TTA-45046 Financial Engineering
Project Work 2019

April 4, 2019

UPDATES:

- April 04: Deadline for returning the project is May 12, 23:55 via Moodle.
- March 20: Typo. t_0 is February 25.
- March 18: Added comments and footnotes about log-returns.
- March 15: Fixed some typos, added some footnotes.
- March 14: First draft.

1 Introduction

The idea of the project work is to learn to price structured derivatives. The structured product to be priced is an actual trade made by Morgan Stanley Bank in 2019.

The programming environment is MATLAB or R. Python is possible as well, but a full support concerning problems related to the code will not be provided. You should write your code by yourself and you should comment it, so that others can easily get the idea what your code is doing. The code should consist of one "Main".m file. If you like to write your own functions, also separate .m-files for those are allowed. If necessary, there can be additional files for variables (e.g. a .mat file containing prices). Do the whole assignment using English only.

The project work must be returned no later than 12.05.2018 (23:59:59) via Moodle.

1.1 Points and grading

In order to get the maximum number of points, you must return your solution in time with no errors. Factual errors, coding errors, missing functionality and very poor coding style (e.g. blind use of loops) will reduce the points according to their severity.
The next three errors are critical since will affect the whole exercise: (i) mistake in computing returns[1] (50% reduction on the final grade), mistake in the volatility estimation (35% reduction on the final grade), mistake in estimating the correlation (20% reduction on the final grade) and (iv) any combination of the three (75% reduction, instinctively). Excel is allowed *only* for organizing the data (e.g. not for computing returns), a 30% reduction will apply otherwise.
The final grade will not be a purely arithmetic operation, but will consider the goodness of the implementation and the comments you make for each question: writing a well-organized and efficient code matters, providing good comments too. See section 8 for further details.
The pricing method is either antithetic Monte-Carlo for those students who are pursuing 8 points, or basic Monte-Carlo for those pursuing 6 points at most. This option applies to questions 1,3,4,5,6. See section 8.

If there are some major errors in your delivered code which prevents from accepting your work, it is possible to correct them according to the feedback given by the course assistant (martin.magris@tuni.fi) and get the project work accepted with lowered points. After you have submitted your work, you may be interviewed about it, especially if there are suspects that two or more students shared their codes and the assignment is not fruit of your own work.

[1] in their magnitude *and/or* sign: i.e. wrong implementation of section 6 point ii.

2 Objective

The task is to price the structured contract "Market-Linked Certificates of Deposit Based on the Average Value on Quarterly Averaging Dates of an Equally Weighted Basket Composed of the S&P 500 Index, the EURO STOXX 50 Index and the Hang Seng Index" developed by Morgan Stanley Bank. The contract was issued in February 2019 and expires in February 2024: this assignment will help you to understand the contract from a technical point of view and perhaps to decide whether is a good investment or not. You are required to answer the questions in section 8.

3 Terms of the contract

The contract is based on the following three indexes:

- S&P 500 Index (the "SPX Index"), ticker symbol: SPX
- EURO STOXX 50 (the "SX5E Index"), ticker symbol: SX5E
- Hang Seng Index (the "HSI Index"), ticker symbol: HSI

The three indexes constitute a basket, where the basket component weighting of each index is 33.33%. The pricing date of the contract is 25 February 2019 (t_0). You will therefore make all the evaluations based on this day, specifically (i) you will use historical data up to and including 25 February 2019 and (ii) price the contract on 25 February 2019 as well. Assume the the pricing, issuing and purchasing dates coincide (all equal to t_0), the contract expires (and pays) on t_{20} corresponding to 26 February 2024.

You enter the contract by paying its issue price (that you are later required to estimate), between the the issuing date t_0 and the expiry date t_{20} no interest are paid. The payment at maturity (P) consists in a cash payment of \$1000 plus the supplemental amount. The supplemental amount (S) payable at maturity for each contract equals the greater of (i) the product of (a) \$1000 by (b) the average basket percent change (B_r) and (ii) the minimum supplemental amount (S_m). The minimum supplemental amount is set to \$70:

$$P = \$1000 + S$$
$$S = \max\left(\$1000 \cdot B_r, S_m\right)$$

The average basket percent change is computed as:

$$B_r = \frac{C - B_{t_0}}{B_{t_0}} \tag{1}$$

where C is the basket closing value (determined at t_{20}) and B_{t_0} the basket initial value.
The basket initial value is set to 100. This means that the index prices at time t_0 need to be rescaled according to a multiplier m in such a way that the sum of the rescaled indexes adds up to 100 and the stocks are equally weighted (33.33%). The multiplier m is a 3-element vector $m = \left(m^{SPX}, m^{SX5E}, m^{HSI}\right)$ where $m^j = 33.33/I_{t_0}^j$[2] and $I_{t_i}^j$ represents the value of the $j-$th index at time t_i, $j \in \{SPX,\ SX5E,\ HSI\}$. The multiplier computed at t_0 remains constant for the term of the contract.
The basket closing value is computed as an average of the basket values on 20 different epochs (observation dates), t_i, $i = 1, ..., 20$. At each epoch the basket price B_{t_i} is recorded and at time t_{20} the closing basket price C is determined by considering their average:

$$C = \frac{1}{20}\sum_{i=1}^{20} B_{t_i} \tag{2}$$

The basket value B_{t_i} at each observation dates t_i, $i = 1, ..., 20$ is given by the product of the indexes values at t_i and their respective multiplier:

$$B_{t_i} = I_{t_i}^{SPX} m^{SPX} + I_{t_i}^{SX5E} m^{SX5E} + I_{t_i}^{HSI} m^{HSI} \tag{3}$$

For clarity, each B_{t_i} is unknown at t_0 and is a random variable based on the future values of the three indexes. Therefore, C, B_r and the payoff P are random as well.
The observation dates are: May 28, 2019 (t_1), August 26, 2019, November 25, 2019, February 25, 2020, May 26,

[2] E.g. The initial indexes values are \$20, \$30, \$70, the multiplier therefore is $m = (1.666, 1.111, 0.476)$, indeed each stock accounts for 33.33 (% and \$) of the initial basket value $B_{t_0} = \$100$.

2020, August 25, 2020, November 25, 2020, February 25, 2021, May 25, 2021, August 25, 2021, November 26, 2021, February 25, 2022, May 25, 2022, August 25, 2022, November 25, 2022, February 27, 2023, May 25, 2023, August 25, 2023, November 27, 2023, February 26, 2024 (t_{20}). When considering date differences, you either use Matlab's functions for time/date arithmetic, or you implement your own by assuming a year has (always) 365 days.

4 Stock Price model

In this work we assume that the value (price) of each index $I^j_{t_i}, i = 1, ..., 20$ follows a geometric Brownian motion, $j \in \{SPX, SX5E, HSI\}$:

$$dI^j_{t_i} = \mu_j I^j_{t_i} dt + \sigma_j I^j_{t_i} dW_{t_i}$$

where μ_j is a constant instantaneous expected return, $\sigma_j > 0$ is a constant volatility parameter, $I^j_{t_0} > 0$, the initial stock price, and W_{j,t_i} is a Wiener process. Under the risk neutral probability measure Q the price of index j follows the process

$$dI^j_t = rI^j_t dt + \sigma_j I^j_t dW^Q_t$$

where r is the risk free interest rate and $W^Q_{t_i}$ is the Wiener process under the risk neutral probability measure. For the sake of simplicity we assume that the risk free rate is positive and constant, and common for all the indexes. In the lecture and exercise classes, you learned how to simulate this process. Keep in mind that, since the three indexes are correlated, the random numbers (ϵ) used for the one-step-ahead simulation are not independent (see section 7).

5 Historical data

Historical closing prices for the stocks can be downloaded e.g. from "The wall street journal" web page[3]. The data will be used to estimate the volatility and return correlation. Make sure you download the correct data for the correct periods. You may use excel to organize the data and create one or multiple .csv/.xlsx files to be imported into Matlab. Use excel only to combine your downloaded-data into a single file (or better, do it straight in Matlab): do not compute log-returns nor volatility nor correlations in excel. Use closing prices, ignore other variables (of course, not timestamps).

6 Volatility

Volatility can be estimated for all the different indexes using the historical time series data. For the evaluations use the *most recent* price information[4]. For the estimation follow this procedure:

i Import the closing price data.

ii Calculate the daily log-returns[5]

$$u_i = \ln I_{t_i} - \ln I_{t_{i-1}}$$

for $i = 1, ..., m$, where m is the number of price observations.

iii The unbiased estimate of the volatility is obtained by dividing the standard deviation of the returns by the square root of the length of the time interval between the observations:

$$\hat{\sigma} = \frac{Std\,(u)}{\sqrt{\Delta t}}$$

For daily observations, $\Delta t = 1/252$. u is the vector collecting the log-returns: $u = \{u_1, u_2, ..., u_m\}$.

[3] e.g. `https://quotes.wsj.com/index/XX/XSTX/SX5E/historical-prices`

[4] since evaluating the contract in t_0, then download returns from the period 25 Feb 2014 to t_0 (both included). Keep in mind you have to answer question 6 as well.

