MA374 – Financial Engineering Laboratory

Lab - 03

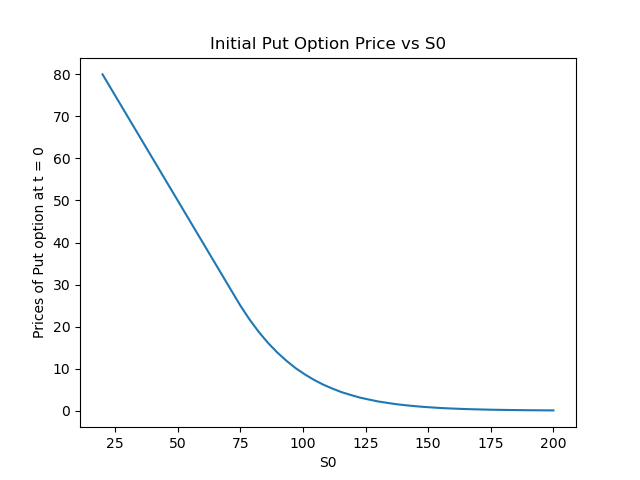
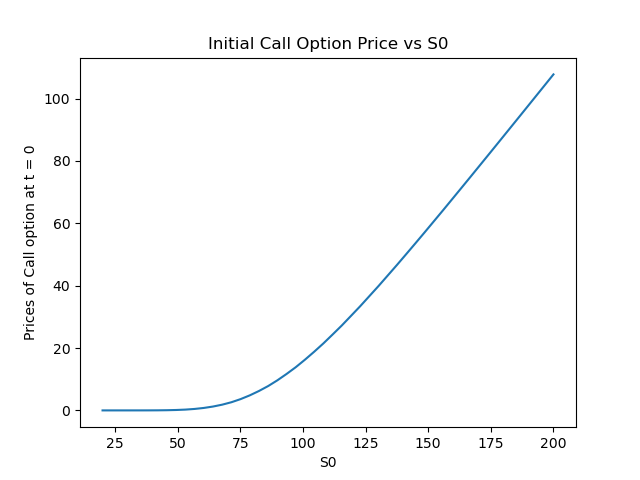
Dipanshu Goyal 210123083

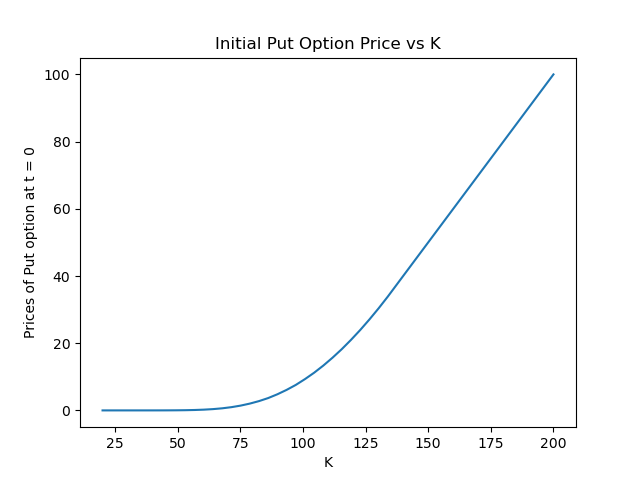
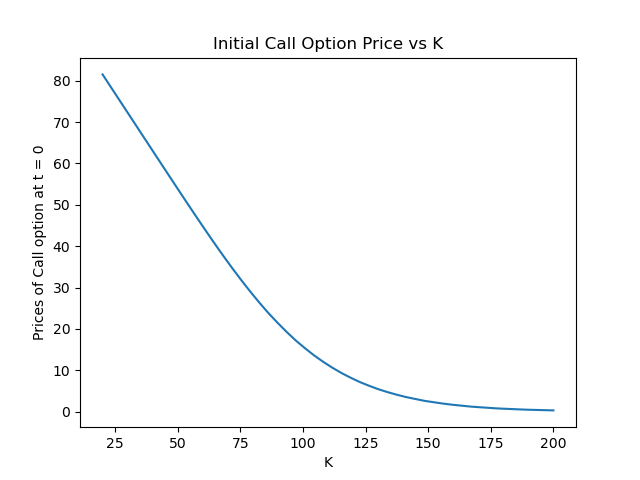
# Ques - 1

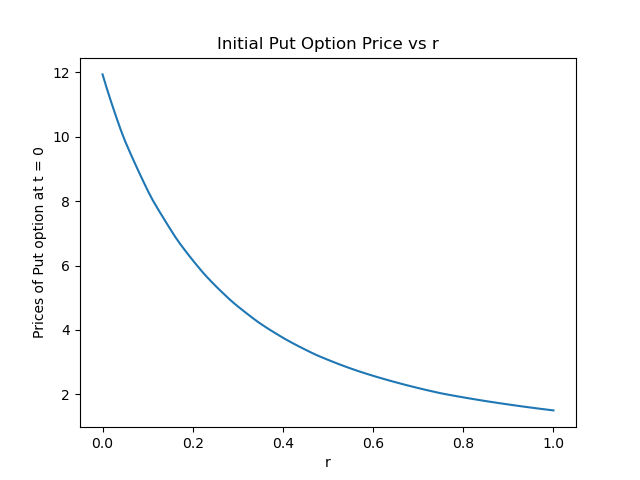
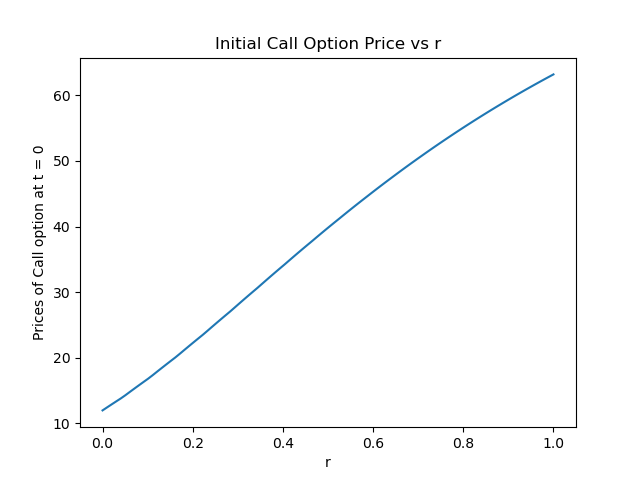
The initial option prices for American Call Option and American Put Option are: -

