FI6121 Financial Modelling and Data Analytics AY 2019/20

Group Technical Assignment: Report and Presentation (40%)

Due: Sunday, 1st December 2019 5PM SHARP

(Sulis Submission with TurnitIn plagiarism check)

There are two parts to this project assignment, which is designed to test your theoretical understanding of, and ability to practically implement the Heston Stochastic Volatility and Local Volatility Models:

1. Use the Bloomberg local volatility surface (or matrix) to build a 1-month implied price distribution for the S&P500 market index as of a given historical reference date.

   Use this implied price distribution to derive an independent "at-risk" benchmark to corroborate what the level of the CBOE SKEW index is implying about the downside tail risk exposure to investors in the SPX market index.

   Your written report for this section of the assignment should be about 5 pages or so (not including Matlab code which should be presented as 'Appendix Material').

2. Implement a Monte Carlo pricing application of the Heston Stochastic Volatility model to price a 100%/130% 90-day up-and-out Call option[1] on the SPX index as of 13th August 2019.

   Use the same reference date as in Part 1 to fix your Heston parameters $\{\kappa, \theta, \rho, \sigma_v\}$ and instantaneous variance V_t for your Heston Monte Carlo simulations.

   Write a short (c. 3-page) analysis which summarises the ability of the Heston model to explain or account for the implied volatility skew in the SPX options market, and which explains any differences between this 'model price' with a) the Bloomberg Local Volatility 'market price', and b) the constant-volatility Black-Scholes 'model price'[2] .

[1] This is an example of a "barrier Call option" in which the strike price X equals the initial stock price (i.e. we say is ATM or the strike is 100% of the initial stock price), and in which the 'knock-out' barrier is set a level 30% above the initial stock price (i.e. the barrier is 130% of the initial stock price). IF at any time prior to expiry, the stock price touches the 130% level as defined, then the Call gets 'knocked-out' or gets terminated (typically ... but not always ... with zero payoff). Conversely, if the barrier Call option survives to maturity (i.e. stock price never touches the 130% barrier level), and finishes ITM but below 130%, then there will be a conventional European-exercise style payoff on the Call. Clearly, a 100%/130% up-and-out Barrier Call option must be cheaper than an otherwise equivalent 100% (i.e. ATM) European Call option ! Can you see why ?

[2] You can either re-use your Local Volatility Monte Carlo simulation code from Part 1. but with the finely-interpolated (explained below) Bloomberg local volatilities (St = K; t = T) replaced by a constant volatility. You should graph a plot of the BS MC barrier Call values as a function of a range of possible implied volatility values, to highlight the difference between the Bloomberg Local Volatility price and the Black Scholes Monte Carlo price (i.e. in which the path-dependency of the barrier Call can still be modelled).

Report

The report should address the requirements with the results presented with the aid of explained graphical visualisations, and should include rationalisations of the approaches taken.

- Matlab code should be included in the appendix.
- Submission through Sulis with TurnitIn plagiarism check
- Reference sources using UL Harvard referencing style.
- University of Limerick regulations on plagiarism applies.
- A late penalty of 1% per day applies.

Presentation

Present your report in a 10-minute (SHARP) presentation describing the data and descriptive/predictive modeling methodology, and outlining the rationale for the approach taken.

Presentations will take place from **Monday 2nd December at 2PM**. Prompt attendance is expected.

Grading Template

20 Meets Report Requirements
10 Technical Skills
10 Group Presentation: preparation, engagement, communication, projection, clarity, visuals, content and aural cohesion

Assignment Groups (Randomly chosen):

FI6032 Class Spring 2017 - Assignment			
FORENAME	**SURNAME**	**Number Chosen**	**Group for Assignment**
AMRIT ADIVEPPA	CHACHADI	10	1
CHEN	DENG	3	1
KRISHNA	MAHESHWARI	13	2
EOGHAN	O'SHEA	4	2
KEYU	WU	9	2
AIHUA	LIU	7	3
AKHIL	MENON	8	3
QIANG	MA	2	3
DARSHAN	GALA	5	4
NIALL	O'DONNELL	6	4
WUQIONG	YANG	12	4
YUZHEN	HAN	1	5
MERRY ELIZABETH	JONE	11	5

Assessment of Coursework – 40% Group Project (Individual Project if Fail by end August)

In order to reward individual endeavour as well as collective effort resulting from group synergies, your module tutor will grade each member of the group both collectively and individually. In the latter case, each member of the group will be required to provide a brief feedback synopsis and submit a 'peer grade' evaluation of the contribution made by each co-member of the project group - these grades will be confidential and will not be disclosed to other members of the group.

