Financial Engineering Lab   MA – 374   Lab – 3

Name – Rasesh Srivastava

Roll Number – 210123072

Branch – Mathematics and Computing

Question 1:

1. Write a program to determine the initial price of an American call and an American put option in the binomial model with the following data:

$$S(0) = 100; K = 100; T = 1; M = 100; r = 8\%; \sigma = 30\%.$$

Use the following set of u and d for your program:

$$u = e^{\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)\Delta t};\ d = e^{-\sigma\sqrt{\Delta t}+(r-\frac{1}{2}\sigma^2)\Delta t}.$$

Here $\Delta t = \frac{T}{M}$, with M being the number of subintervals in the time interval $[0, T]$. Use the continuous compounding convention in your calculations (i.e., both in $\tilde{p}$ and in the pricing formula).

Now, plot the initial prices of both call and put options by varying one of the parameters at a time (as given below) while keeping the other parameters fixed (as given above) :

(a) $S(0)$.

(b) K.

(c) r.

(d) σ.

(e) M (Do this for three values of K, $K = 95, 100, 105$).

Using binomial model with continuous compounding convention:

The initial price of the American Call Option = 15.736778626185727

The initial price of the American Put Option = 8.923113287677717

```
No arbitrage exists for M = 100
The initial price of the American Call Option = 15.736778626185727
The initial price of the American Put Option = 8.923113287677717
```

No-arbitrage Condition:

The code checks for the arbitrage possibilities using the following conditions necessary for the market to be arbitrage free -

$$d < e^{rt} < u$$

That is, for no arbitrage opportunity to exist, following relations must hold true:

$$d < R < u$$

where,

$$R = e^{rt}$$

$$d = e^{-\sigma\sqrt{t} + \left(r - \frac{\sigma^2}{2}\right)t}$$

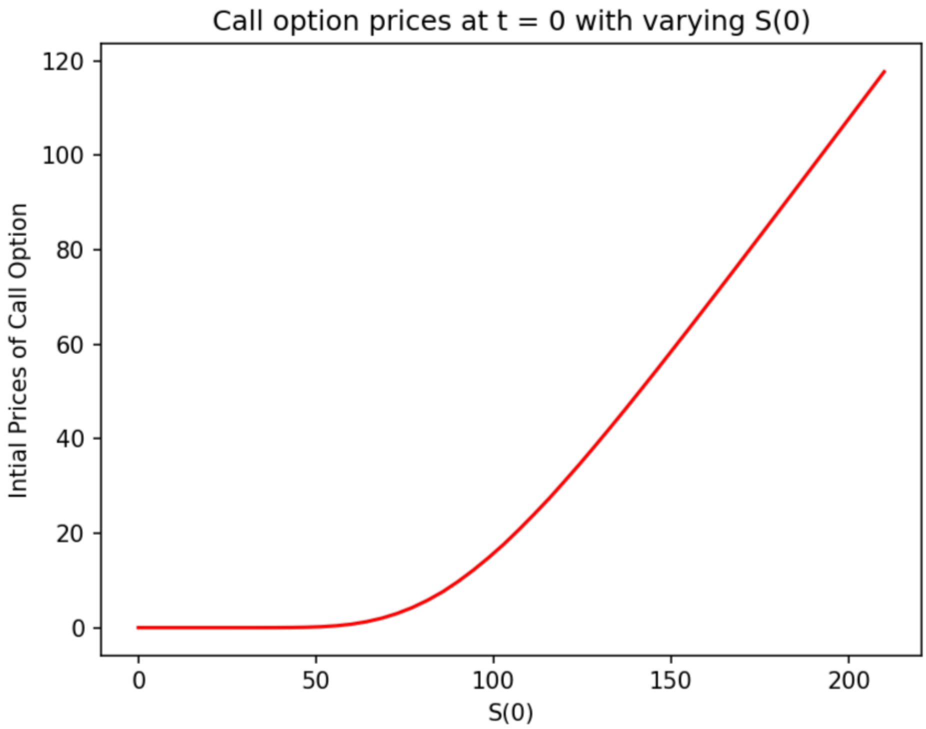
$$u = e^{\sigma\sqrt{t} + \left(r - \frac{\sigma^2}{2}\right)t}$$

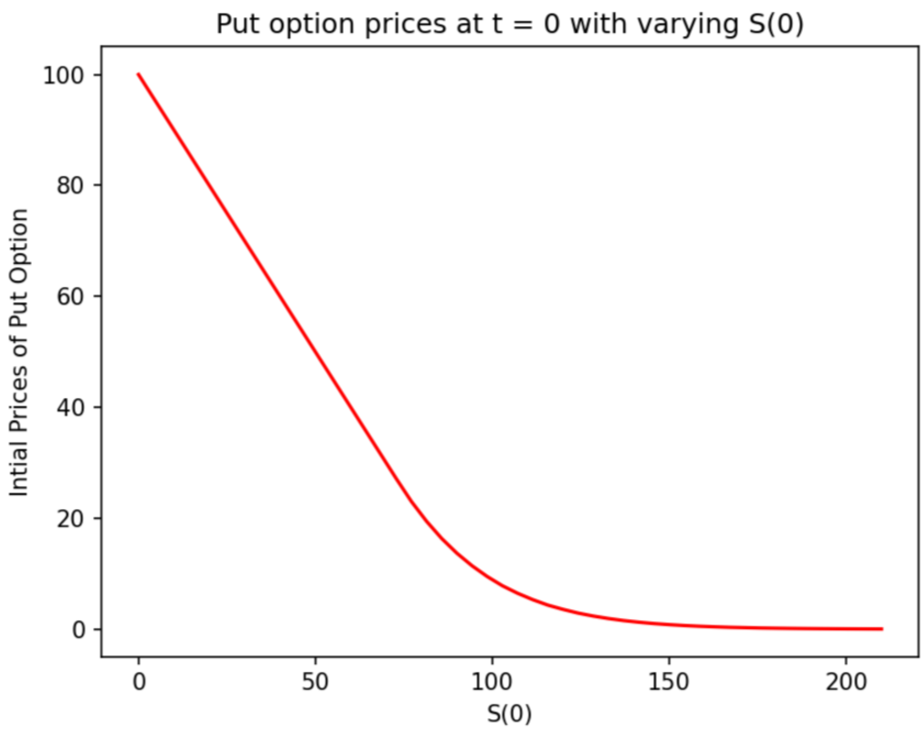
$$t = \frac{T}{M}$$

Since continuous compounding is followed and the final amount in continuous compounding is present value times e^{rt}, so $R = e^{rt}$.

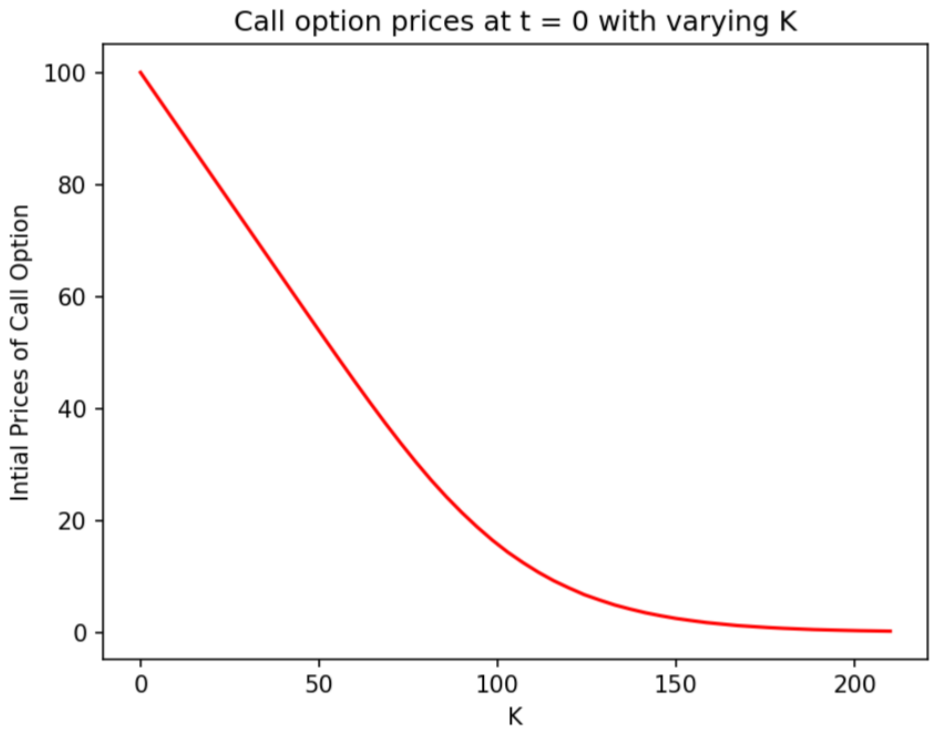
Plotting the initial prices of both call and put options by varying one of the parameters at a time while keeping the other parameters fixed:

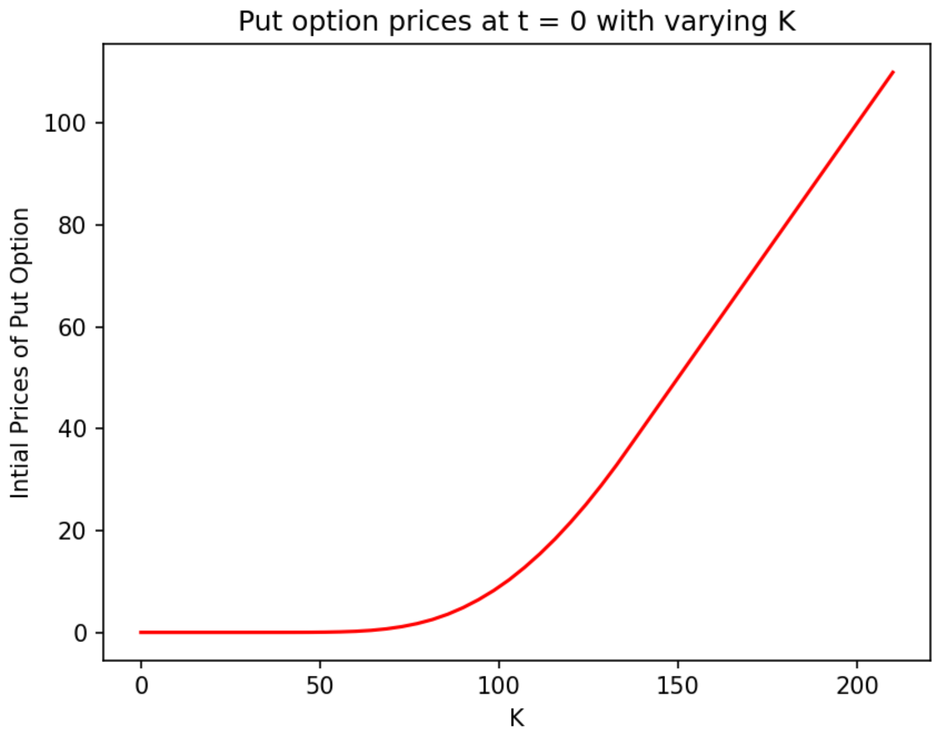
(a)


Call option prices at t = 0 with varying S(0)
Intial Prices of Call Option
S(0)
0
20
40
60
80
100
120
0
50
100
150
200

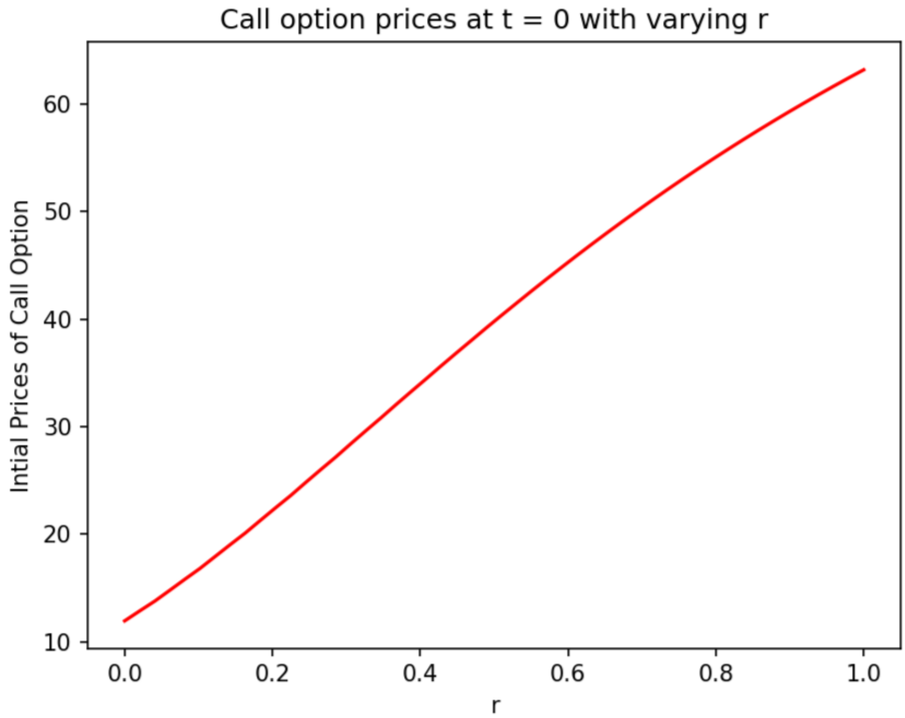

Put option prices at t = 0 with varying S(0)
Intial Prices of Put Option
S(0)
0
20
40
60
80
100
0
50
100
150
200

(b)

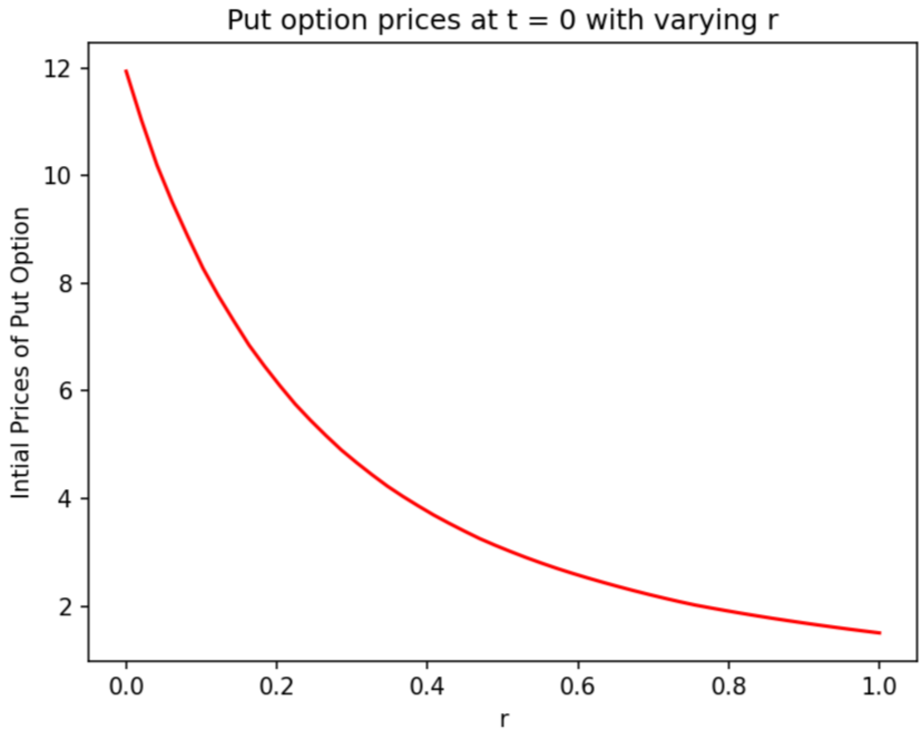

Call option prices at t = 0 with varying K
Intial Prices of Call Option
100
80
60
40
20
0
0
50
100
150
200
K


Put option prices at t = 0 with varying K
Intial Prices of Put Option
100
80
60
40
20
0
0
50
100
150
200
K

(c)

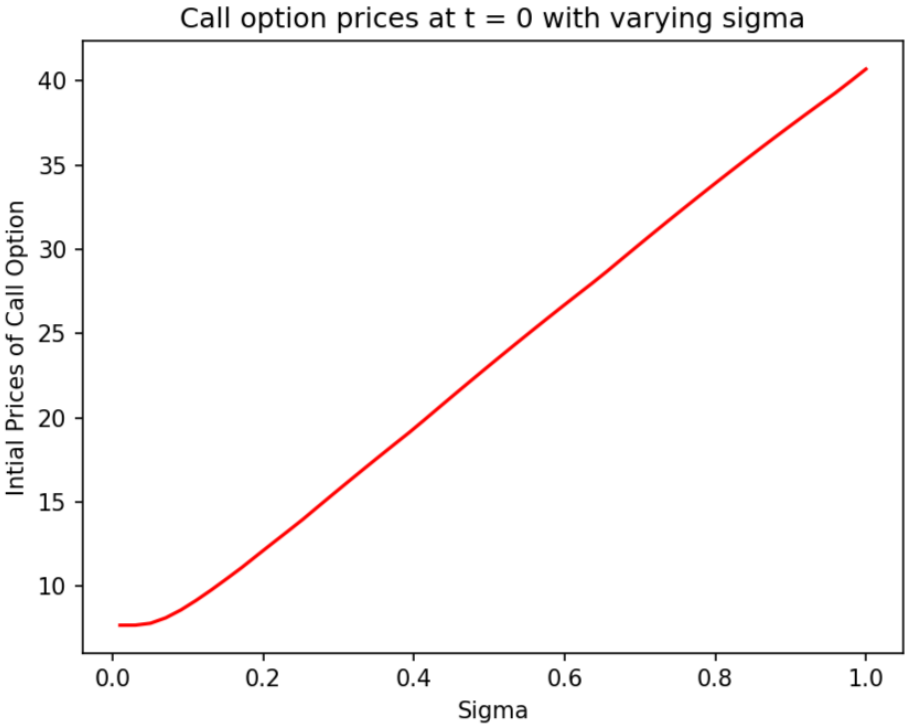


Call option prices at t = 0 with varying r
Intial Prices of Call Option
60
50
40
30
20
10
0.0
0.2
0.4
0.6
0.8
1.0
r