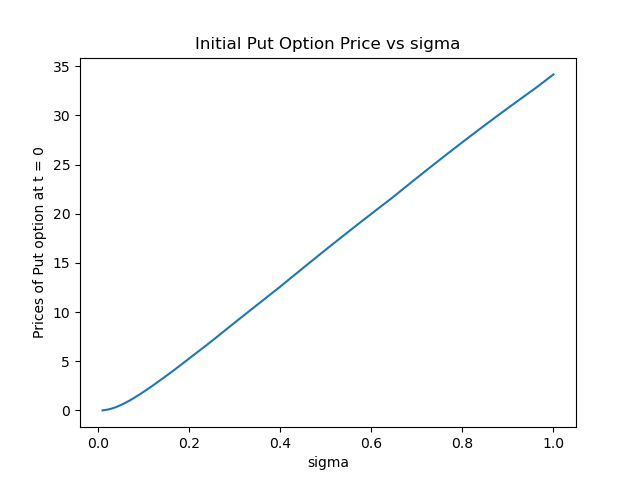
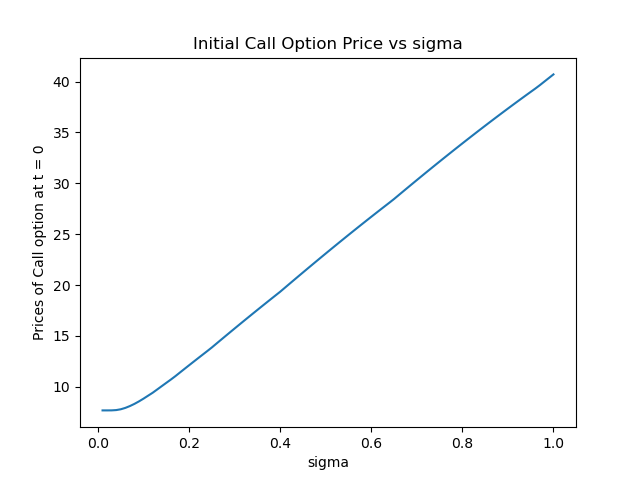


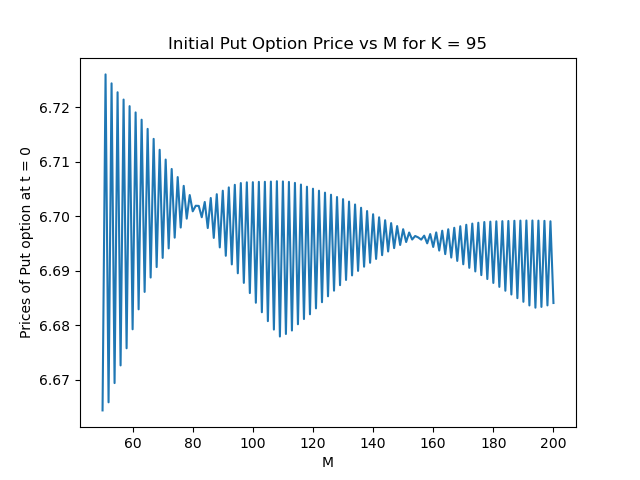
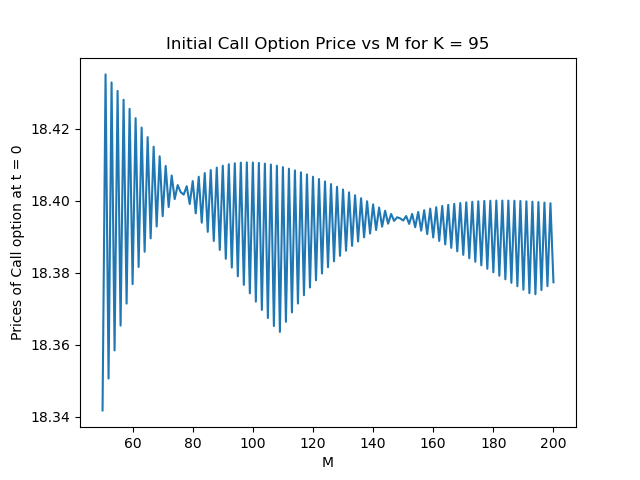
**Sensitivity Analysis**

