

# MA374 Financial Engineering lab: 03

Name: Naman Goyal

Roll No. 180123029

## Ques.1

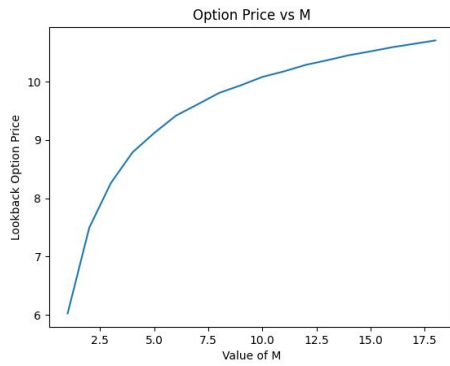
- To execute my .py file  
Run `$python3 180123029_NamanGoyal_q1.py` on the terminal. The snapshot is given below

```
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/Lab03$ python3 180123029_NamanGoyal_q1.py
For M = 5 Lookback Option Price is 9.119298985864688
The time elapsed for M = 5 => 0.0001678466796875
For M = 10 Lookback Option Price is 10.080582906831015
The time elapsed for M = 10 => 0.002832174301147461
For M = 20 Lookback Option Price is 10.805118587154096
The time elapsed for M = 20 => 3.391453742980957
Option Price value for different time intervals are
For t = 1.0
[0.0, 11.181413117784501, 0.0, 19.452691543130413, 0.0, 9.349916553291678, 6.374517470614265, 25.39456347506497, 0.0, 9.349916553291678, 0.0, 16.2
66373556657385, 0.0, 13.578002496522686, 13.578002496522686, 29.48259712227059, 0.0, 9.349916553291678, 0.0, 16.266373556657385, 0.0, 7.8184160295
867144, 5.330382286201839, 21.234976911949744, 0.0, 7.8184160295867144, 2.9013504971397026, 18.805945122887607, 2.9013504971397026, 18.80594512288
7607, 18.805945122887607, 32.10539403853048]
For t = 0.8
[5.501638813873981, 9.57139153170023, 4.600479677676438, 15.631851880479829, 4.600479677676438, 8.003613780975444, 6.680842999256647, 21.188089345
345652, 4.600479677676438, 8.003613780975444, 3.846928884415608, 13.07138097092879, 3.846928884415608, 10.680904426029972, 10.680904426029972, 25.
05122945703703]
For t = 0.6000000000000001
[7.416771005131012, 9.95527127295782, 6.201916453882752, 13.712862965988537, 6.201916453882752, 8.324614669633142, 7.1484182081901215, 17.58206271
4095425]
For t = 0.4
[8.548076183576446, 9.79911875354703, 7.1479157567747444, 12.168664659721797]
For t = 0.2
[9.027951165547757, 9.504839866450858]
For t = 0.0
[9.11929898586469]
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/Lab03$
```

- Values of **M** were taken to be **5, 10, 20** and using the **lookback option** in the **binomial model**. Higher values of M were not taken as the algorithm is **exponential** and takes a lot of time.

M	Initial Option Price	Time elapsed
5	9.1193	0.00016
10	10.0805	0.00283
20	10.8051	3.39145

- Value of initial price will converge as the value of **M** is increased, time taken will become exponential as M increases.



- The graph eventually converges. Max value of the M algorithm will handle will be around  $M = 23$ .

- The loopback option values at intermediate time points for  $M = 5$ :

T	0	1	2	3	4
Value of Options	9.1192	9.0279	7.1479	6.2019	3.8469
		9.5048	8.5480	6.2019	3.8469
			9.7991	7.1484	4.6004
			12.1686	7.4167	4.6004
				8.3246	4.6004
				9.9552	5.5016
				13.7128	6.6808
				17.5820	8.0036
					8.0036
					9.5713
					10.6809
					10.6809
					13.0713
					15.6318
					21.1880
					25.0512

---

### Ques.2

- To execute my .py file  
Run `$python3 180123029_NamanGoyal_q2.py` on the terminal. The snapshot is given below

```
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/Lab03$ python3 180123029_NamanGoyal_q2.py
For M = 5 Lookback option Value is 9.11929898586469
Time elapsed for M = 5 => 0.0001785755157470703
For M = 10 Lookback option Value is 10.08058290683101
Time elapsed for M = 10 => 0.00128936767578125
For M = 15 Lookback option Value is 10.519164595672923
Time elapsed for M = 15 => 0.005560636520385742
For M = 25 Lookback option Value is 11.003495335646338
Time elapsed for M = 25 => 0.0824744701385498
For M = 50 Lookback option Value is 11.510862222177268
Time elapsed for M = 50 => 5.079269170761108
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/Lab03$ |
```

- This is an **efficient algorithm** and can handle large values of M as the **time of the algorithm is less** as compared to the normal one and **converges on increasing M**.

M	Initial Option Price	Time elapsed
5	9.1192	0.00017
10	10.0805	0.00128
15	10.5191	0.00556
25	11.0034	0.08247
50	11.5108	5.07926

- Since **large values of M can be handled** but I suspect the **max limit to be around 56**. The algorithm is efficient than the previous algorithm.
-

### Ques.3

- To execute my .py file  
Run `$python3 180123029_NamanGoyal_q3.py` on the terminal. The snapshot is given below

```
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/lab03$ python3 180123029_NamanGoyal_q3.py
Value of European Call for M = 10 Normally 13.193895951751232
Value of European Call for M = 10 Efficiently 12.27732781922299
Time elapsed for M = 10 => 0.0024385452270507812
Value of European Call for M = 15 Normally 12.986335667031911
Value of European Call for M = 15 Efficiently 12.05200499188289
Time elapsed for M = 15 => 0.07149386405944824
Normal method can't handle value of M = 25
Value of European Call for M = 25 Efficiently 12.136745963232967
Time elapsed for M = 25 => 0.0005824565887451172
Normal method can't handle value of M = 50
Value of European Call for M = 50 Efficiently 12.085361510072184
Time elapsed for M = 50 => 0.0020580291748046875
Normal method can't handle value of M = 100
Value of European Call for M = 100 Efficiently 12.12304707401244
Time elapsed for M = 100 => 0.007727146148681641
naman-ubuntu@naman-ubuntu:~/Desktop/FE Labs/lab03$ |
```

- Now the **efficient algorithm** is used to calculate **European Call Option Price**. The **time of the algorithm is very less meaning faster algorithm** and can **handle larger values of M**. As the value of M increases, Option price converges the same as the loopback Option.

M	European Price	Time elapsed
10	12.2773	0.00243
15	12.0520	0.07149
25	12.1367	0.00058
50	12.0853	0.00205
100	12.1230	0.00772