When allocating grades to individual members of the group, your supervisor / module tutor will in the first instance evaluate the overall quality of the group effort, but will additionally use the individual peer grading feedback as an important source of information in guiding their grading evaluation of the group research project.

Due to the introduction of this assessment protocol each member of the group will therefore not necessarily receive the same final grade for the group research project.

<u>Mandatory Contribution to Group Project (Coursework)</u>

Unless a student seeks an exemption from attending at module lectures (e.g. a linked-in student needing to clear a deficit grade in the end of term exam) and / or group work during semester, a student who attends at lectures:

- Will be deemed to have also made himself / herself available for assessment by both the coursework and the exam components of the module,
- Will be expected to attend at regular group meetings, and
- Will be expected to make a meaningful and equitable contribution to the research and completion of the group project.

Appendices

1) Building the 1-Month (c. 22-25 trading day) Implied Price Distribution

Today's Date: 05/11/2014

Index Value: 2023.22

Expiry	T (years)	ImpFwd	30.0%	32.5%	35.0%	…	165.0%	167.5%	170.0%
07/11/2014	0.00397	2022.04	0.5627	0.5746	0.5775	…	0.2247	0.2265	0.2286
14/11/2014	0.02381	2021.19	1.0146	1.0192	1.0208	…	0.1357	0.1367	0.1374
21/11/2014	0.04365	2020.21	0.9183	0.9115	0.9017	…	0.1325	0.1331	0.1335
28/11/2014	0.05952	2019.66	0.4922	0.4648	0.4432	…	0.0865	0.0865	0.0865
05/12/2014	0.07937	2019.00	0.5803	0.5505	0.5190	…	0.1022	0.1022	0.1022

The Excel screenshot from Sample_Surface.xlsx shows a 5 x 57 matrix[3] subset spanning an (approx.) 1-month trading time horizon of the Bloomberg Local Volatility table or matrix which was manually down-loaded into an Excel spreadsheet as of a reference date of 5th November 2014.

Since your Monte Carlo price simulation (*Equation 1*) over a 1-month (c. 22-25 trading days) "risk horizon" requires a daily (trading) time step (dt = 1/252) in order that the *Euler discretisation* term $\sqrt{\Delta t}\varepsilon(0,1)$ is approximately correct, the following code fragment illustrates how you need to finely interpolate the local volatility matrix/surface in the time dimension (i.e. daily time increments) into a 25 x 57 matrix of values. You do not need to interpolate in the strike dimension, as the 2.5% strike or moneyness increments used in the Bloomberg local volatility table are already sufficiently finely interpolated in the strike dimension.

Equation 1: GBM with stochastic volatility term included

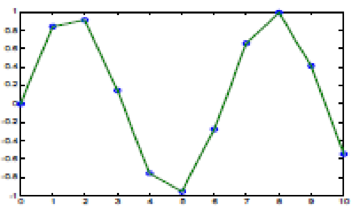
$$S_{t+dt} = S_t \, e^{\left(\mu - \frac{1}{2}\sigma^2(S_t,t)\right)dt + \sigma(S_t,t)dW_t}$$

$$S_t = K$$

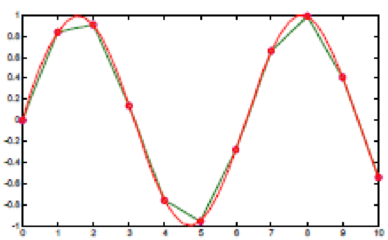
[3] 57 columns since the Bloomberg local volatility matrix/surface, which is imputed from a smoothed (in fact finely interpolated mesh) implied volatility surface, is arrayed column-wise in 2.5% strike increments, ranging from a minimum of 30% strike or moneyness value to 170%.

Example 1: First, demonstrate <interp1> by plotting a "coarse-granularity" sine curve, and then demonstrate how to implement a “fine-mesh" interpolation using the (cubic) "spline" function in Matlab. As this is precisely what you need to do in order to finely interpolate the local volatility matrix in the time dimension for Monte Carlo simulation of the underlying price or market index (SPX).

```
x=0:10;
y=sin(x);
xi=0:.25:10;
yi=interp1(x,y,xi,'linear');
plot(x,y,'o',xi.yi)
hold
```



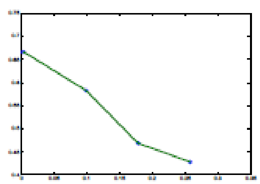
```
yi=interp1(x,y,xi,'spline');
plot(x,y,'o',xi,yi,'Color','red')
```



Example 2: Now, we demonstrate how to finely interpolate a fixed-moneyness skew (e.g. moneyness fixed at 30%) across the time horizon 16-Aug-13 through 15-Nov-13 before moving on a fine-mesh interpolation of the volatility matrix in the time dimension.