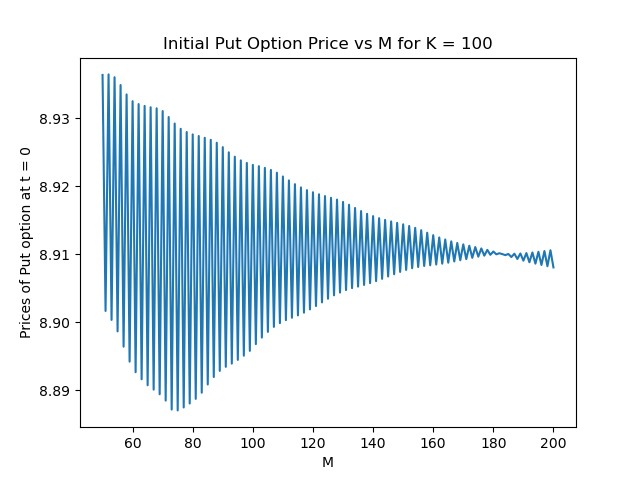
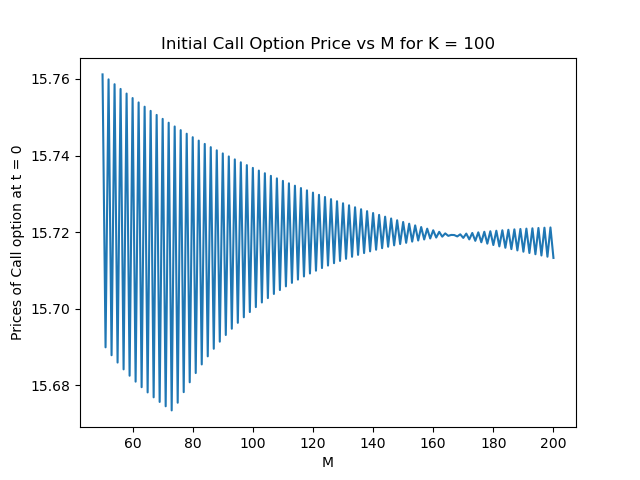


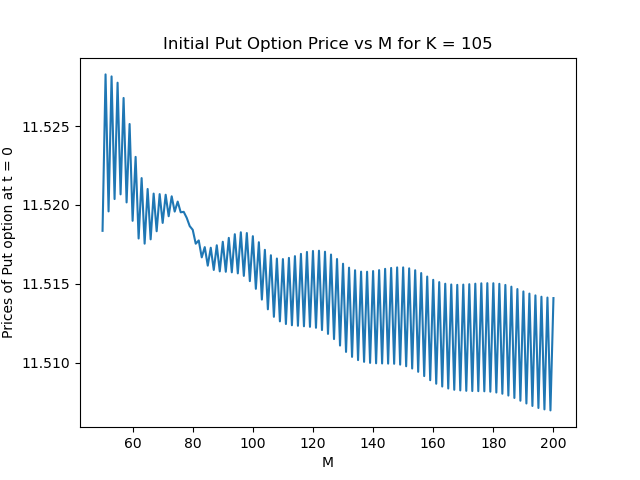
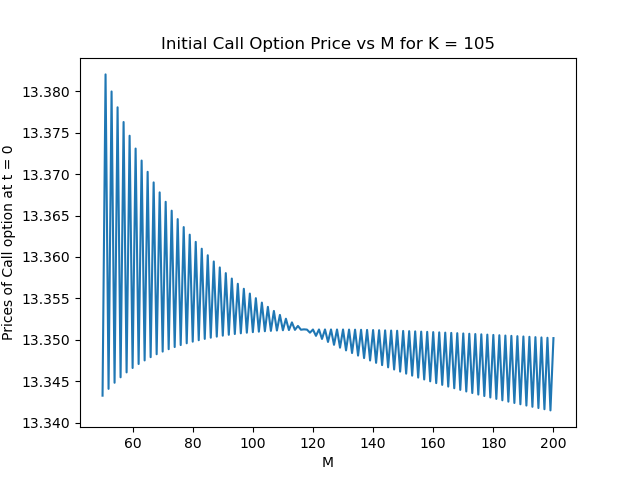












