MA 374 Financial Engineering Lab Lab 03

Karan Garg - 210123076

Q1)

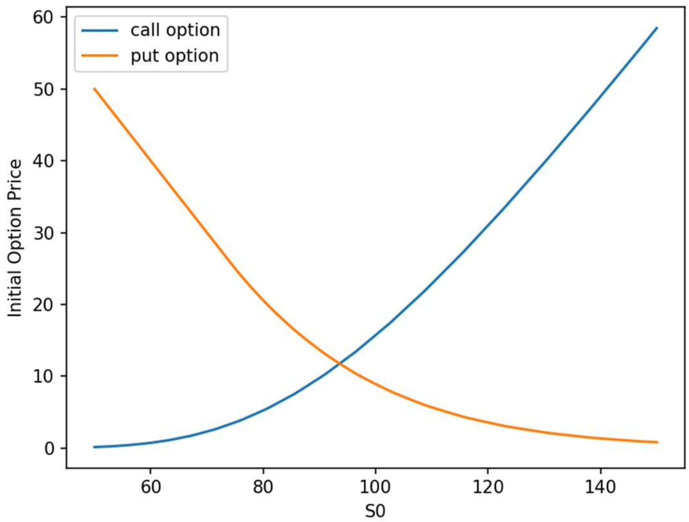
$S(0) = 100; K = 100; T = 1; M = 100; r = 8\%; \sigma = 30\%.$

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

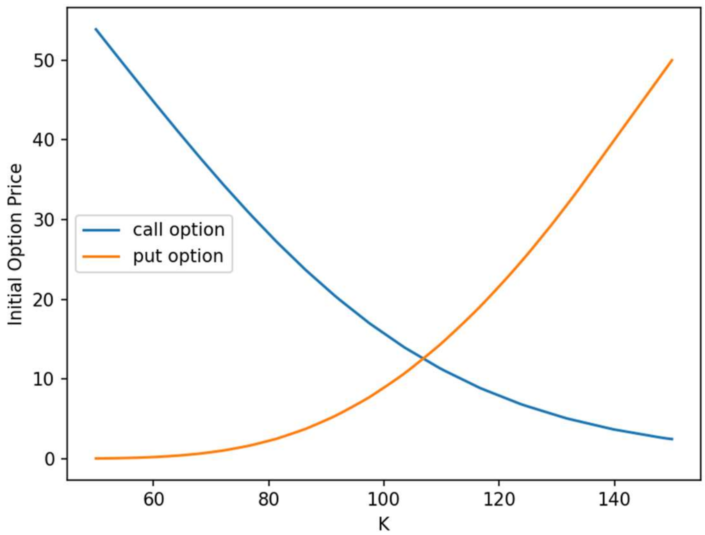
initial call option value: **15.7367**

initial put option value: **8.92311**

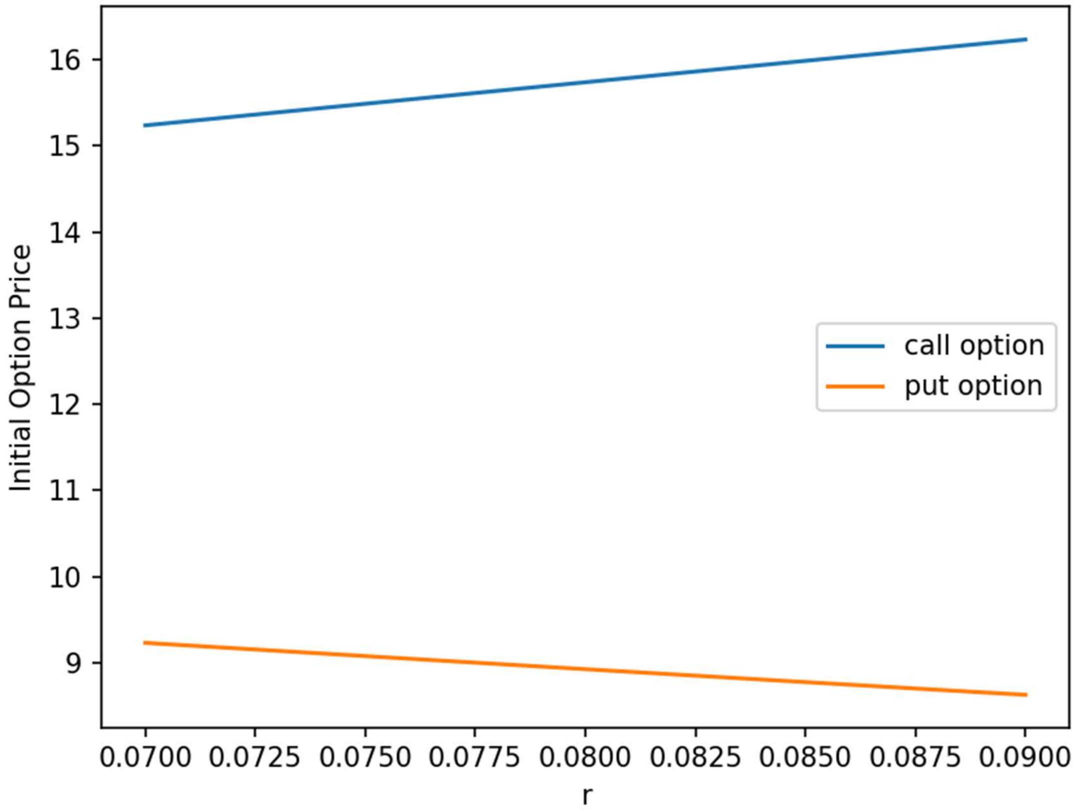
1-varying S0:



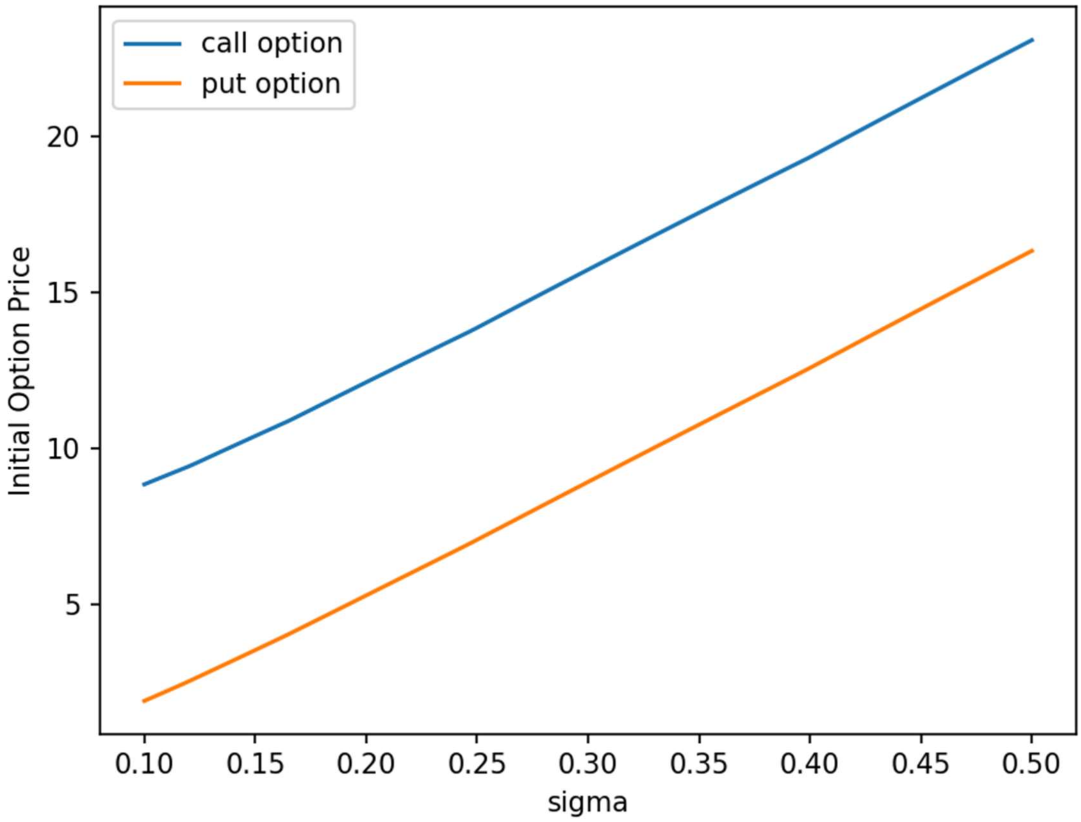
2- varying K



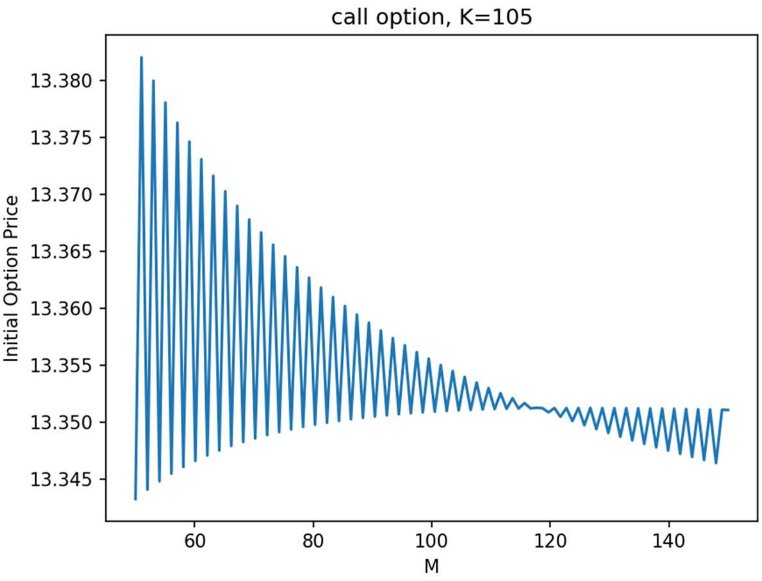
3- varying r

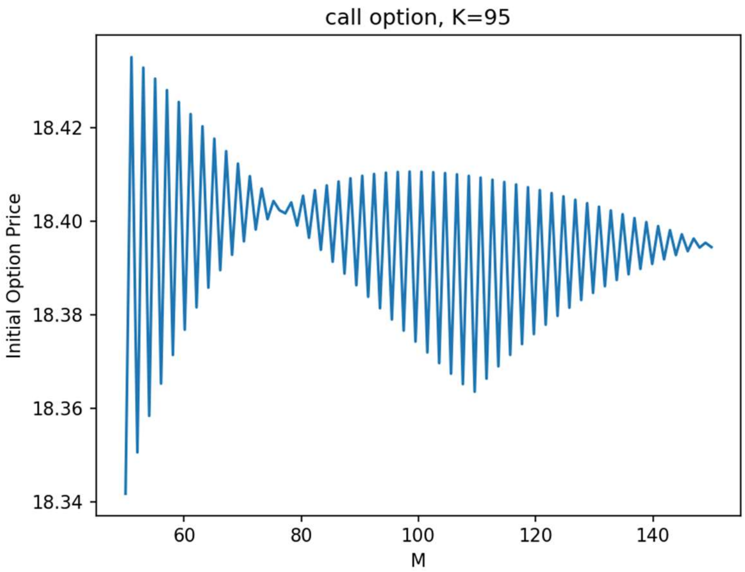


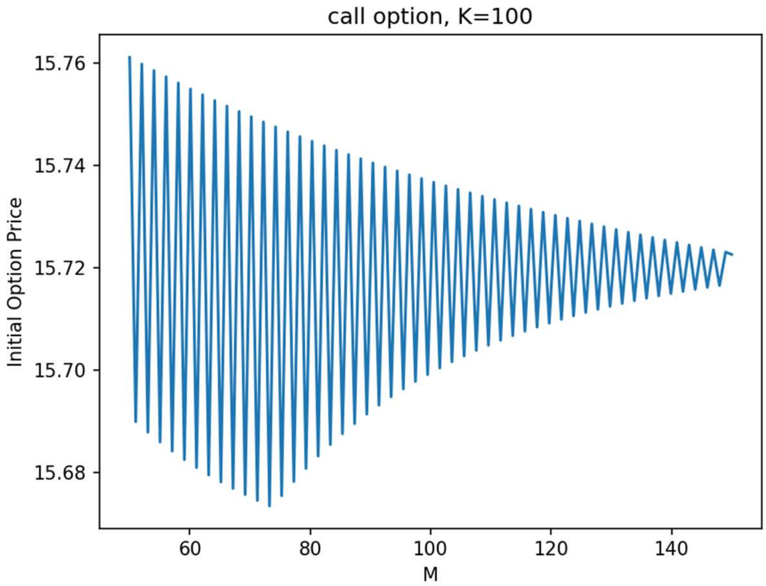
4-varying sigma

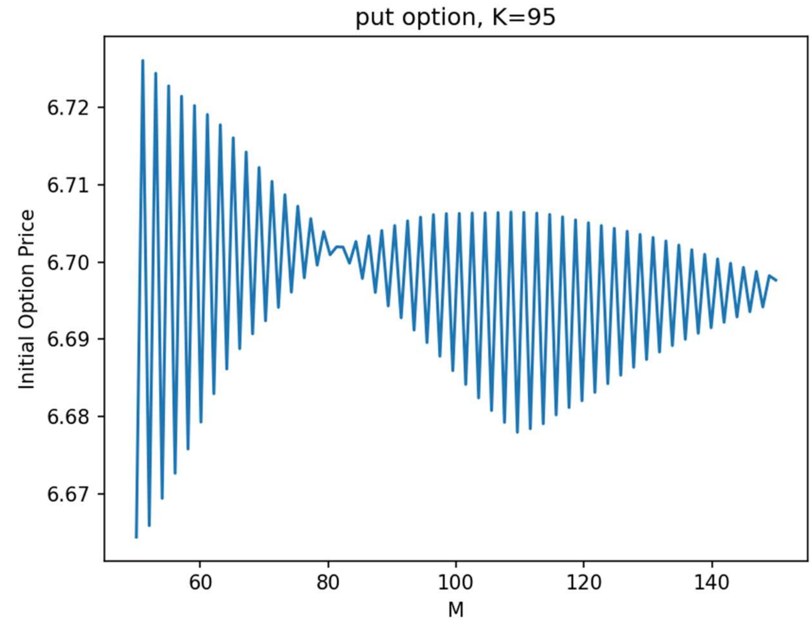


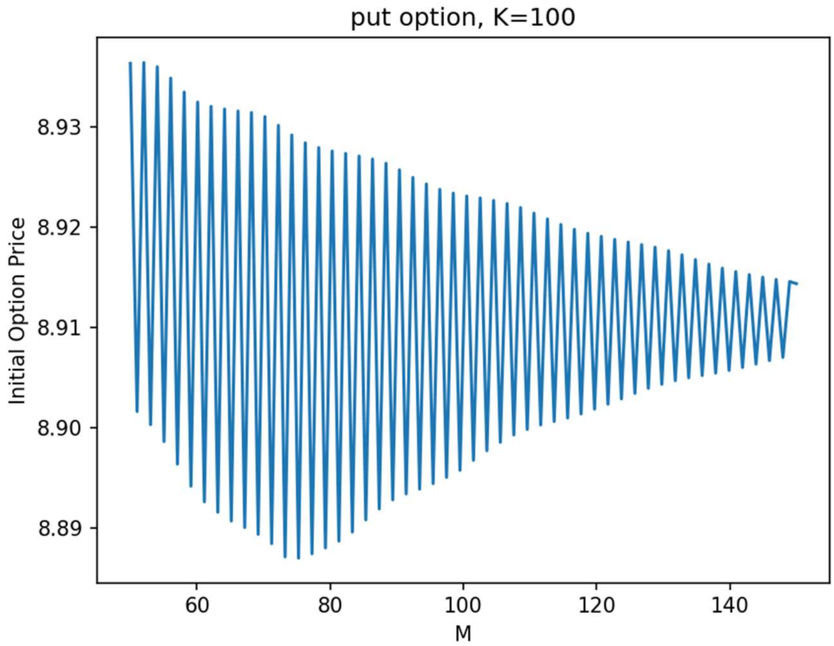
5- Varying M

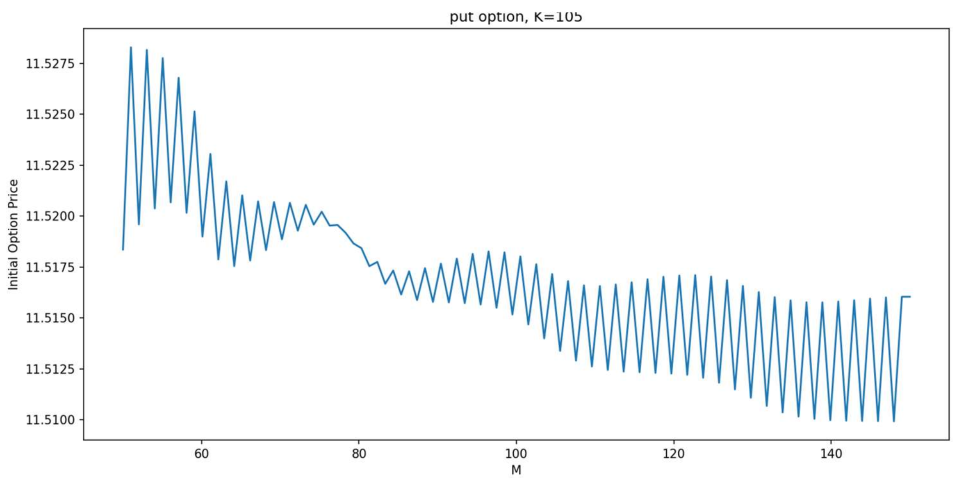

call option, K=105
Initial Option Price
13.380
13.375
13.370
13.365