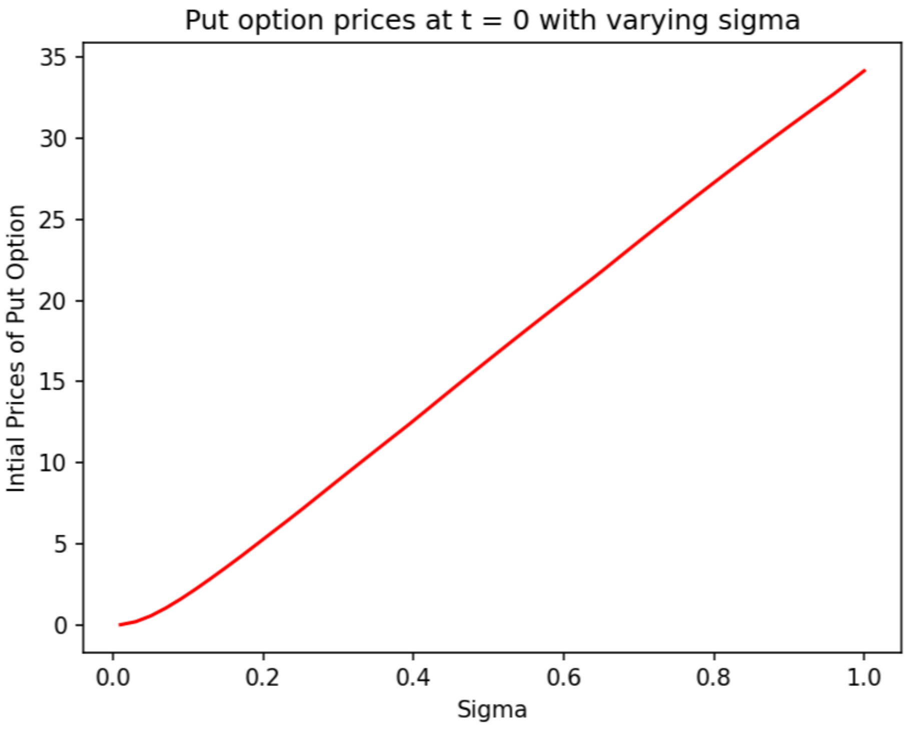


Put option prices at t = 0 with varying r
Intial Prices of Put Option
12
10
8
6
4
2
0.0
0.2
0.4
0.6
0.8
1.0
r

(d)

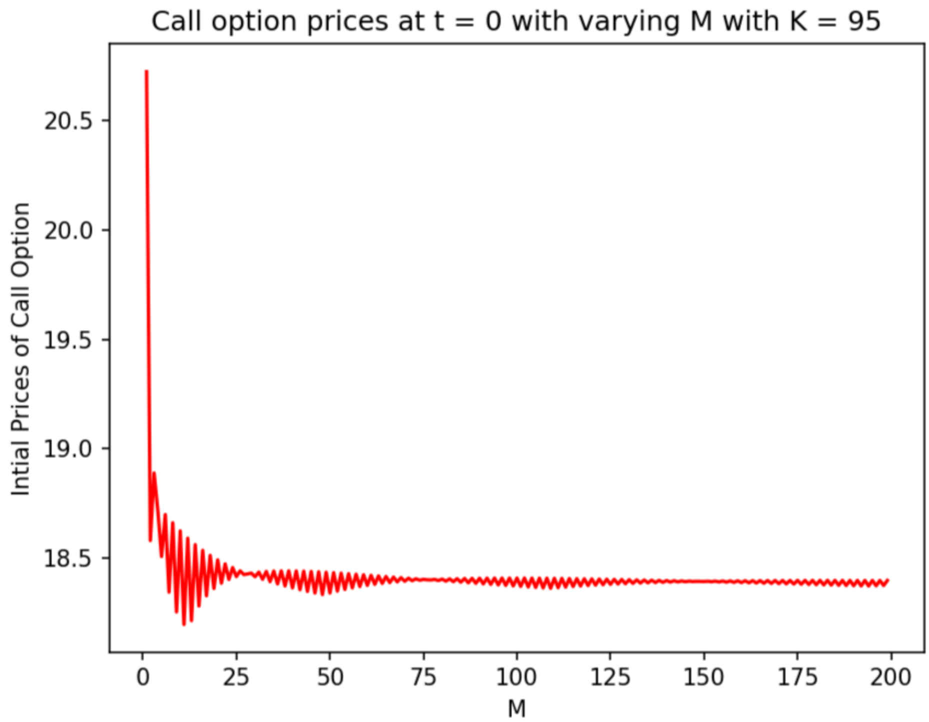


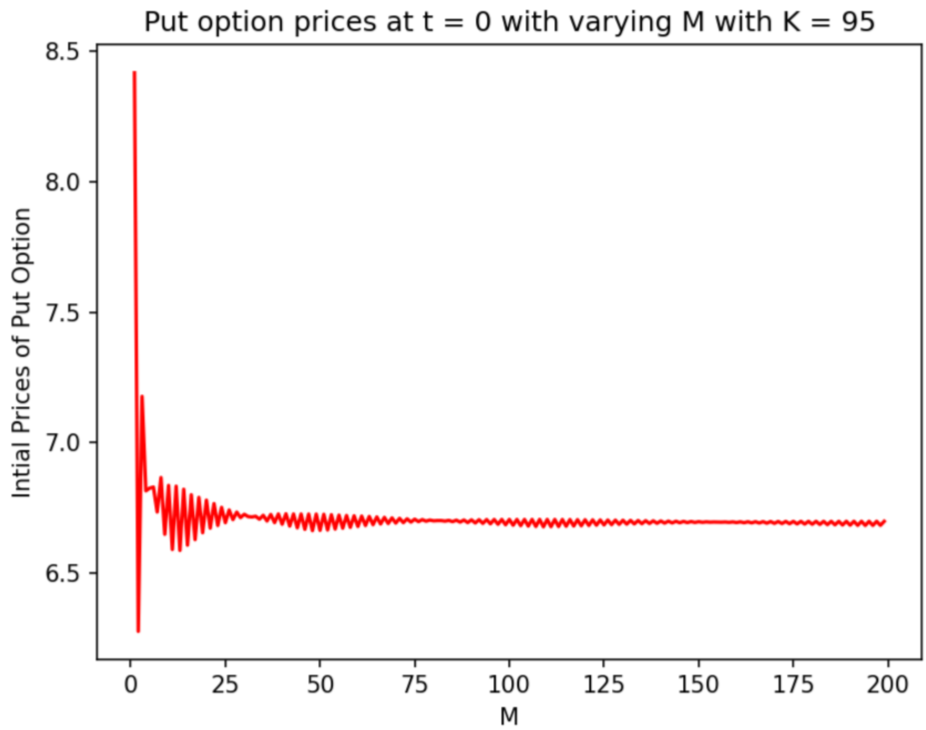
Call option prices at t = 0 with varying sigma
Intial Prices of Call Option
Sigma
0.0
0.2
0.4
0.6
0.8
1.0
10
15
20
25
30
35
40