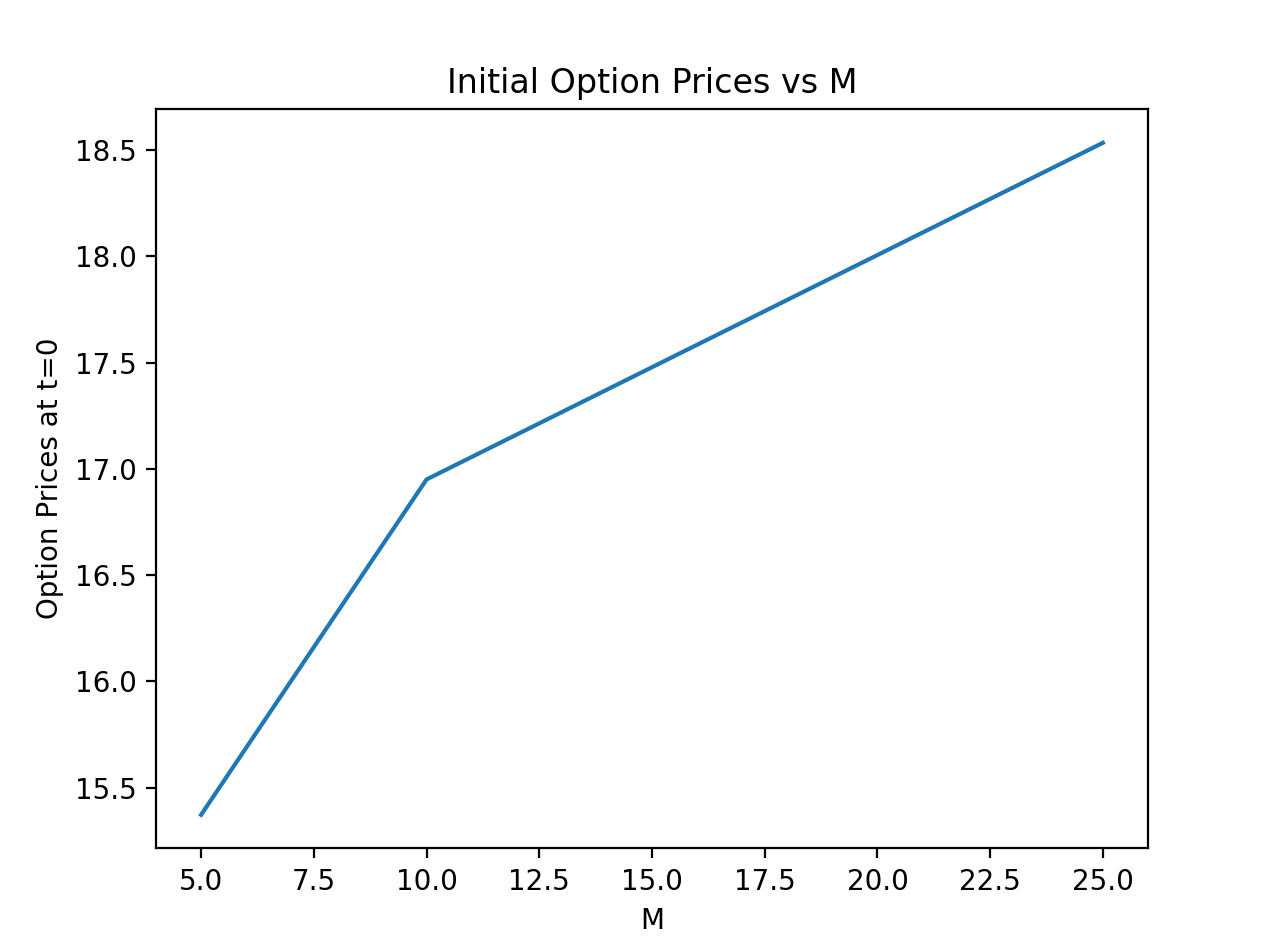
# Ques – 2

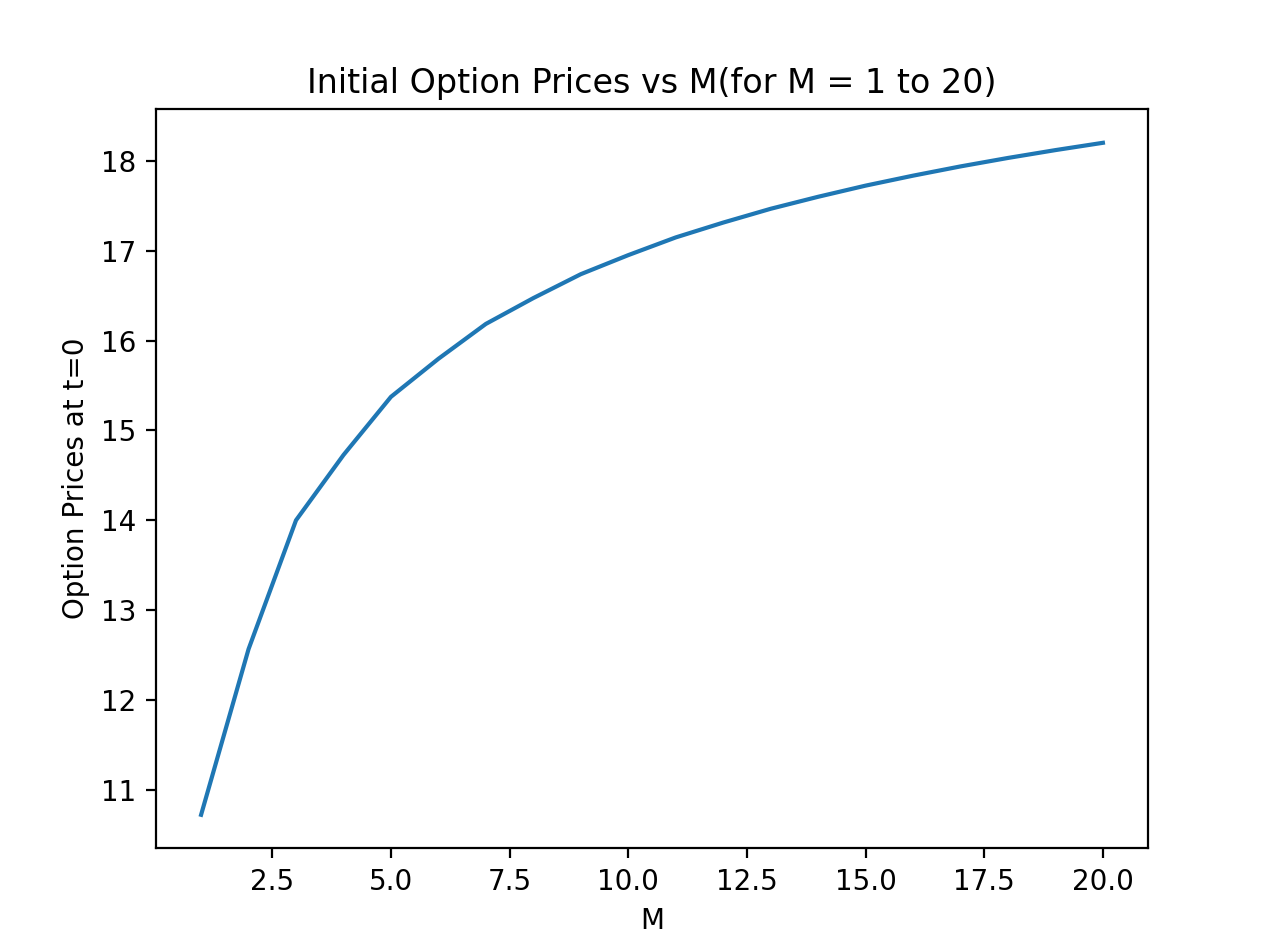
1. The initial option prices for Lookback Option are: -