13.360
13.355
13.350
13.345
60
80
100
120
140
M


call option, K=95
Initial Option Price
18.42
18.40
18.38
18.36
18.34
60
80
100
120
140
M


call option, K=100
Initial Option Price
15.76
15.74
15.72
15.70
15.68
60
80
100
120
140
M


put option, K=95
Initial Option Price
M
6.67
6.68
6.69
6.70
6.71
6.72
60
80
100
120
140


put option, K=100
Initial Option Price
M
8.89
8.90
8.91
8.92
8.93
60
80
100
120
140


put option, K=105
Initial Option Price
M
11.5100
11.5125
11.5150
11.5175
11.5200
11.5225
11.5250
11.5275
60
80
100
120
140

Q2) *Lookback European Option:*

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

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

a)

For M = 5

Initial Price of lookback Option = 15.372952215663778

Execution Time = 0.001020193099975586 sec

For M=10

Initial Price of lookback Option = 16.95034049177767

Execution Time = 0.024021148681640625 sec

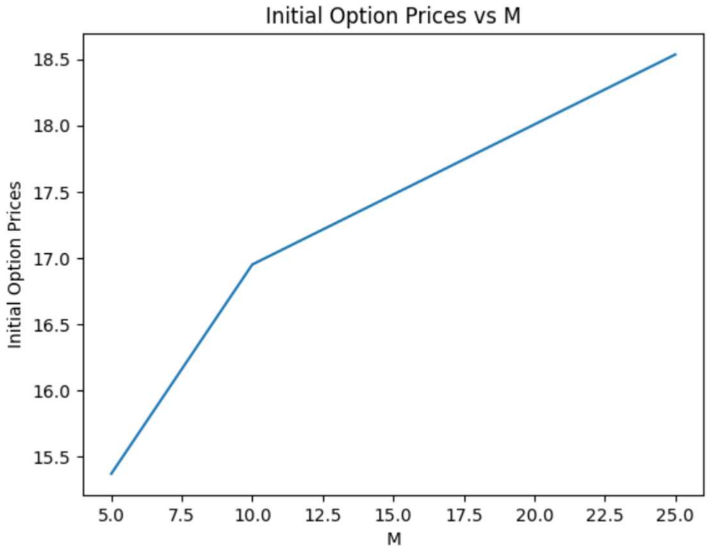
For M=20

Initial Price of lookback Option = 18.533781500094165

Execution Time = 261.82046461105347 sec

For M=50, the time taken to evaluate pricing becomes extremely using this method, hence it is unfeasible to do so.

b)



Hence, as M increases, prices increase,
but tend to converge after as values of M increase further.