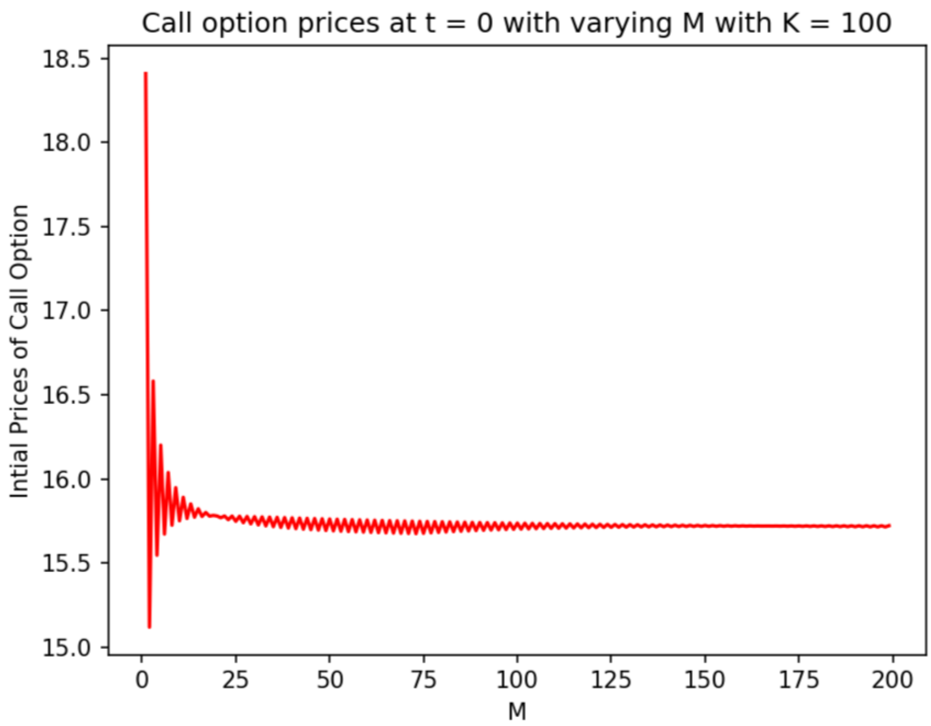


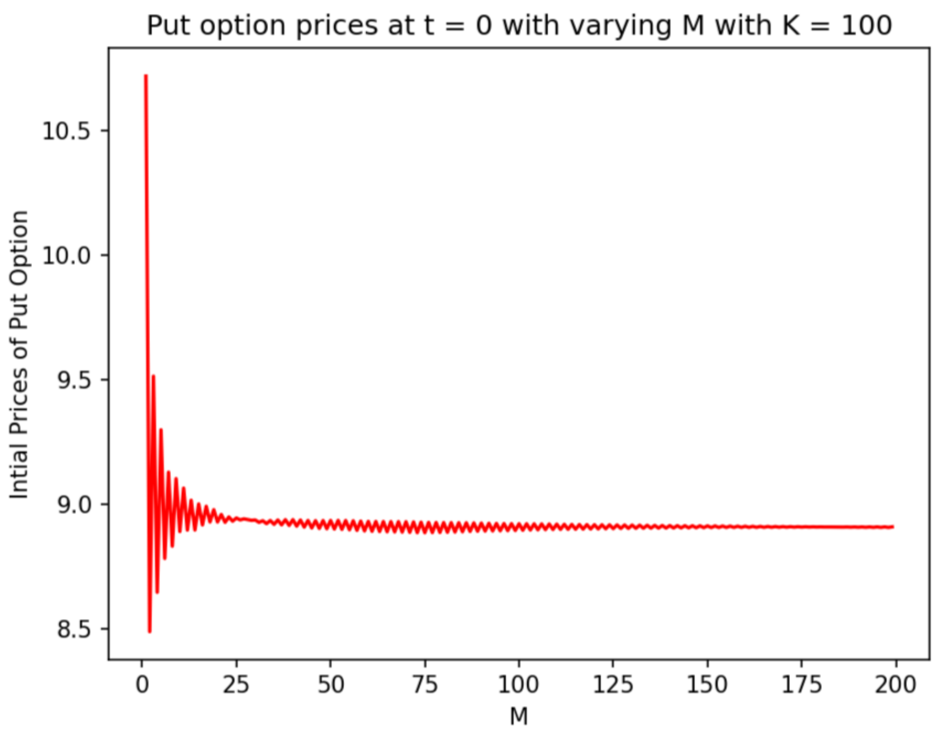
Put option prices at t = 0 with varying sigma
Intial Prices of Put Option
Sigma
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
35

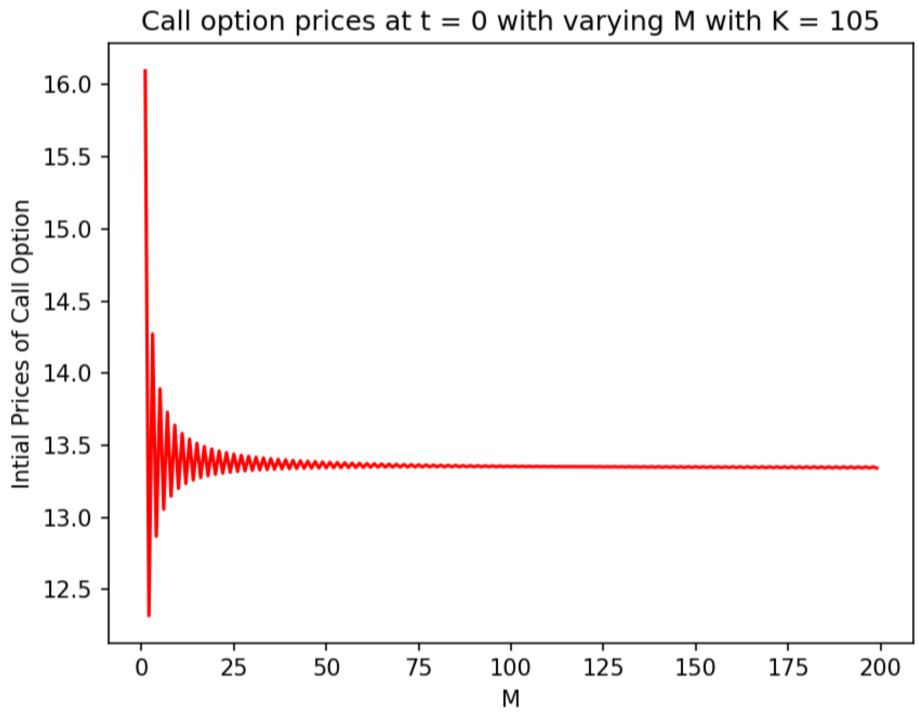
(e)

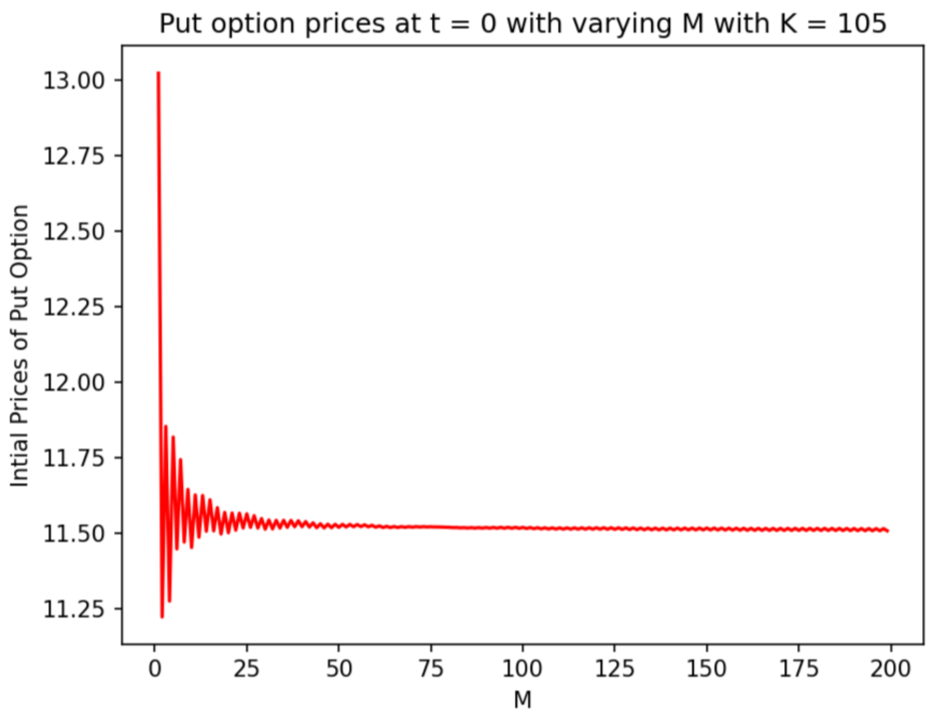

Call option prices at t = 0 with varying M with K = 95
Intial Prices of Call Option
20.5
20.0
19.5
19.0
18.5
0
25
50
75
100
125
150
175
200
M


Put option prices at t = 0 with varying M with K = 95
Intial Prices of Put Option
8.5
8.0
7.5
7.0
6.5
0
25
50
75
100
125
150
175
200
M


Call option prices at t = 0 with varying M with K = 100
Intial Prices of Call Option
M
18.5
18.0
17.5
17.0
16.5
16.0
15.5
15.0
0
25
50
75
100
125
150
175
200


Put option prices at t = 0 with varying M with K = 100
Intial Prices of Put Option
M
10.5
10.0
9.5
9.0
8.5
0
25
50
75
100
125
150
175
200





Observations:

- The price of the Call Option has positive dependence on S(0), and the price of the Put Option price has negative dependence on S(0).
- The price of the Put Option has positive dependence on K, and the price of the Call Option price has negative dependence on K.
- The price of the Call Option has positive dependence on r, and the price of the Put Option price has negative dependence on r.