[5] For clarity, "log-returns" (difference of log-prices) and "returns" (price difference over a period divided by the price at the beginning of the period) have here the same meaning. One-day returns are generally small (close to zero), it can be proved that: $\frac{P_{t_i} - P_{t_{i-1}}}{P_{t_{i-1}}} \approx \log P_{t_i} - \log P_{t_{i-1}}$, where P stands for "price". Commonly, and in this document as well, "returns" and "log-returns" are used with no distinction, as synonyms (since they are approximately the same). However, what you have to compute are actual "log-returns", as outlined in point ii (this is the common practice).

This is meant to be done for each index independently (so think each equation with a j superscript). A single function to be called three times is the most effective way to proceed. For the calculation of log-returns, do not use any for loop[6]. Note that there might be some missing values, depending from where you download the data. Use properly Matlab's "omitnan"-like options in the functions you call (or implement a turnaround by yourself -not suggested-) and do not blindly remove the days where one of the indexes is not available.

7 Correlation matrix

The indexes are correlated, and hence for the price simulation you need to estimate the correlation matrix of the (log)returns. Use the same historical data used for volatility estimation[7].

i Use matlab `corr` function to estimate the correlation matrix `R` in the log-returns of the 3 stocks (first create a $(m-1)$-by-3 log-return matrix from which to estimate `R`)

ii To generate three random draws under the correlation matrix `R`, implement (lecture 3):

```
epsilon = sqrtm(R)*randn(3,1)
```

Thus, `epsilon` is a vector of correlated (according to `R`) random numbers. To simulate one step ahead, `epsilon(1)` is used as the value of ϵ_1 for stock one, `epsilon(2)` as the value of ϵ_2 for stock two, and so on.

8 You are required to

1 Determine the expected payoff at t_{20} and the fair value of the contract at the issue date. Report the different results obtained by taking the data from the (i) last year (ii) last two years and (iii) last five years[8]. Comment how the different horizons for the historical data affect the evaluation. Print a histogram for the three cases *1.5 points.*

2 Determine the 0.025, 0.5, 0.95 quantiles of the payoff. Provide a comment. *0.5 points.*

3 Discuss how different weights in the basked affect the final payoff. First set to zero the weight of the Index with highest volatility and to one the others, then (i) assume the basked is made of the SP 500 index only. Comment on this, report the expected payoff and the multipliers in the two cases. *1 point.*

4 Make a plot of the expected payoff against different risk free rates (say $r = 0.005 : 0.005 : 0.3$). Make another plot of the expected payoff for different volatility values ($\sigma = 0.05 : 0.05 : 0.3$). Provide a comment. *1.5 points.*

5 The contract does not limit the bank exposure in case the average basket percentage change is high. Set a cap so that the contract never pays more than \$1300. What is the impact of this choice in the expected payoff an its 0.95 quantile? Is this a sensitive adjustment? *0.5 points.*

6 Find the value of the contract on 25 March 2019. To do so you will need to evaluate the expected payoff on March 25 2019, and use the historical data up to this day[9]. Comment your findings wrt. question 1. *1.5 points*

7 Now apply some inverse reasoning. First, reconstruct the (hypothetical) historical basked values. That is, the basket values that would have been observed in the past, based on the observed prices (use the same multiplier as before, so that at t_0 the index values is still 100 and each index accounts for 33%). Estimate the basked volatility from this (unique) time series, simulate future basket values[10] and determine its expected payoff (and its fair price) under this approach. Comment on the two methods and tell whether they should give the very same estimates or not. *1.5 points.*

8 Provide an overall comment on the contract. See section 9.4. *0.5 points.*

[6] This is a basic operation that you will likely be asked to do dozens of times when working at a company: make sure you do it in the best and most efficient way. This would be a way less than half-command-line.

[7] I.e. use the same estimation window for both (e.g. 5-years and 5-years).

[8] i.e. from 25 Feb 2014, 2017 and 2018.

[9] Question 1 is about evaluating the contract on t_0, here the evaluation is on a different date. Given that you answer this question after March 25, a small portion of the overall random index path from t_0 to t_{20} is observable and known (i.e. you can download the data from t_0 to 25 March 2019). So you can refine your initial estimate on the final payoff, and thus recover the contract value on 25 March 2019.

[10] Now you don't simulate three distinct paths, but directly simulate the values B_{t_i}.

Where not differently specified, (i) use the data from the last *five* years wrt. the evaluation date (ii) assume a market risk-free rate is 2.25%. Answer the questions by running at least 10^5 simulations, simulations required for questions 7 can be based on antithetic methods or not. Provide your comments to the questions within the same .mat file (by commenting the rows), max. 5 rows per comment.