|  |  |  |
| --- | --- | --- |
| **M** | **Initial Option Price** | **Execution Time** |
| 5 | 15.372952215663778 | 0.0005900859832763672 sec |
| 10 | 16.95034049177767 | 0.006894111633300781 sec |
| 25 | 18.533781500094165 | 217.85930705070496 sec |
| 50 | Not Feasible | Not Feasible |

For M = 50, the algorithm is unable to calculate the option price since it has exponential time complexity. So, computation for M = 50 is not feasible with this algorithm for finding out the loopback option.

1. The plots comparing the above values are as follows:





As the value of M increases, the initial option price increases, and it seems that the prices tend to converge as M is increased further.

1. The values of the options at all intermediate time points for M = 5:

**(Note: ‘t’ shows time intervals wrt M)**

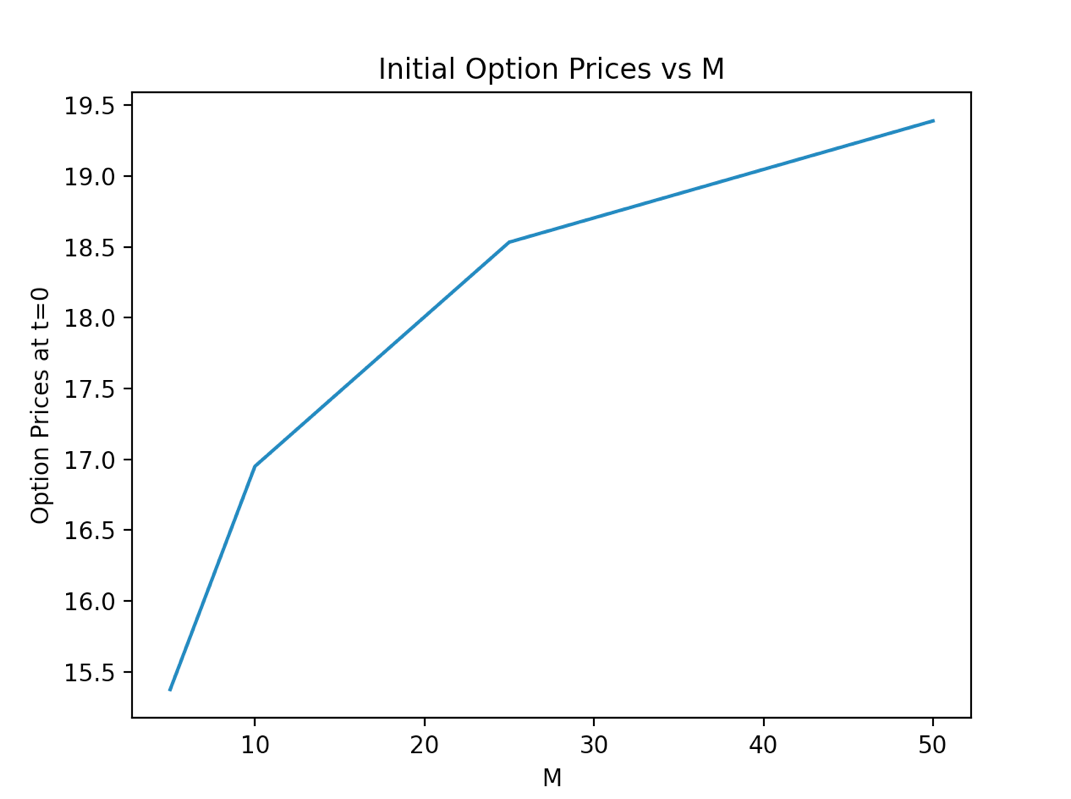
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **t=0** | **t=1** | **t=2** | **t=3** | **t=4** | **t=5** |
| **1.** | 15.3729522 | 15.5321315 | 15.1997501 | 13.3861693 | 10.3324806 | 0.0 |
| **2.** | x | 15.7096998 | 16.3657735 | 17.5044647 | 16.8729784 | 21.0024917 |
| **3.** | x | x | 11.6225925 | 10.2358255 | 7.9008017 | 0.0 |
| **4.** | x | x | 20.3053101 | 23.0262154 | 27.6767603 | 34.2971452 |
| **5.** | x | x | x | 10.2358255 | 7.9008017 | 0.0 |
| **6.** | x | x | x | 13.3849082 | 12.9020379 | 16.0596983 |
| **7.** | x | x | x | 12.7023232 | 12.1032854 | 14.1899412 |
| **8.** | x | x | x | 28.5664894 | 34.6964628 | 42.0619748 |
| **9.** | x | x | x | x | 7.9008017 | 0.0 |
| **10.** | x | x | x | x | 12.9020379 | 16.0596983 |
| **11.** | x | x | x | x | 6.0414018 | 0.0 |
| **12.** | x | x | x | x | 21.1632233 | 26.2255457 |
| **13.** | x | x | x | x | 6.0414018 | 0.0 |
| **14.** | x | x | x | x | 19.7757554 | 24.6019481 |
| **15.** | x | x | x | x | 19.7757554 | 24.6019481 |
| **16.** | x | x | x | x | 38.2824322 | 45.9144885 |
| **17.** | x | x | x | x | x | 0.0 |
| **18.** | x | x | x | x | x | 16.0596983 |
| **19.** | x | x | x | x | x | 0.0 |
| **20.** | x | x | x | x | x | 26.2255457 |
| **21.** | x | x | x | x | x | 0.0 |
| **22.** | x | x | x | x | x | 12.2801577 |
| **23.** | x | x | x | x | x | 10.8504352 |
| **24.** | x | x | x | x | x | 32.1629756 |
| **25.** | x | x | x | x | x | 0.0 |
| **26.** | x | x | x | x | x | 12.2801577 |
| **27.** | x | x | x | x | x | 9.4405893 |
| **28.** | x | x | x | x | x | 30.7531297 |
| **29.** | x | x | x | x | x | 9.4405893 |
| **30.** | x | x | x | x | x | 30.7531297 |
| **31.** | x | x | x | x | x | 30.7531297 |
| **32.** | x | x | x | x | x | 47.0499089 |