Question 2:

2. Write a program to determine the initial price of the *lookback* (European) option in the binomial model, using the basic binomial algorithm (used in earlier lab assignments), with the following data:

$$S(0) = 100; T = 1; r = 8\%; \sigma = 30\%.$$

The payoff of the *lookback* option is given by

$$V = \max_{0 \le i \le M} S(i) - S(M),$$

where $S(i) = S(i\Delta t)$ with $\Delta t = \frac{T}{M}$ (M being the number of subintervals of the time interval $[0, T]$). Use the continuous compounding convention in your calculations (i.e., both in $\tilde{p}$ and in the pricing formula). Use the following values of u and d for your program:

$$u = e^{\sigma\sqrt{\Delta t} + (r - \frac{1}{2}\sigma^2)\Delta t}; \qquad d = e^{-\sigma\sqrt{\Delta t} + (r - \frac{1}{2}\sigma^2)\Delta t}$$

(a) Obtain the initial price of the option for $M = 5, 10, 25, 50$.

(b) How do the values of options at time $t = 0$ compare for the above values of M that you have taken ?

(c) Tabulate the values of the options at all intermediate time points for $M = 5$.

(a)

The initial prices of the lookback (European) option in the binomial model, using the basic binomial algorithm for different values of M are as follows:

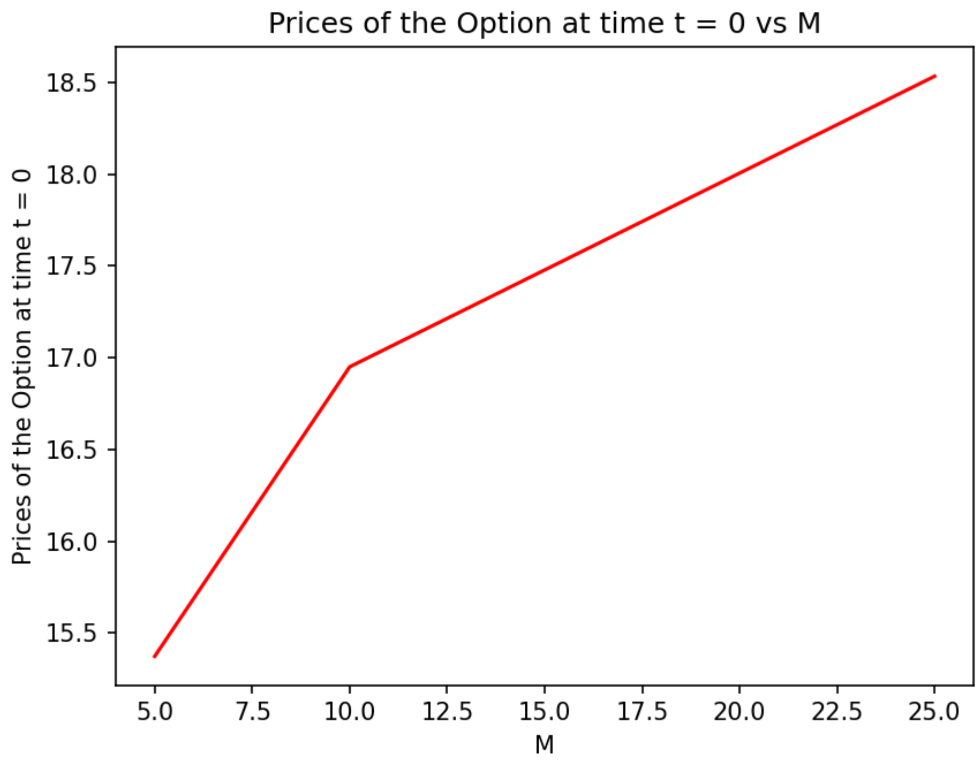
Serial Number	M	Initial Price of Lookback European Option	Execution Time in seconds
1	5	15.372952215663778	0.0011849403381347656
2	10	16.95034049177767	0.0050394535064697266
3	25	18.533781500094165	288.1385660171509
4	26	18.590003349885308	561.5937082767487
5	27	18.6418127619738	1192.9899275302887
6	50	Could not compute	Could not compute

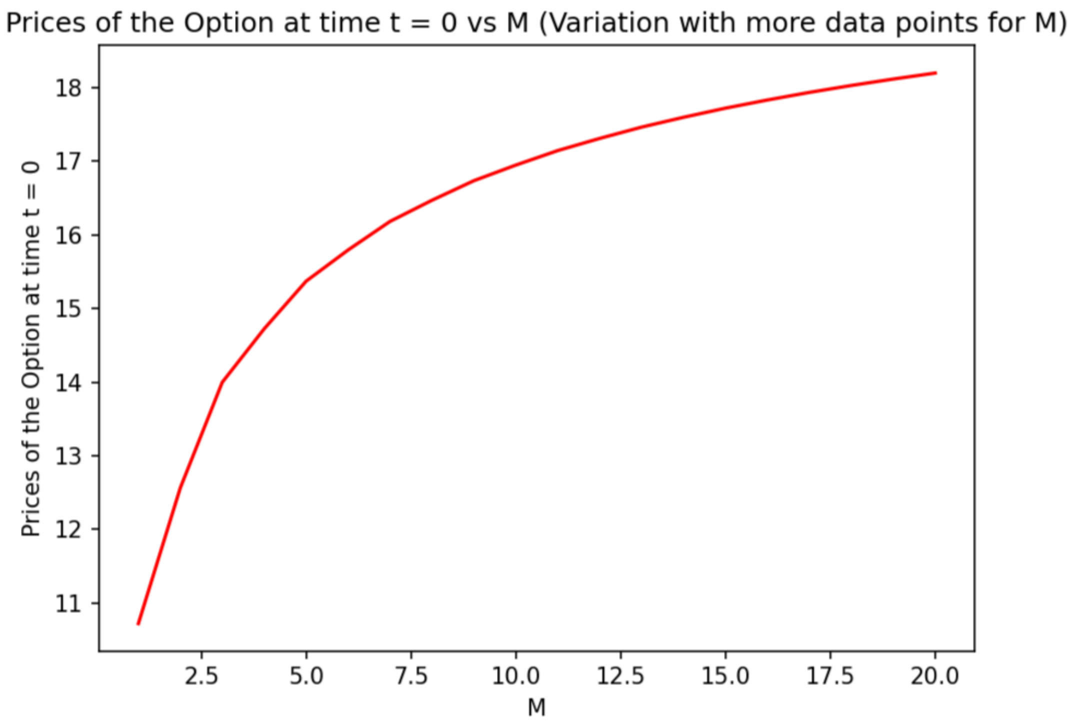
Observations:

For M = 50, the written algorithm is not able to calculate the option price in reasonably small time because the algorithm used has exponential time complexity. For M = 26 and 27, we can see that as we increase M by 1, the execution time almost doubles. So, computation for M = 50 is infeasible with this algorithm for finding the initial price of the European loopback option.

(b)

Comparison of the values of options at time t = 0 for the values of M that I have taken is done using the following plots:





Observations:

As the magnitude of M rises, there is a corresponding increase in the initial option price, and it becomes apparent that the prices gradually approach convergence with further increments in M.