Points are rescaled if no antithetic method is used[11]: 0.5 points (and multiples) become 0.375 (and multiples). The total sum of points is 8.5, however the maximum for the assignment is 8 in any case. The exceeding 0.5 points can help compensating other minor mistakes, imprecise comments, inefficient coding and so on.

9 Recommendations

9.1 Phase 1: Make a plan

Every good coding work start with planning. Write down, what your code should do. What is the first task, what the second and so on. How do you arrange the price data, how to simulate the prices and generate the random numbers, how to implement the dividend correction and so on. Think carefully of how to deal with date variables (if needed), if use tables, timetables, arrays.
Important: Read and understand *all* the questions first, then start coding. Understand what you are required to do on an overall level. E.g. to answer question 1 you will probably want to avoid copy-pasting the same code three times but maybe use a function. If you write a function, would you include a switch on the payoff formula to re-use the code for questions 1(iii) and 5? Be wise and think broadly in advance.

9.2 Phase 2: Implement basic functionality

You may find convenient to implement a "light" version of your code, to run with a reduced number of simulations, where you can easily check that everything is working fine. In this phase you have to double check that the construction of any basic functionality is smooth, correct and *fast*. At this point you may define basic functions to be reused in answering different questions.

9.3 Phase 3: complete your code

Once the basic aspects of your code are working correctly, implement more advanced features. Once you finished, review all you variable names (make sure the names are easy to understand) and nicely comment your code. Include calls for `clear var1,var2,...` to remove junk workspace variables as they become not longer necessary in the code. Make sure that (i) volatility and correlation matrix (ii) time epochs and time intervals and (iii) simulated prices are clearly saved and of easy access and interpretation. Refine the code by replacing whatever is unnecessary or slowing down the code: inconsiderate use of `for` loops for trivial operations will affect the grade.

9.4 Your conclusion

At the end of the code, create a new code section (start a line with `%%`) and give your own opinion about this contract, based on the results you've got during this work. You can just tell why this is such a contract you'd like to enter (or not)into? Also take the banks perspective. You can also describe how risky is this contract i.e. is there big variance in payoff and what are the risks this contract has? Be brief - max 10 sentences, just give your view of this contract mentioning anything you find relevant.

9.5 Informal review

It is possible to get comments about your code after you have done the phase 2. Any errors found (and fixed) at this phase will *not* lower your points. You can also ask comments about your code during the remaining weekly exercises. If you stuck with some problem, you can ask for a "bigger" help, by agreeing on a consequent lowering of the final score.

9.6 Some hints for coding

- When you start coding, use a small number of paths, less than 10, so that you can easily check that all you do makes sense.

[11] except for question 7, where it does not matter.

- Asses your code performance with (i) `tic toc` functions and **(ii)** "Run and Time" option. This helps a lot in improving your code performance[12].
- Use "breakpoints" to pause the code execution and enter/visualize the variable space within a function. In this way you check that everything is working or fix mistakes in the functions.
- Use significant and descriptive but compact variable names (e.g. use "logP" instead of "y" or "logPrice-Stock1Year" for "log-price").
- Aviod using `textscan` to import excel files, it's rather complex and takes a lot of time to get started with.
- Avoid for loops when possible, try to implement vectorized versions (`arrayfun`, `cellfun`, `structfun`). Use matrix operations when necessary.
- Make you code clear: try using *tables* and *timetables* when possible, avoid indexing matrices by row-column numbers use instead `Table.Properites.VariabeNames` (check documentation).
- Read about date functions and operations in Matlab, e.g. `timetables`, `datetime` - these can ease your work significantly. `datetime` is preferable over `datenum`.

9.7 Fair play

Every student has to write their own code. It is fine, though, to discuss your implementation and possible problems with others. Share thoughts, not code! Be prepared to orally present your project upon request.

9.8 Final Phase: Submit your work before 12.05.2019 23:55

You should return your work via Moodle. Submit only one zip-file, which contains all the necessary files your code needs to run, including the comments.
A well-commented code that produces clear output[13] (plus conclusion) is sufficient and no extra report is needed[14]. At the very beginning of your script insert a comment with your (i) name (ii) student ID (iii) email (iv) Maltab version you used.
Name the zip-file as xxx_ProjectWork2019.zip, where xxx is your student ID. Although the deadline for submission is rather late, you are encouraged to do the work before the exam, as doing this has shown a positive correlation with exam points.

[12]In Matlab, under the "Editor" tab, in the "Run" section, run the code by clicking the "Run and Time" button.
[13]Make sure it's clear which question(s) is(are) answered by each part of the code.
[14]E.g. no .pdf files for your comments.