13/08/2013		**30**
Expiry	**T (years)**	**0.3**
16-Aug-13	0.0082	0.6671
20-Sep-13	0.1041	0.5833
18-Oct-13	0.1808	0.4688
15-Nov-13	0.2575	0.4284

```
sum(isbusday(busdays('08/16/13','11/15/13')));   = 65 trading days
% Day1=16Aug13, Day25=20Sep13, Day45=18Oct13, Day65=15Nov13
x=[1/252;25/252;45/252;65/252];  y=[0.6671;0.5833;0.4688;0.4284];
xi=1/252:1/252:65/252;  yi=interp1(x,y,xi,'linear');
plot(x,y,'o',xi,yi)
hold
```



```
    yi=interp1(x,y,xi,'spline');
plot(x,y,'o',xi,yi,'Color','black')
```

Note: You should be able to observe from this graphic that there is a non-trivial difference between the short-dated and longer-dated linear- and spline-interpolated local volatilities.

Also note that the local volatility 'matrix' created by this fine-granularity interpolation or smoothing algorithm is in fact a 65 x 1 'vector'.

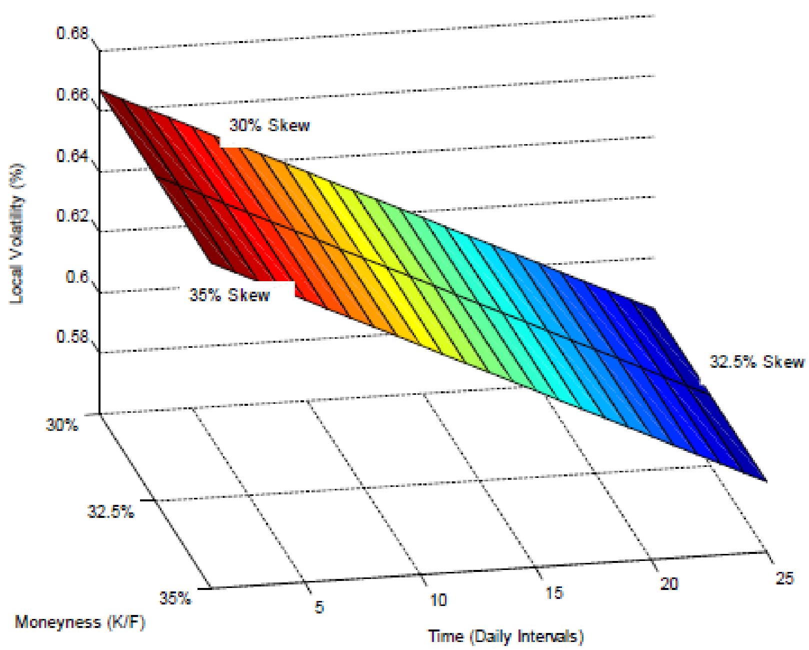
In the Monte Carlo simulation you will need to implement in order to build the 1-month (25-day) price distribution, you will need to "look up" the appropriate local volatility value (K = St) in a 25 x 57 'matrix' or table as you simulate your index across future times t and at various levels of the index S_t = K.

This means that although you will only need to implement a 'daily-granularity' interpolation across the 1-month time interval 16Aug13 - 20Sep13, you will need to do so for all 57 skews.

You can either do this manually (i.e. repeatedly or recursively replicate the above interpolation one skew at a time for all 157 moneyness skews listed in the Bloomberg local volatility matrix - i.e. 30%, 32.5%, 35% ... 170%), or you can automate the calculations by building on the sample code fragment overleaf and completing the pseudocode which follows:

Example 3 *Finely interpolating (daily granularity) across the 25-day 1-month time horizon (16-Aug-13 through 20-Sep-13) and across a subset of moneyness points (30% to 35%). To get the fine-mesh local volatility surface as in Table below, copy the following code fragement into Matlab, run and 'surface-plot' the resulting finely-interpolated local volatility matrix.*

```
listed_expiry_dates=[1/252;25/252];
lvmatrix=[0.6671 0.6671 0.6671;
0.5833 0.5833 0.5833];
simulation_dates=1/252:1/252:25/252;
interpolated_lvmatrix=interp1(listed_expiry_dates,lvmatrix,simulation_dates,'spline');
```