(c) Here, t depicts the time intervals with respect to M

The prices of the options at all intermediate time points for M = 5 are as follows:

```
Question 2 Part (c)
At time t = 0
Index Number = 0        Option Price = 15.372952215663778

At time t = 1
Index Number = 0        Option Price = 15.532131468492956
Index Number = 1        Option Price = 15.709699760878111

At time t = 2
Index Number = 0        Option Price = 15.199750099616727
Index Number = 1        Option Price = 16.365773501799975
Index Number = 2        Option Price = 11.62259245758552
Index Number = 3        Option Price = 20.30531014128848

At time t = 3
Index Number = 0        Option Price = 13.386169289151374
Index Number = 1        Option Price = 17.50446467389843
Index Number = 2        Option Price = 10.235825536366997
Index Number = 3        Option Price = 23.026215406441317
Index Number = 4        Option Price = 10.235825536367
Index Number = 5        Option Price = 13.384908157013323
Index Number = 6        Option Price = 12.702323203700722
Index Number = 7        Option Price = 28.566489442465258

At time t = 4
Index Number = 0        Option Price = 10.33248062285694
Index Number = 1        Option Price = 16.872978416162187
Index Number = 2        Option Price = 7.900801695311674
Index Number = 3        Option Price = 27.676760285887045
Index Number = 4        Option Price = 7.900801695311674
Index Number = 5        Option Price = 12.902037888217311
Index Number = 6        Option Price = 12.103285439254641
Index Number = 7        Option Price = 34.69646280474455
Index Number = 8        Option Price = 7.900801695311674
Index Number = 9        Option Price = 12.902037888217318
Index Number = 10       Option Price = 6.041401838252844
Index Number = 11       Option Price = 21.163223292550345
Index Number = 12       Option Price = 6.041401838252844
Index Number = 13       Option Price = 19.775755431345573
Index Number = 14       Option Price = 19.77575543134555
Index Number = 15       Option Price = 38.28243217635733
```

```
At time t = 5
Index Number = 0        Option Price = 0.0
Index Number = 1        Option Price = 21.002491662264447
Index Number = 2        Option Price = 0.0
Index Number = 3        Option Price = 34.29714522948986
Index Number = 4        Option Price = 0.0
Index Number = 5        Option Price = 16.05969832296735
Index Number = 6        Option Price = 14.189941164644068
Index Number = 7        Option Price = 42.06197481701972
Index Number = 8        Option Price = 0.0
Index Number = 9        Option Price = 16.05969832296735
Index Number = 10       Option Price = 0.0
Index Number = 11       Option Price = 26.225545739139193
Index Number = 12       Option Price = 0.0
Index Number = 13       Option Price = 24.601948051238253
Index Number = 14       Option Price = 24.601948051238267
Index Number = 15       Option Price = 45.914488453717624
Index Number = 16       Option Price = 0.0
Index Number = 17       Option Price = 16.05969832296735
Index Number = 18       Option Price = 0.0
Index Number = 19       Option Price = 26.225545739139207
Index Number = 20       Option Price = 0.0
Index Number = 21       Option Price = 12.280157724719814
Index Number = 22       Option Price = 10.850435176426544
Index Number = 23       Option Price = 32.162975578905915
Index Number = 24       Option Price = 0.0
Index Number = 25       Option Price = 12.280157724719814
Index Number = 26       Option Price = 9.440589282577335
Index Number = 27       Option Price = 30.75312968505669
Index Number = 28       Option Price = 9.440589282577307
Index Number = 29       Option Price = 30.753129685056678
Index Number = 30       Option Price = 30.753129685056678
Index Number = 31       Option Price = 47.04990888934698
```

Question 3:

3. Repeat Problem 2 using a (Markov based) computationally efficient binomial algorithm. Make a comparative analysis of the two algorithms, like computational time, the value of M it can handle, etc.

(a)

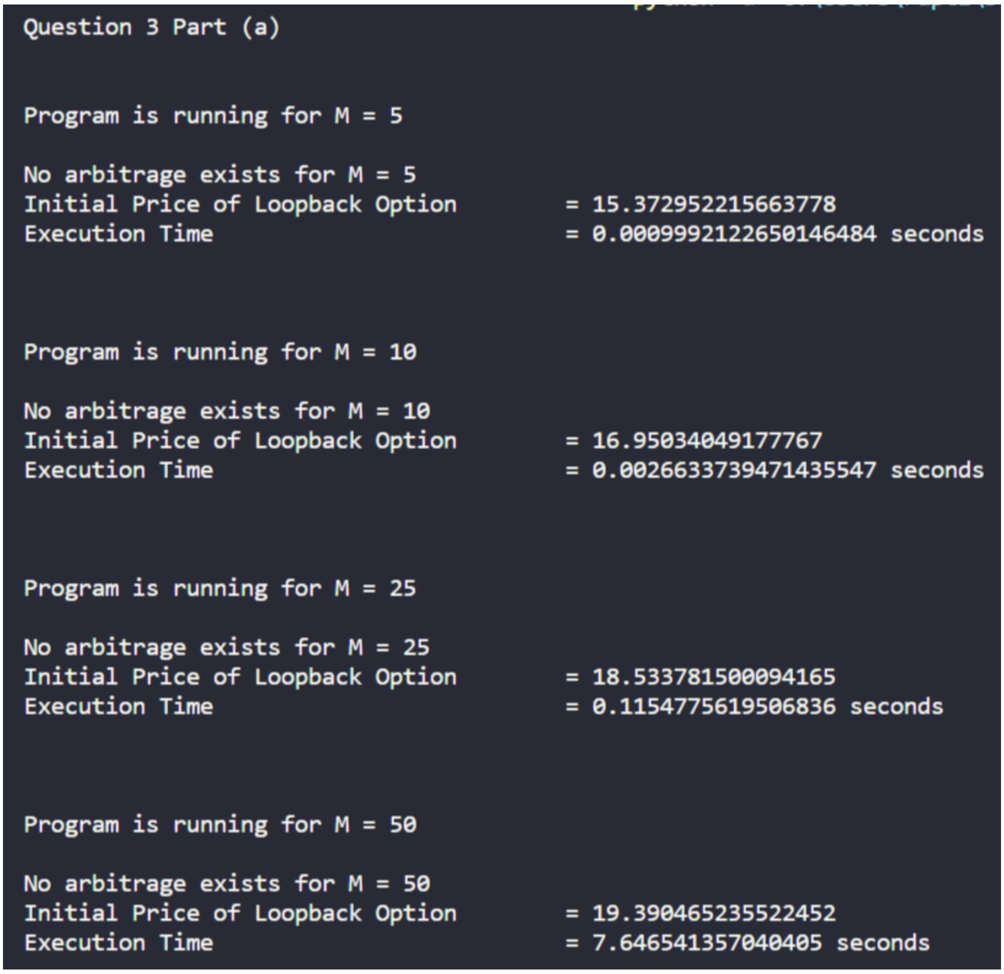
Using Markov property, we derive the following recurrence relation for finding out the initial price of the lookback (European) option:

$$v_n(s, m) = \frac{(1-p)v_{n+1}(d.s, m) + pv_{n+1}(u.s,\ max(u.s,\ m))}{R}$$

The initial prices of the lookback (European) option in the binomial model, using a (Markov based) computationally efficient binomial algorithm for different values of M are as follows:

Serial Number	M	Initial Price of Lookback European Option	Execution Time in seconds

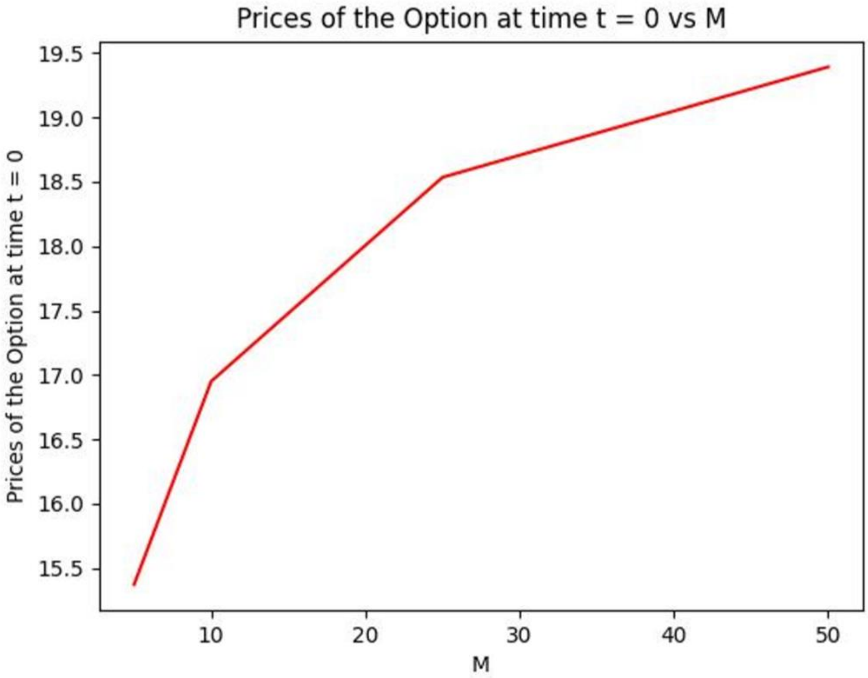
1	5	15.372952215663778	0.0009992122650146484
2	10	16.95034049177767	0.0026633739471435547
3	25	18.533781500094165	0.1154775619506836
4	50	19.390465235522452	7.646541357040405