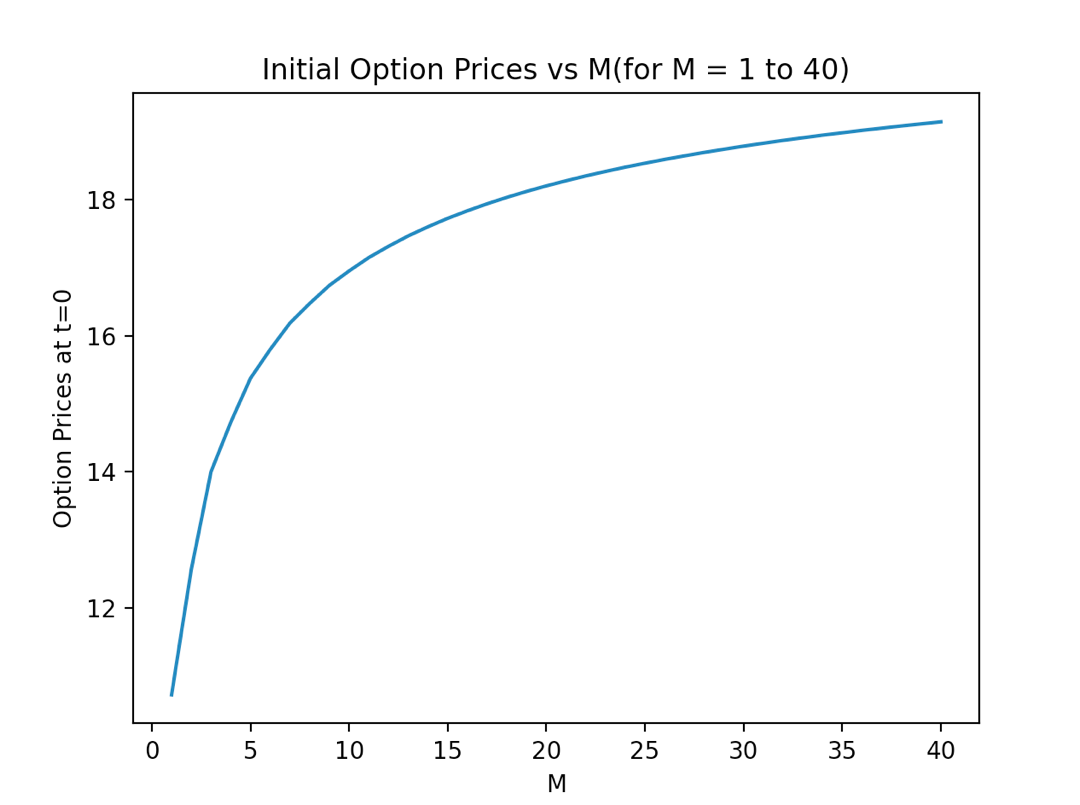
## Ques - 3

Using Markov property, the efficient algorithm gives option prices for lookback option as follows: -

|  |  |  |
| --- | --- | --- |
| **M** | **Initial Option Price** | **Execution Time** |
| 5 | 15.372952215663778 | 0.0002429485321044922 sec |
| 10 | 16.95034049177767 | 0.0016510486602783203 sec |
| 25 | 18.533781500094165 | 0.11969828605651855 sec |
| 50 | 19.390465235522452 | 7.935082197189331 sec |

