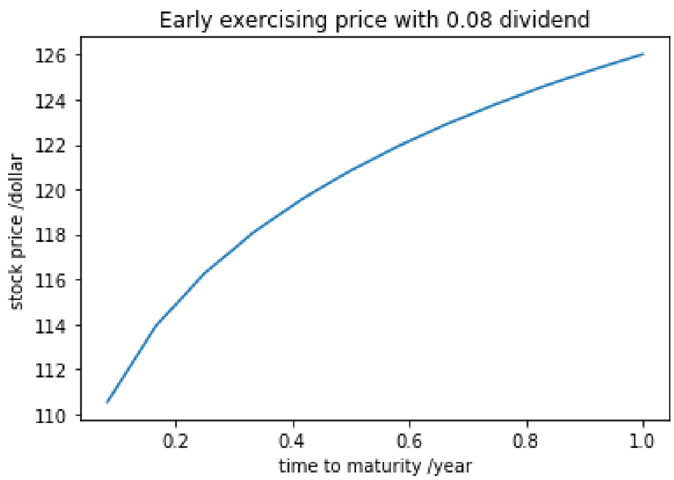
“**q4_008_option_prices.csv**”.



(4) q=0.08, as shown in "**q4_008_ critical.csv**",

the early exercise boundary clearly had a positive relationship with time to maturity.

S_i	Crit_Price
1	110.54
2	113.93
3	116.26
4	118.07
5	119.55
6	120.81
7	121.93
8	122.92
9	123.81
10	124.61
11	125.35
12	126.02



(5) Putting them together, as **q** goes from 0.04 to 0.08, we found that the call price with larger dividend (0.08) is slightly cheaper and the early exercise boundary gets higher. The intuition behind this would be when the continuous dividend yield exists (or is larger), the option holder is facing less risk, and thus is less likely to face the cost for higher price.