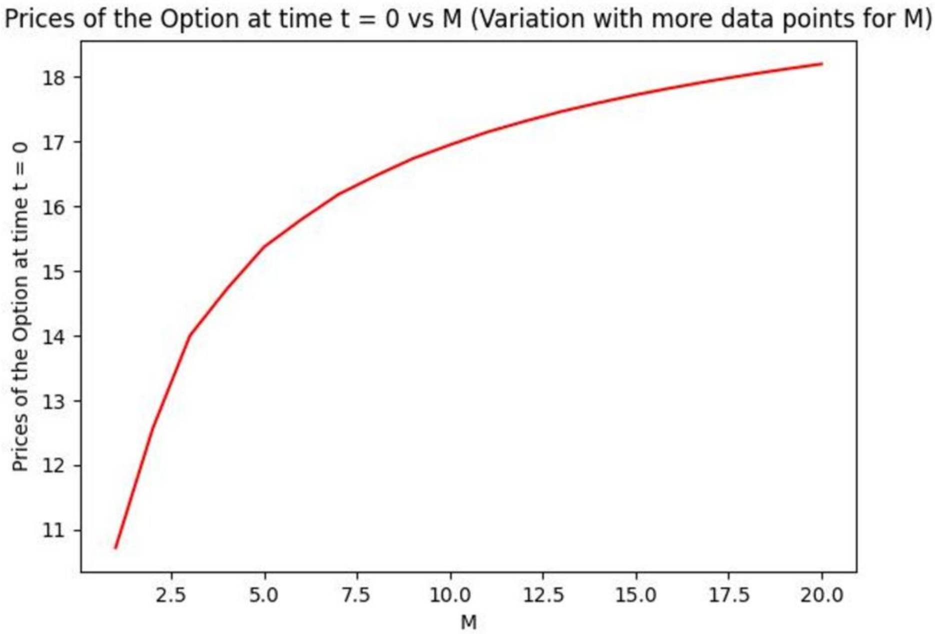


```
Question 3 Part (a)


Program is running for M = 5

No arbitrage exists for M = 5
Initial Price of Loopback Option        = 15.372952215663778
Execution Time                          = 0.0009992122650146484 seconds



Program is running for M = 10

No arbitrage exists for M = 10
Initial Price of Loopback Option        = 16.95034049177767
Execution Time                          = 0.0026633739471435547 seconds



Program is running for M = 25

No arbitrage exists for M = 25
Initial Price of Loopback Option        = 18.533781500094165
Execution Time                          = 0.1154775619506836 seconds



Program is running for M = 50

No arbitrage exists for M = 50
Initial Price of Loopback Option        = 19.390465235522452
Execution Time                          = 7.646541357040405 seconds
```

(b)

Comparison of the values of options at time t = 0 for the values of M that I have taken is done using the following plots:





Observations:

As the magnitude of M rises, there is a corresponding increase in the initial option price, and it becomes apparent that the prices gradually approach convergence with further increments in M.

(c)

Here, t depicts the time intervals with respect to M

The prices of the options at all intermediate time points for M = 5 are as follows:

Note: Each state is defined by a tuple representing:

(i) the current stock price at that moment

(ii) the highest stock price encountered along the entire path up to the current state.

```
Question 3 Part (c)
At time t = 0
Intermediate state = (100, 100)        Option Price = 15.372952215663778

At time t = 1
Intermediate state = (115.16135876866093, 115.16135876866093)          Option Price = 15.532131468492956
Intermediate state = (88.05891748599798, 100)          Option Price = 15.709699760878111

At time t = 2
Intermediate state = (132.6213855344424, 132.6213855344424)            Option Price = 15.199750099616727
Intermediate state = (101.40984589384922, 115.16135876866093)          Option Price = 16.365773501799975
Intermediate state = (101.40984589384924, 101.40984589384924)          Option Price = 11.62259245758552
Intermediate state = (77.543729488058, 100)            Option Price = 20.30531014128848

At time t = 3
Intermediate state = (152.7285895992882, 152.7285895992882)            Option Price = 13.386169289151374
Intermediate state = (116.78495645656189, 132.6213855344424)           Option Price = 17.50446467389843
Intermediate state = (116.78495645656187, 116.78495645656187)          Option Price = 10.235825536366997
Intermediate state = (89.3004125183424, 115.16135876866093)            Option Price = 23.026215406441317
Intermediate state = (116.78495645656189, 116.78495645656189)          Option Price = 10.235825536367
Intermediate state = (89.30041251834241, 101.40984589384924)           Option Price = 13.384908157013323
Intermediate state = (89.3004125183424, 100)           Option Price = 12.70232320370072
Intermediate state = (68.28416876545448, 100)          Option Price = 28.566489442465258

At time t = 4
Intermediate state = (175.88431901075205, 175.88431901075205)          Option Price = 10.33248062285694
Intermediate state = (134.4911426927657, 152.7285895992882)            Option Price = 16.872978416162187
Intermediate state = (134.4911426927657, 134.4911426927657)            Option Price = 7.90080169531167
Intermediate state = (102.8395684421425, 132.6213855344424)            Option Price = 27.676760285887045
Intermediate state = (134.49114269276566, 134.49114269276566)          Option Price = 7.90080169531167
Intermediate state = (102.8395684421425, 116.78495645656187)           Option Price = 12.902037888217311
Intermediate state = (102.8395684421425, 115.16135876866093)           Option Price = 12.103285439254641
Intermediate state = (78.63697657418294, 115.16135876866093)           Option Price = 34.69646280474455
Intermediate state = (102.8395684421425, 116.78495645656189)           Option Price = 12.902037888217318
Intermediate state = (102.83956844214251, 102.83956844214251)          Option Price = 6.04140183825284
Intermediate state = (78.63697657418295, 101.40984589384924)           Option Price = 21.163223292550345
Intermediate state = (102.8395684421425, 102.8395684421425)            Option Price = 6.04140183825284
Intermediate state = (78.63697657418294, 100)          Option Price = 19.775755431345573
Intermediate state = (78.63697657418295, 100)          Option Price = 19.77575543134555
Intermediate state = (60.130299829171165, 100)         Option Price = 38.2824321763573
```

```
At time t = 5
Intermediate state = (202.5507716337883, 202.5507716337883)          Option Price = 0.0
Intermediate state = (154.8818273484876, 175.88431901075205)         Option Price = 21.002491662264447
Intermediate state = (154.8818273484876, 154.8818273484876)          Option Price = 0.0
Intermediate state = (118.43144436979834, 152.7285895992882)         Option Price = 34.29714522948986
Intermediate state = (118.43144436979834, 134.4911426927657)         Option Price = 16.05969832296735
Intermediate state = (118.43144436979833, 132.6213855344424)         Option Price = 14.18994116464068
Intermediate state = (90.5594107174226 8, 132.6213855344424)          Option Price = 42.06197481701972
Intermediate state = (154.8818273484875 8, 154.8818273484875 8)        Option Price = 0.0
Intermediate state = (118.43144436979831, 134.49114269276566)        Option Price = 16.05969832296735
Intermediate state = (118.43144436979833, 118.43144436979833)        Option Price = 0.0
Intermediate state = (90.55941071742268, 116.78495645656187)         Option Price = 26.225545739139193
Intermediate state = (90.55941071742268, 115.16135876866093)         Option Price = 24.601948051238253
Intermediate state = (90.55941071742267, 115.16135876866093)         Option Price = 24.601948051238267
Intermediate state = (69.24687031494331, 115.16135876866093)         Option Price = 45.914488453717624
Intermediate state = (90.55941071742268, 116.78495645656189)         Option Price = 26.225545739139207
Intermediate state = (118.43144436979834, 118.43144436979834)        Option Price = 0.0
Intermediate state = (90.5594107174227, 102.83956844214251)          Option Price = 12.280157724719814
Intermediate state = (90.5594107174227, 101.40984589384924)          Option Price = 10.850435176426544
Intermediate state = (69.24687031494332, 101.40984589384924)         Option Price = 32.162975578905915
Intermediate state = (90.55941071742268, 102.8395684421425)          Option Price = 12.280157724719814
Intermediate state = (90.55941071742267, 100)          Option Price = 9.440589282577335
Intermediate state = (69.24687031494331, 100)          Option Price = 30.75312968505669
Intermediate state = (90.5594107174227, 100)           Option Price = 9.440589282577307
Intermediate state = (69.24687031494332, 100)          Option Price = 30.753129685056678
Intermediate state = (52.95009111065302, 100)          Option Price = 47.04990888934698
```

Comparative Analysis of the two algorithms:

- The unoptimized algorithm exhibits exponential space complexity, whereas the Markov-based optimized algorithm does not share this characteristic. This distinction arises from the incorporation of tabulation and memoization for temporary storage, leveraging dynamic programming principles.
- The unoptimized algorithm proves impractical as its time complexity can escalate dramatically for small values of M, such as 50. In contrast, the Markov-based algorithm demonstrates resilience, effectively handling such scenarios.
- The following table compares the execution time (in seconds) of the two algorithms for different values of M:

M	Unoptimized Algorithm	Optimized Markov based Algorithm
5	0.0011849403381347656	0.0009992122650146484
10	0.005039453506469726 6	0.0026633739471435547
25	288.1385660171509	0.1154775619506836
50	Infeasible	7.646541357040405

- In the unoptimized algorithm, both the time and space complexity are exponential. So, it can handle maximum M of around 30 on most processors/RAM since after that, the RAM wouldn't have such huge computing power. After M = 25, the execution time almost doubles as M is increased by 1.

- In the optimized Markov based algorithm, we use memoization concept of dynamic programming which reduces both the space and time complexities of the unoptimized algorithm. So, this algorithm can handle larger values of M at least upto 50, as I have run the code till M = 50 for the previous questions.

Question 4:

4. As in Problem 3, use a (Markov based) computationally efficient binomial algorithm to price an European call option. Make a comparative analysis of the two algorithms, like computational time, the value of M it can handle, etc.