Prices at different timepoints:

```
At t = 0
Index no = 0    Price = 15.372952215663778

At t = 1
Index no = 0    Price = 15.532131468492956
Index no = 1    Price = 15.709699760878111

At t = 2
Index no = 0    Price = 15.199750099616727
Index no = 1    Price = 16.365773501799975
Index no = 2    Price = 11.62259245758552
Index no = 3    Price = 20.30531014128848

At t = 3
Index no = 0    Price = 13.386169289151374
Index no = 1    Price = 17.50446467389843
Index no = 2    Price = 10.235825536366997
Index no = 3    Price = 23.026215406441317
Index no = 4    Price = 10.235825536367
Index no = 5    Price = 13.384908157013323
Index no = 6    Price = 12.702323203700722
Index no = 7    Price = 28.566489442465258
```

```
At t = 4
Index no = 0    Price = 10.33248062285694
Index no = 1    Price = 16.872978416162187
Index no = 2    Price = 7.900801695311674
Index no = 3    Price = 27.676760285887045
Index no = 4    Price = 7.900801695311674
Index no = 5    Price = 12.902037888217311
Index no = 6    Price = 12.103285439254641
Index no = 7    Price = 34.69646280474455
Index no = 8    Price = 7.900801695311674
Index no = 9    Price = 12.902037888217318
Index no = 10   Price = 6.041401838252844
Index no = 11   Price = 21.163223292550345
Index no = 12   Price = 6.041401838252844
Index no = 13   Price = 19.775755431345573
Index no = 14   Price = 19.77575543134555
Index no = 15   Price = 38.28243217635733
```

```
At t = 5
Index no = 0    Price = 0.0
Index no = 1    Price = 21.002491662264447
Index no = 2    Price = 0.0
Index no = 3    Price = 34.29714522948986
Index no = 4    Price = 0.0
Index no = 5    Price = 16.05969832296735
Index no = 6    Price = 14.189941164644068
Index no = 7    Price = 42.06197481701972
Index no = 8    Price = 0.0
Index no = 9    Price = 16.05969832296735
Index no = 10   Price = 0.0
Index no = 11   Price = 26.225545739139193
Index no = 12   Price = 0.0
Index no = 13   Price = 24.601948051238253
Index no = 14   Price = 24.601948051238267
Index no = 15   Price = 45.914488453717624
Index no = 16   Price = 0.0
Index no = 17   Price = 16.05969832296735
Index no = 18   Price = 0.0
Index no = 19   Price = 26.225545739139207
Index no = 20   Price = 0.0
Index no = 21   Price = 12.280157724719814
Index no = 22   Price = 10.850435176426544
Index no = 23   Price = 32.162975578905915
Index no = 24   Price = 0.0
Index no = 25   Price = 12.280157724719814
Index no = 26   Price = 9.440589282577335
Index no = 27   Price = 30.75312968505669
Index no = 28   Price = 9.440589282577307
Index no = 29   Price = 30.753129685056678
Index no = 30   Price = 30.753129685056678
Index no = 31   Price = 47.04990888934698
```

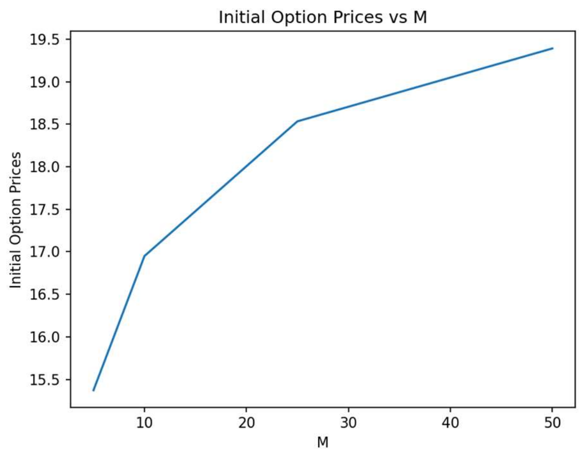
Q3)

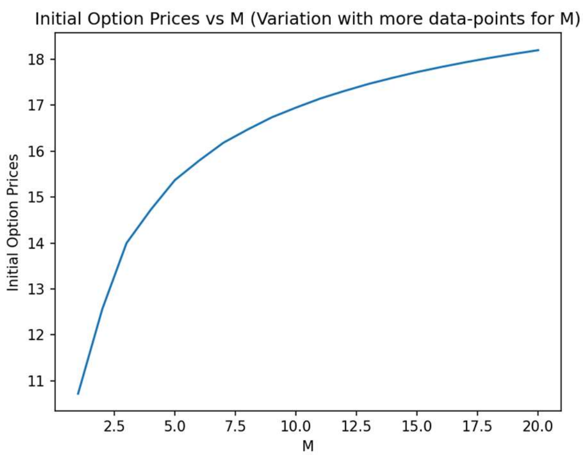
Comparative analysis of the two algorithms, like computational time, the value of M it can handle, etc.

a)

```
*********  Executing for M = 5  *********

No arbitrage exists for M = 5
Initial Price of Loopback Option        = 15.372952215663778
Execution Time                          = 0.0 sec



*********  Executing for M = 10  *********

No arbitrage exists for M = 10
Initial Price of Loopback Option        = 16.95034049177767
Execution Time                          = 0.0 sec



*********  Executing for M = 25  *********

No arbitrage exists for M = 25
Initial Price of Loopback Option        = 18.533781500094165
Execution Time                          = 0.11603450775146484 sec



*********  Executing for M = 50  *********

No arbitrage exists for M = 50
Initial Price of Loopback Option        = 19.390465235522452
Execution Time                          = 8.667919874191284 sec
```

b)





c)