13/08/2013		30	32.5	35
Expiry	**T (years)**	**0.3**	**0.325**	**0.35**
16-Aug-13	0.0082	0.6671	0.6671	0.6671
20-Sep-13	0.1041	0.5833	0.5833	0.5833

Example 4 *By way of illustrating how to properly "look up" the appropriate volatility to use in your Monte Carlo price simulation $\sigma(S_t, t)$ you should bear in mind that since the Bloomberg "moneyness" matrix is defined as K/F you must (for the purpose of looking-up the appropriate local volatility value and in order to remain consistent with the latter's moneyness definition in Bloomberg) express your simulated index price S_t at time t as a ratio $\frac{S_t}{F(t)}$ where $F(t)$ denotes the current SPX forward price for the future time interval t, and* then *interpolate the local volatility skew for that maturity to find the corresponding-moneyness local volatility $\sigma(\frac{S_t}{F(t)}, t)$ from the finely-interpolated matrix illustrated in the graphic below.*

So for example, if the *simulated* index price on Day 10 $S_{t=10}$ equals 542.50 say (an extremely low value for the SPX index), and if the currently observable forward price for Day10 $F_0(t = 10)$ is say 1750, then we have that the current simulated index moneyness level $\frac{S_t=K}{F(t)} = 31\%$ As illustrated below[3], we then interpolate the fixed local volatility skew corresponding to Day 10 and to a 31% moneyness level - i.e. $\sigma\left(\frac{S_t=K}{F(t)} = 31\%, t = 10\right) = 63.22\%$ (an extremely high value for equity market volatility ... as we should expect to see) as circled (in red) in the local volatility table overleaf.

[3]This is a much simpler and easier to understand way of describing the exact interpolation of the *simulated-moneyness* local volatility value from adjacent local vols in the daily-interpolated Time x 2.5% increment Moneyness local volatility matrix - i.e. compared to the description on p.8 in the Lundkvist 'white paper' on Bloomberg's Monte Carlo Implementation of the Local Volatility Model

31% LocalVolMoneyness = *m*

Day \ Moneyness	30%	32.5%	35%
1	0.6671	0.6671	0.6671
2	0.6636	0.6636	0.6636
3	0.6601	0.6601	0.6601
4	0.6566	0.6566	0.6566
5	0.6531	0.6531	0.6531
6	0.6496	0.6496	0.6496
7	0.6462	0.6462	0.6462
8	0.6427	0.6427	0.6427
9	0.6392	0.6392	0.6392
10	0.6357	0.6357	0.6357
11	0.6322	0.6322	0.6322
12	0.6287	0.6287	0.6287
13	0.6252	0.6252	0.6252
14	0.6217	0.6217	0.6217
15	0.6182	0.6182	0.6182
21	0.5973	0.5973	0.5973
22	0.5938	0.5938	0.5938
23	0.5903	0.5903	0.5903
24	0.5868	0.5868	0.5868
25	0.5833	0.5833	0.5833

LocalVolMoneyness

Use this value to simulate from S_0 to S_1 if *m* = 30%

Use this value to simulate from S_{10} to S_{11} if *m* = 30%

LocalVolMoneynessi

= 31%

= simulations(i=10,:)./forwardPrice(i=10)

LocalVolSurface(i+1,:) fixed skew for simulations which start from Day i=10

lv = interp1(LocalVolMoneyness,LocalVolSurface(i+1,:),simulations(i,:)./forwardPrice(i),'spline','extrap');

= interp1(X,Y,Xi,'spline','extrap');

We then simulate forward over the next daily time interval as follows

$$S_{t=11} = S_{t=10}.e^{\{\mu-\frac{1}{2}x0.6322^2\}\frac{1}{252}+.6322\sqrt{\frac{1}{252}}\varepsilon(0,1)}$$

Corollary: This means that given a start date 16-Aug-13 and end date 20-Sep-13, you will need to record the currently-observable respective forward prices F(t =16-Aug-13) and F(t =20.Sep-13) on Bloomberg, and finely interpolate a daily-granularity forward curve between these two forward price values. I leave it to you to write the code to accomplish this.

Local Volatility "Look-up" Procedure

The following pseudo-code[5] (which you must insert in a for-loop for i = 1:25) provides a good deal of help in illustrating how to implement the above local volatility "look-up" procedure, without necessarily doing all the work for you.