The plots comparing the above values are as follows: -





The values of the options at all intermediate time points for M = 5: -

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **t=0** | **t=1** | **t=2** | **t=3** | **t=4** | **t=5** |
| **1.** | 15.3729522 | 15.5321315 | 15.1997501 | 13.3861693 | 10.3324806 | 0.0 |
| **2.** | x | 15.7096998 | 16.3657735 | 17.5044647 | 16.8729784 | 21.0024917 |
| **3.** | x | x | 11.6225925 | 10.2358255 | 7.9008017 | 0.0 |
| **4.** | x | x | 20.3053101 | 23.0262154 | 27.6767603 | 34.2971452 |
| **5.** | x | x | x | 10.2358255 | 7.9008017 | 16.0596983 |
| **6.** | x | x | x | 13.3849082 | 12.9020379 | 14.1899412 |
| **7.** | x | x | x | 12.7023232 | 12.1032854 | 42.0619748 |
| **8.** | x | x | x | 28.5664894 | 34.6964628 | 0.0 |
| **9.** | x | x | x | x | 12.9020379 | 16.0596983 |
| **10.** | x | x | x | x | 6.0414018 | 0.0 |
| **11.** | x | x | x | x | 21.1632233 | 26.2255457 |
| **12.** | x | x | x | x | 6.0414018 | 24.6019481 |
| **13.** | x | x | x | x | 19.7757554 | 24.6019481 |
| **14.** | x | x | x | x | 19.7757554 | 45.9144885 |
| **15.** | x | x | x | x | 38.2824322 | 26.2255457 |
| **16.** | x | x | x | x | x | 0.0 |
| **17.** | x | x | x | x | x | 12.2801577 |
| **18.** | x | x | x | x | x | 10.8504352 |
| **19.** | x | x | x | x | x | 32.1629756 |
| **20.** | x | x | x | x | x | 12.2801577 |
| **21.** | x | x | x | x | x | 9.4405893 |
| **22.** | x | x | x | x | x | 30.7531297 |
| **23.** | x | x | x | x | x | 9.4405893 |
| **24.** | x | x | x | x | x | 30.7531297 |
| **25.** | x | x | x | x | x | 47.0499089 |

**Comparative Analysis: -**

1. Comparing Execution times.

|  |  |  |
| --- | --- | --- |
| **M** | **Basic** | **Markov Based** |
| 5 | 0.0005900859832763672 sec | 0.0002429485321044922 sec |
| 10 | 0.006894111633300781 sec | 0.0016510486602783203 sec |
| 25 | 217.85930705070496 sec | 0.11969828605651855 sec |
| 50 | Not Feasible | 7.935082197189331 sec |

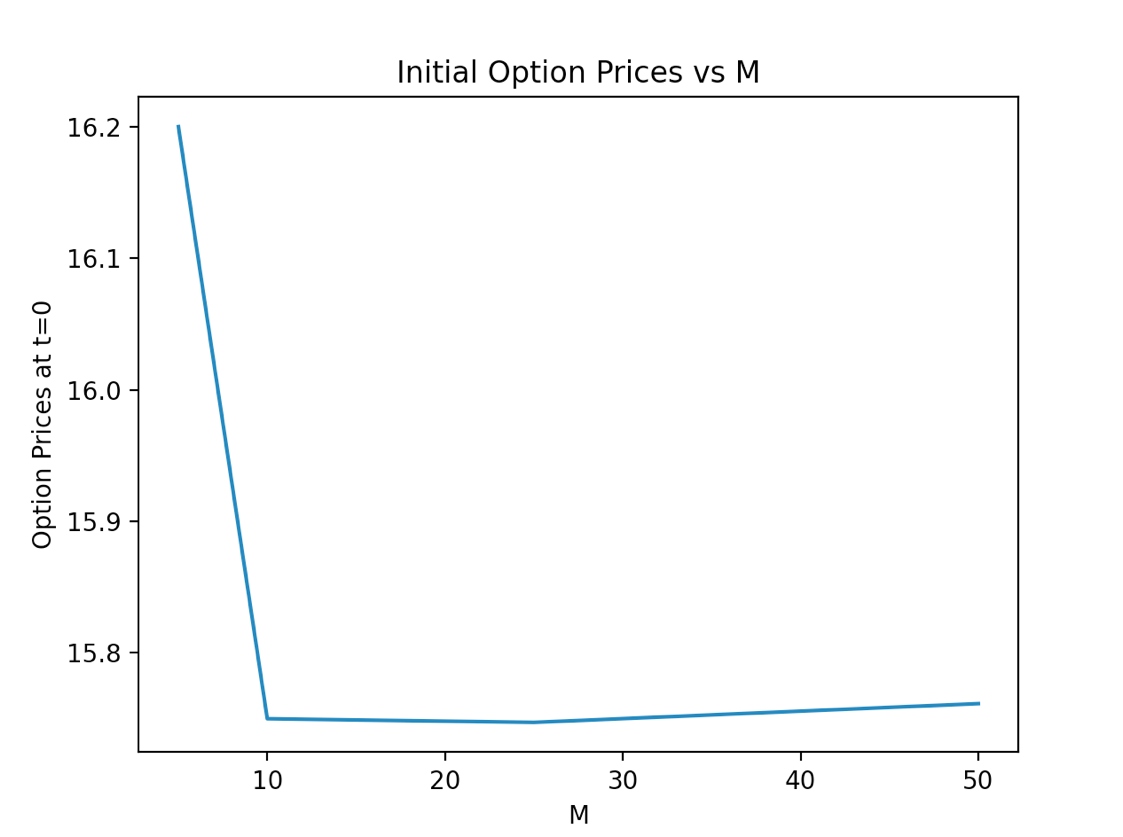
1. The highest value of M that can be handled by algorithm: -
2. Basic: - Around 25 to 30.
3. Markov Based: - Around 85-87.
4. The basic algorithm has exponential time complexity, while Markov based doesn’t have exponential time complexity as it is implemented using memorization technique in dynamic programming.