Prices at different timepoints for optimized algorithm:

```
At t = 0
Intermediate state = (100, 100)        Price = 15.372952215663778

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

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

At t = 3
Intermediate state = (152.7285895992882, 152.7285895992882)            Price = 13.386169289151374
Intermediate state = (116.78495645656189, 132.6213855344424)           Price = 17.50446467389843
Intermediate state = (116.78495645656187, 116.78495645656187)          Price = 10.235825536366997
Intermediate state = (89.3004125183424, 115.16135876866093)            Price = 23.026215406441317
Intermediate state = (116.78495645656189, 116.78495645656189)          Price = 10.235825536367
Intermediate state = (89.30041251834241, 101.40984589384924)           Price = 13.384908157013323
Intermediate state = (89.3004125183424, 100)           Price = 12.702323203700722
Intermediate state = (68.28416876545448, 100)          Price = 28.566489442465258

At t = 4
Intermediate state = (175.88431901075205, 175.88431901075205)          Price = 10.33248062285694
Intermediate state = (134.4911426927657, 152.7285895992882)            Price = 16.872978416162187
Intermediate state = (134.4911426927657, 134.4911426927657)            Price = 7.900801695311674
Intermediate state = (102.8395684421425, 132.6213855344424)            Price = 27.676760285887045
Intermediate state = (134.49114269276566, 134.49114269276566)          Price = 7.900801695311674
Intermediate state = (102.8395684421425, 116.78495645656187)           Price = 12.902037888217311
Intermediate state = (102.8395684421425, 115.16135876866093)           Price = 12.103285439254641
Intermediate state = (78.63697657418294, 115.16135876866093)           Price = 34.69646280474455
Intermediate state = (102.8395684421425, 116.78495645656189)           Price = 12.902037888217318
Intermediate state = (102.83956844214251, 102.83956844214251)          Price = 6.041401838252844
Intermediate state = (78.63697657418295, 101.40984589384924)           Price = 21.163223292550345
Intermediate state = (102.8395684421425, 102.8395684421425)            Price = 6.041401838252844
Intermediate state = (78.63697657418294, 100)          Price = 19.775755431345573
Intermediate state = (78.63697657418295, 100)          Price = 19.77575543134555
Intermediate state = (60.130299829171165, 100)         Price = 38.28243217635733
```

```
At t = 5
Intermediate state = (202.5507716337883, 202.5507716337883)            Price = 0.0
Intermediate state = (154.8818273484876, 175.88431901075205)           Price = 21.002491662264447
Intermediate state = (154.8818273484876, 154.8818273484876)            Price = 0.0
Intermediate state = (118.43144436979834, 152.7285895992882)           Price = 34.29714522948986
Intermediate state = (118.43144436979834, 134.4911426927657)           Price = 16.05969832296735
Intermediate state = (118.43144436979833, 132.6213855344424)           Price = 14.189941164644068
Intermediate state = (90.55941071742268, 132.6213855344424)            Price = 42.06197481701972
Intermediate state = (154.88182734848758, 154.88182734848758)          Price = 0.0
Intermediate state = (118.43144436979831, 134.49114269276566)          Price = 16.05969832296735
Intermediate state = (118.43144436979833, 118.43144436979833)          Price = 0.0
Intermediate state = (90.55941071742268, 116.78495645656187)           Price = 26.225545739139193
Intermediate state = (90.55941071742268, 115.16135876866093)           Price = 24.601948051238253
Intermediate state = (90.55941071742267, 115.16135876866093)           Price = 24.601948051238267
Intermediate state = (69.24687031494331, 115.16135876866093)           Price = 45.914488453717624
Intermediate state = (90.55941071742268, 116.78495645656189)           Price = 26.225545739139207
Intermediate state = (118.43144436979834, 118.43144436979834)          Price = 0.0
Intermediate state = (90.5594107174227, 102.83956844214251)            Price = 12.280157724719814
Intermediate state = (90.5594107174227, 101.40984589384924)            Price = 10.850435176426544
Intermediate state = (69.24687031494332, 101.40984589384924)           Price = 32.162975578905915
Intermediate state = (90.55941071742268, 102.8395684421425)            Price = 12.280157724719814
Intermediate state = (90.55941071742267, 100)          Price = 9.440589282577335
Intermediate state = (69.24687031494331, 100)          Price = 30.75312968505669
Intermediate state = (90.5594107174227, 100)           Price = 9.440589282577307
Intermediate state = (69.24687031494332, 100)          Price = 30.753129685056678
Intermediate state = (52.95009111065302, 100)          Price = 47.04990888934698
```

Comparitive Analysis:

Using Markov based optimization, we can now compute the option prices for larger values of M.