(a)

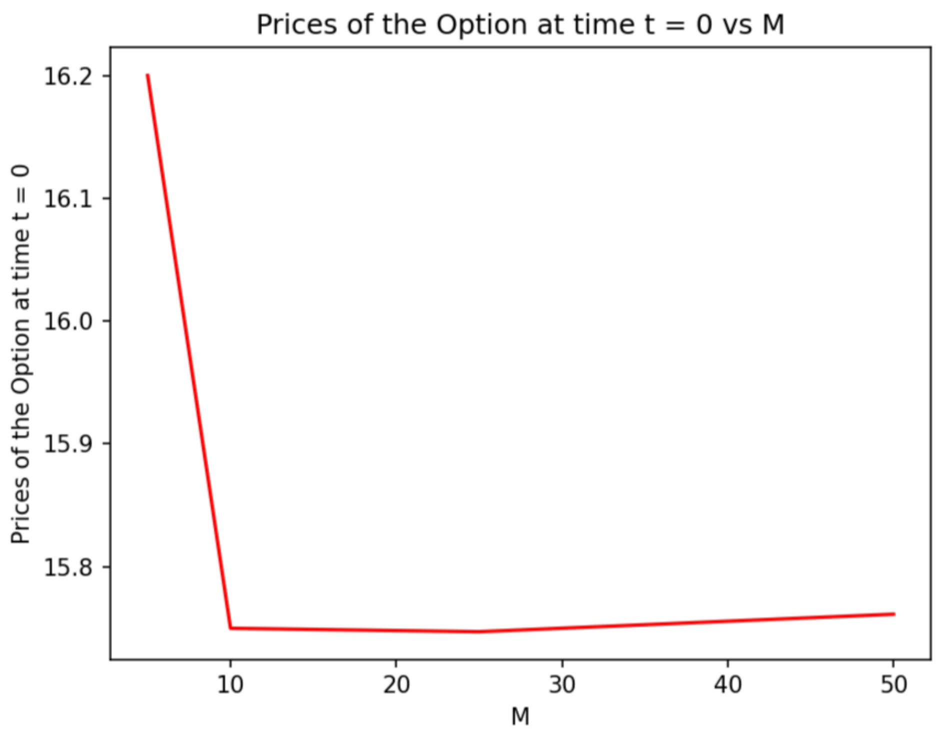
Using binomial model with continuous compounding convention and K=100:

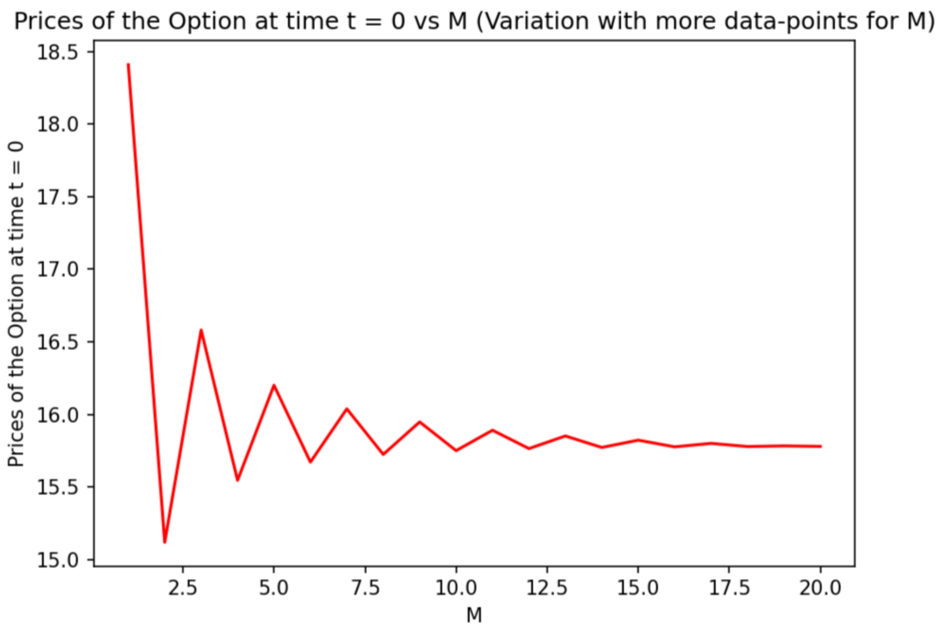
The prices of the European Call Option at time t = 0 for different values of M are as follows:

M	**European Call Option Price at t = 0**	**Execution Time in seconds**
5	16.200135785709463	0.000099945068359375
10	15.749706920472502	0.000575362843355387
25	15.746918255600455	0.000826182123710216
50	15.761196879829429	0.001001596450805664

(b)

Comparison of the values of options at time t = 0 for the values of M that I have taken is done using the following plots:





Observations:

The initial price of the European Call Option converges approximately around 15.75 as M increases.

(c) Here, t depicts the time intervals with respect to M

The prices of the options at all intermediate time points for M = 5 are as follows:

```
Question 4 Part (c)
At t = 0
Index Number = 0        Option Price = 16.200135785709463

At t = 1
Index Number = 0        Option Price = 25.375255893366354
Index Number = 1        Option Price = 7.543996674048174

At t = 2
Index Number = 0        Option Price = 38.432095157524756
Index Number = 1        Option Price = 13.131857964608423
Index Number = 2        Option Price = 2.1972816917220994

At t = 3
Index Number = 0        Option Price = 55.877931391368456
Index Number = 1        Option Price = 22.219195424615478
Index Number = 2        Option Price = 4.464542360415781
Index Number = 3        Option Price = 0.0

At t = 4
Index Number = 0        Option Price = 77.47158700522355
Index Number = 1        Option Price = 36.078410687237174
Index Number = 2        Option Price = 9.071271363629885
Index Number = 3        Option Price = 0.0
Index Number = 4        Option Price = 0.0

At t = 5
Index Number = 0        Option Price = 102.55077163378829
Index Number = 1        Option Price = 54.881827348487604
Index Number = 2        Option Price = 18.431444369798328
Index Number = 3        Option Price = 0
Index Number = 4        Option Price = 0
Index Number = 5        Option Price = 0
```

Comparative Analysis of the algorithms:

- The execution times of all the algorithms are compared below:

```
Unoptimised Binomial Algorithm to price an European Call Option is running
No arbitrage exists for M = 5
Price of European Call Option             = 16.200135785709463
Execution Time                        = 0.0009987354278564453 seconds

No arbitrage exists for M = 10
Price of European Call Option             = 15.749706920472505
Execution Time                        = 0.0010008811950683594 seconds

No arbitrage exists for M = 25
Price of European Call Option             = 15.746918255600471
Execution Time                        = 44.5668089389801 seconds
```

```
Efficient Binomial Algorithm to price an European Call Option is running (Markov Based)
No arbitrage exists for M = 5
Price European Call Option             = 16.200135785709463
Execution Time                     = 0.0 seconds

No arbitrage exists for M = 10
Price European Call Option             = 15.749706920472503
Execution Time                     = 0.0010001659393310547 seconds

No arbitrage exists for M = 25
Price European Call Option             = 15.746918255600457
Execution Time                     = 0.0 seconds

No arbitrage exists for M = 50
Price European Call Option             = 15.761196879829438
Execution Time                     = 0.0010001659393310547 seconds
```

```
Most Efficient Binomial Algorithm to price an European Call Option is running (Markov Based)
No arbitrage exists for M = 5
Price of European Call Option            = 16.200135785709463
Execution Time                      = 0.00099945068359375 seconds

No arbitrage exists for M = 10
Price of European Call Option            = 15.749706920472502
Execution Time                      = 0.0 seconds

No arbitrage exists for M = 25
Price of European Call Option            = 15.746918255600455
Execution Time                      = 0.0 seconds

No arbitrage exists for M = 50
Price of European Call Option            = 15.761196879829429
Execution Time                      = 0.001001596450805664 seconds
```

(Note: The execution times change when the code is run again and again and on different machines. This is just one screenshot of one instance of execution times)

- The unoptimized algorithm exhibits both exponential time and space complexity. On the other hand, the efficient algorithm demonstrates quadratic time and space complexity in terms of M, although the space complexity can be reduced to linear. However, considering the additional requirement to print intermediate information, I implemented an algorithm with quadratic complexity. The most efficient algorithm achieves nearly linear time complexity, especially when leveraging memoization for calculating nCr, and also maintains linear space complexity.

- The maximum values of M that these algorithms can handle are as follows:

(a) Unoptimized Algorithm: Around 30, as explained in the previous question, question number 3, of this assignment.

(b) Optimized and most Optimized Algorithms: They worked perfectly fine with M = 50 and M =100 also, which was the largest value of M mentioned in the Assignment 3 question paper. So, these algorithms can work correctly and in reasonably small time for fairly large values of M.