NATIONAL UNIVERSITY OF SINGAPORE
Department of Mathematics

2017/2018 **QF4102 Financial Modelling** **Semester I**

QF4102 Assignment 1

A1.1 (i) Write a Matlab function for the exact solution of a European down-and-out call option (see John Hull's chapter on exotic options) denoted by C_{do}. Your function must be able to **work with the initial underlier price S_0 in a vector form** (as done in tut 1 qn 4).

If $H \leq X$ then

$$C_{do} = C - S_0 e^{-q\tau}(H/S_0)^{2\lambda} N(y) + X e^{-r\tau}(H/S_0)^{2\lambda-2} N(y - \sigma\sqrt{\tau})$$

else

$$\begin{aligned} C_{do} &= S_0 N(x_1) e^{-q\tau} - X e^{-r\tau} N(x_1 - \sigma\sqrt{\tau}) - S_0 e^{-q\tau}(H/S_0)^{2\lambda} N(y_1) \\ &\quad + X e^{-r\tau}(H/S_0)^{2\lambda-2} N(y_1 - \sigma\sqrt{\tau}) \end{aligned}$$

with

- C_{do}: option price for the down-and-out call,
- C: Black-Schole price for the equivalent vanilla call,
- q: the continuous dividend rate,
- H: the barrier level,
- S_0: current underlier price,
- X: strike price,
- τ: time to maturity, in years,
- r: risk free rate,
- $\lambda = \dfrac{r - q + \sigma^2/2}{\sigma^2}$,
- $y = \dfrac{\ln[H^2/(S_0 X)]}{\sigma\sqrt{\tau}} + \lambda\sigma\sqrt{\tau}$,
- $x_1 = \dfrac{\ln[S_0/H]}{\sigma\sqrt{\tau}} + \lambda\sigma\sqrt{\tau}$,
- $y_1 = \dfrac{\ln[H/S_0]}{\sigma\sqrt{\tau}} + \lambda\sigma\sqrt{\tau}$.

(ii) Consider a European down-and-out option which has a time to maturity of 0.5 year, strike price of \$6.5, underlier's volatility of 30%, dividend yield of 0%, and the risk free rate is 2%. Use the Matlab function written in (i) to compute C_{do}'s for $H =$ \$7 and values of S_0 being \$7 to \$10 in increments of \$0.1. Plot these values versus S_0. Repeat with H changed to \$6, and plot these new values versus S_0 on the same graph.

(iii) For the same set of parameters used in (ii), obtain the Black-Scholes price for European vanilla call and plot them on the same graph obtained in (ii). Comment on the relationship between the three sets of option prices.

(iv) Write a Matlab function which implements the binomial tree method for a European down-and-out option.

- With the same option parameters given in (ii) but fixing S_0 at \$8 and H at \$6, use this Matlab function to obtain approximate values of C_{do} for $N = 210, 211, \ldots, 250$. Denote these values by C_{do}^N.
- Use the Matlab function written in (i) to to obtain the exact value C_{do}.
- Hence compute the errors $C_{do}^N - C_{do}$, $N = 210, 211, \ldots, 250$.
- Plot these errors versus N. Comment on the plot obtained.

(v) Given the ith price level in the binomial tree contains all nodes with the same price values of $S_0 \exp(i\sigma\sqrt{\Delta t})$. What are the values of i which correspond to the two values of N yielding the locally minimum errors in (iv)?

Important: Besides the two .m files for the required Matlab functions, also prepare a Matlab script file (named `A1p1.m`) to contain all Matlab statements used to generate results in (ii), (iii) and (iv). This file will be executed during the grading of the assignment.

A1.2 (i) Write a Matlab function that implements version 1 of the single-state variable binomial tree method (see lecture notes) for approximating prices of a European floating strike lookback put option (newly issued) which has a time to maturity of 0.5 year. The underlier has a current price of \$1, volatility of 40%, dividend yield of 2%, and the risk free rate is 4%. Obtain option value estimates for number of time steps N being 200 to 20000 in increments of 200. Obtain a graph of these option values versus N. Comment on the plot obtained.

(ii) Write another Matlab function (a modification of the function in (i), see algorithm version 2 in lecture notes) for pricing European floating strike lookback put options which are not newly issued or with underliers having a running maximum greater than S_0. Test your implementation with the same option parameters and the numbers of time steps used in (i) but this time with a running maximum of \$1.3. Plot option values obtained versus N. Comment on the plot obtained.

Important: Besides the two .m files for the required Matlab functions, also prepare a Matlab script file (named `A1p2.m`) to contain all Matlab statements used to generate results in (i) and (ii). This file will be executed during the grading of the assignment.

Due date, requirement, guidelines and regulations

(i) The due date/time for the Matlab programmes and brief report is **2359hr on 8 October, 2017**. No late submission will be accepted.

(ii) Work on the assignment problems should commence soonest possible as programming and debugging can be time consuming.

(iii) Use **Matlab** for all programming tasks.
Please add suitable amount of comments to your codes and test your codes thoroughly. The first line of each Matlab m-file should have a comment line containing the names of the group members.

(v) Prepare your report in the Windows Word format or the PDF format with a description of your work done plus supporting figures and tables etc, as well as all necessary analysis and comments.

(vi) The .doc/.pdf and all .m files should all be archived in a single Zip/Rar file. Name your .zip/.rar file with your group index (such as Gxx_Assignment.zip or Gxx_Assignment.rar where Gxx is your assigned group index), and submit it online to the IVLE workbin set up for this purpose. In case of multiple uploads from a group, only the latest archive file will be used in the grading process.

(vi) This assignment counts 10% towards the final assessment score of this module.

(vii) Plagiarism (copying work from fellow students, groups or others) **would not be tolerated and all parties involved would be penalized severely**.
Please refer to `http://emodule.nus.edu.sg/ac/` for more information on NUS's disciplinary process on plagiarism.

⋈⋈ The End ⋈⋈