NoSimulations=10000; N=NoSimulations;

% F=[F(1,1);F(2,1);...;F(25,1)] ... you must daily-interpolate a forward price vector spanning the start and end dates

forwardPrice=F;

% simulations = [ones(1,N).*S0; (25-1,N)] ... dimensions the 25x10000 simulated index price matrix, where first row = initial spot price of the index

dt=1/252; E=randn(25,N);

LocalVolMoneyness = 0.30:.025:3.0; % defines the moneyness range $\frac{S_t=K}{F}$ in the Bloomberg local volatility matrix

% LocalVolSurface = to be finely-interpolated in time dimension as a t=25 x 109 matrix of finely-interpolated local volatility values

% simulations(i,:) = the row vector of simulated index prices on Day 10 for all N=10,000 simulations (i.e. $S(i = t = 10, 1 : N)$)

% forwardPrice(i) = the interpolated forward price $F_0(t = 10)$ corresponding to future date $t = 10$

lv = interp1(LocalVolMoneyness,LocalVolSurface(i+1,:),simulations(i,:)./forwardPrice(i),'spline','extrap');

% lv is a 1 x N row vector, containing all the interpolated local vols from Day i which one uses in the N = 10000 simultaneous simulations to generate the index levels for Day i+1

% ... where simulations(i,:)./forwardPrice(i) = $\frac{S_t=K}{F(t)}$ = "moneyness-equivalent of the index level" for the purpose of looking up the local volatility from the Bloomberg moneyness table of same

% ... where LocalVolSurface(i+1,:) defines the 1x109 local volatility skew for a single day i+1 = 10+1 ...

% ... and lv returned is in fact a 1 x N=10000 (not 109) row vector

simulations(i+1,:) = simulations(i,:).*exp((mu-.5*lv.^2)*dt+lv.*sqrt(dt).*E(i,:));)

---

[5] This code only uses a single for-loop to simulate the index price across 25 daily time steps. In order to implement the simulations across 10,000 sample paths, the index "price" at each time-step is in fact represented as a 1 x 10,000 row-vector of index prices (rather than a single scalar value). The code then simultaneously simulates this 1x10000 row vector of prices (which implies a 1x10000 row vector of moneyness levels at time i) out to the next time step using the corresponding row-vector of random error terms E(i,:) where " : " means all 10000 columns in the simulations matrix, and using the corresponding+1 row-vector of local volatilities lv, where lv has the same 1x10000 row vector dimensions as the row vector of index prices (and moneyness levels).

Code to Filter Out Sample Paths in which the Knock-Out Barrier has been Hit

```
Call_Payoff = zeros(1,N);
Call_Payoff = max(0,simulations(end,:)-strike);
for m = 1:N
    if max(simulations(:,m)) > KO_Barrier == 1
        Call_Payoff(1,m)=0;
    else
        Call_Payoff(1,m)=exp(-r*T/252)*mean(Call_Payoff);
    end
end
```

Corroborate the CBOE Skew Index Implications for a > 3-Sigma Downside Event

The following was done to compute the implied price distribution probability of a greater than 3-sigma event occurring over the course of the next month (25-day risk horizon):

- Simulated (say) 10000 index prices out one month (i.e. the 25 day "risk-horizon")
- Generated a smoothed histogram of the risk-date returns using the <ksdensity> function (or otherwise), you can use the following code fragment (for a Standard Normal N(0,1) probability density distribution)

```
x=randn(100000,1);
f=ksdensity(x,-1.95,'function','cdf');
n=quantile(x,.025); % gives the value of the N(0.1) random variable
below which there is a 2.5% "tail probability" area - i.e. -1.96.
```

This fragment shows that there is a c. 2.5% probability of a 1.95 or greater downside (minus sign) sigma-event occurring in the case of a N(0,1) perfectly-symmetrical Gaussian distribution.

Based on your CBOE SKEW index value on your download date[6], you should then interpolate between the SKEW values listed in the Table below to impute what the SKEW index is implying about the probability of a 3-sigma or greater deviation below the mean of the implied price distribution.

[6] CBOE SKEW = 130.71 @ 14:15 on 01/11/19 (Source: www.cboe.com/SKEW)

Estimated Risk Adjusted Probability		
SKEW	S&P 500 30-Day Log Return	
	2 Std. Dev	3 Std. Dev.
100	2.30%	0.15%
105	3.65%	0.45%
110	5.00%	0.74%
115	6.35%	1.04%
120	7.70%	1.33%
125	9.05%	1.63%
130	10.40%	1.92%
135	11.75%	2.22%
140	13.10%	2.51%
145	14.45%	2.81%

Source: Chicago Board Options Exchange. "CBOE skew index–skew (2011)." URL: http://www.cboe.com/micro/skew/documents/skewwhitepaperjan2011.pdf