M	Initial Option Price	Computational Time (in seconds)
5	9.1193	0.000160
10	10.0806	0.000744
20	10.8051	0.017535
25	11.0034	0.05155
50	11.51086	3.11132

The highest value of M the algorithms can handle:

a. **Unoptimised Algorithm -** around 25 to 30. The execution time almost doubles up as M is increased by 1 unit. At M = 50, the google colab executing this code crashes due to insufficient RAM.
b. **Markov based Algorithm -** around 85. At M = 90, the google colab executing this code crashes due to insufficient RAM.

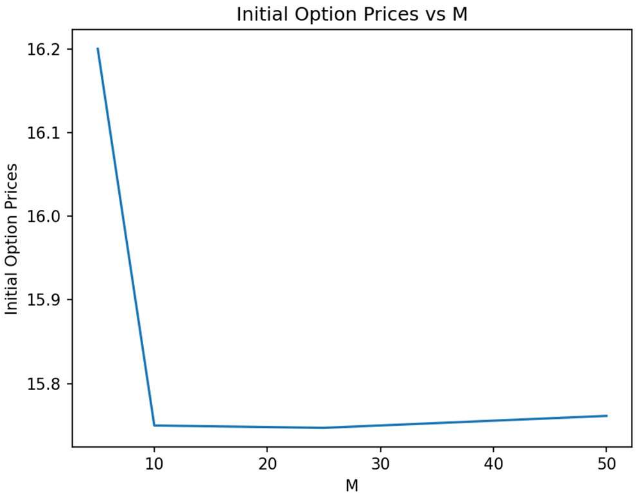
The markov-based algorithm is more efficient than original one, allowing us to calculate values for higher M.

Q4)

a)

```
######################  Efficient Binomial Algorithm executing (Markov Based)  ######################
No arbitrage exists for M = 5
European Call Option            = 16.200135785709463
Execution Time                  = 0.0 sec

No arbitrage exists for M = 10
European Call Option            = 15.749706920472503
Execution Time                  = 0.0 sec

No arbitrage exists for M = 25
European Call Option            = 15.746918255600457
Execution Time                  = 0.0009987354278564453 sec

No arbitrage exists for M = 50
European Call Option            = 15.761196879829438
Execution Time                  = 0.00099945068359375 sec
```

b)



c) prices at different timepoints:

```
At t = 0
Index no = 0    Price = 16.200135785709463

At t = 1
Index no = 0    Price = 25.375255893366354
Index no = 1    Price = 7.543996674048174

At t = 2
Index no = 0    Price = 38.432095157524756
Index no = 1    Price = 13.131857964608423
Index no = 2    Price = 2.1972816917220994

At t = 3
Index no = 0    Price = 55.877931391368456
Index no = 1    Price = 22.219195424615478
Index no = 2    Price = 4.464542360415781
Index no = 3    Price = 0.0

At t = 4
Index no = 0    Price = 77.47158700522355
Index no = 1    Price = 36.078410687237174
Index no = 2    Price = 9.071271363629885
Index no = 3    Price = 0.0
Index no = 4    Price = 0.0

At t = 5
Index no = 0    Price = 102.55077163378829
Index no = 1    Price = 54.881827348487604
Index no = 2    Price = 18.431444369798328
Index no = 3    Price = 0
Index no = 4    Price = 0
Index no = 5    Price = 0
```

Comparitive Analysis:

M	Computational Time with Markov (in seconds)
5	0.00002
10	0.00014
15	0.00020
20	0.00058
25	0.00105
50	0.00839

The markov-based algorithm is more efficient than original one, since the complexity of naïve approach is $O(2^n)$, while for for Markov based algorithm, it is $O(n^2)$, allowing us to calculate values for higher M. The efficient algorithm can handle values of M upto $2*10^4$.