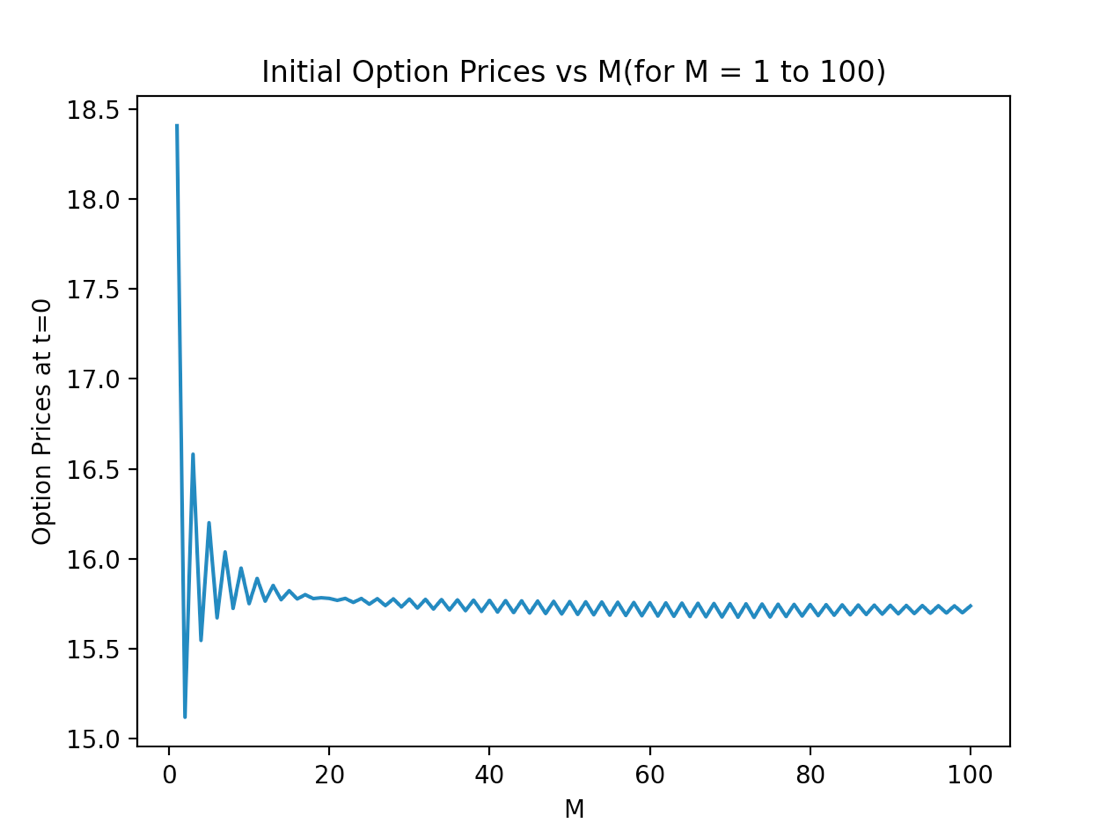
## Ques - 4

1. The initial option prices for European Call Option are: -

|  |  |
| --- | --- |
| **M** | **Initial Option Price** |
| 5 | 16.200135785709463 |
| 10 | 15.749706920472503 |
| 25 | 15.746918255600457 |
| 50 | 15.761196879829438 |

1. The plots comparing above values are: -





The option price is converging to roughly 15.73 as M increases.

1. The values of the option at all intermediate time points for M = 5 is as follows: -

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **t=0** | **t=1** | **t=2** | **t=3** | **t=4** | **t=5** |
| **1.** | 16.200135785 | 25.375255893 | 38.432095157 | 55.877931391 | 7.47158700 | 102.55077163 |
| **2.** | x | 7.543996674 | 13.131857964 | 22.219195424 | 36.078410687 | 54.881827348 |
| **3.** | x | x | 2.1972816917 | 4.464542360 | 9.071271363 | 18.431444369 |
| **4.** | x | x | x | 0.0 | 0.0 | 0 |
| **5.** | x | x | x | x | 0.0 | 0 |
| **6.** | x | x | x | x | x | 0 |

**Comparative Analysis: -**

1. Comparing Execution times.

|  |  |  |  |
| --- | --- | --- | --- |
| **M** | **Unoptimised** | **Better** | **Best** |
| 5 | 0.000440120697021 sec | 5.602836608886e-05 sec | 5.316734313964e-05 sec |
| 10 | 0.004103183746337 sec | 9.393692016601e-05 sec | 6.86645507812e-05 sec |
| 25 | 101.93571686 sec | 0.0003497600555419 sec | 0.0001471042633056 sec |
| 50 | Not Feasible | 0.001171112060546 sec | 0.000314950942993 sec |

1. The highest value of M that can be handled by algorithm: -
2. Unoptimized: - Around 25 to 30.
3. Better: - Around 2\*10^4.
4. Best: - Around 1000 (nCr overflows after 1000)
5. The unoptimized algorithm has exponential time and space complexity. The efficient algorithm has quadratic space and time complexity (in M), although space complexity can be decreased to linear. But since we also needed to print intermediate information, I implemented algorithm which has quadratic complexity. The most efficient algorithm has almost linear time complexity (after making use of memoization to calculate nCr), and linear space complexity.
6. The most efficient algorithm works on the same principle as that of efficient algorithm. The only difference lies in the fact that the most efficient algorithm summarises the computation of efficient algorithm to a formula. So, it is much easier to implement, and is faster than the efficient algorithm. But the upper limit of M is less due to computational issues like integer overflow and stack overflow.