# Columbia University
## MATH GR5260 Spring 2023
## Programming for Quant and Computational Finance
## Ka Yi Ng

## Exam 1
## Mar 10th 2023 (Fri) 6:15pm – 7:45pm

## GUIDELINES

This is an open book exam. You may use any notes, reference materials, internet, Jupyter Notebook or any Python IDEs during the exam period. However, you only have limited time to complete the exam, so you must use your time wisely.

**All answers must be written legibly.**
**In the Python code, add comments to your code where you think will help grader understand your logic.**

**Solution submission**
- Answer the theory questions on the exam paper provided.
- Submit your python code (and output results) as HTML or pdf or Jupyter notebook files onto Courseworks under 'Exam 1' section, in the same way you uploaded your homework previously. If you have technical issues in submitting your solution, please notify your TA or Prof Ng and email your solution directly to your TA to get a proper timestamp of submission.
- You may name these files whichever way you like.
- Make sure that in each of these files, you have your name and UNI on it.

During the exam, if you suspect of any typos or have questions, notify your TA or Prof Ng.

Prof Ng: kyn@math.columbia.edu
Ying Wu: yw3521@columbia.edu

| Question | 1 | 2 | 3 | Total |
|---|---|---|---|---|
| Points | 30 | 25 | 15 | 70 |

## HONOR CODE AFFIRMATION

I affirm that I will not plagiarize, use unauthorized materials, or give or receive illegitimate help on assignments, papers, or examinations. I will also uphold equity and honesty in the evaluation of my work and the work of others. I do so to sustain a community built around this Code of Honor. (https://www.college.columbia.edu/honorcode#:~:text=)

| First name | Last name | UNI | Signature |
|---|---|---|---|
| | | | |

## QUESTION 1 (30 POINTS)

In homework 1, log-linear interpolation was used to generate a discount factor curve using swap rates. The projected daily forward rates exhibit a piecewise constant curve.

In this question, we'll use a different interpolation method to investigate the shape of the implied daily forward rate curve.

Suppose the discount factors $DF_i$ for a set of dates $D_i$ $(0 \le i \le n)$ are given as follows.

| **Date** | 3/10/23 | 4/10/23 | 5/10/23 | 6/10/23 |
|---|---|---|---|---|
| **DF** | 1.0 | 0.99963 | 0.99563 | 0.99162 |

Let $t_i$ be the time in years from today to the date $D_i$ and let the zero rate for discount factor $DF_i$ be $r_i = -\ln(DF_i)/t_i$ for $0 \le i \le n$.

For any time $t$ where $t_i \le t < t_{i+1}$ $(0 \le i \le n)$, the interpolated discount factor for time $t$ is computed as:

$$DF(t) = e^{-rt}$$

$$r = r_i + (r_{i+1} - r_i)(t - t_i)/(t_{i+1} - t_i)$$

For $t > t_n$,

$$DF(t) = e^{-r_n t}$$

a) Write a python function to return the interpolated discount factor for a given time $t$. The function signature shall look something like this:

```
def get_df(t, df_times, df_values)
```

b) Use the example data and your python function to compute the discount factors for 100 calendar days (including 3/10/2023). Plot the computed discount factors.

c) Explain how you would compute the daily forward rate on a given time $t$ if you have the discount factors $DF_i$ for a set of times $t_i$ $(0 \le i \le n)$.

**Answer**

d) Write a python function to return the daily forward rate on a given time. The function signature shall look something like this:

```
def get_daily_forward_rate(t, df_times, df_values)
```

e) Use the example data and your python function to compute the daily forward rates for the next 100 calendar days (including 3/10/2023). Plot the computed daily forward rates.

f) Comment on the shape of the curve.

**Answer**

## QUESTION 2 (25 POINTS)

This question is about applying Monte Carlo simulation to the valuation of a European option on a basket of two stock prices.

Suppose that the two stock prices $S_1(t)$ and $S_2(t)$ follow a 2-dimensional correlated Geometric Brownian Motion (GBM). That is, $dS_1(t) = S_1(t)(\mu_1 dt + \sigma_1 dW_1)$ and $dS_2(t) = S_2(t)(\mu_2 dt + \sigma_2 dW_2)$ where $W_1(t)$ and $W_2(t)$ are correlated standard Brownian Motions with constant correlation $\rho$.

Here we assume that the drift and volatility parameters in the GBM and the zero rate $r$ are known constants.

Suppose the time to option expiry $T = 0.5$.

At option expiry time $T$, the payoff of the option is computed as the notional amount $AMT$ multiplied by a payoff rate $R(T)$ where $R(T)$ is the average of relative returns of the individual stock prices, capped at 10%, floored at 0%. Mathematically:

$$R(T) = \max(0.0, \min(10\%, (u_1(T) + u_2(T))/2\ ))$$

$$u_i(T) = \frac{S_i(T) - p_i}{p_i}, i = 0,1$$

$p_i$ is some fixed price for $S_i$ as a reference to compute the relative return $u_i$.

a) Write down the present value $V$ of the option as an expected value.

**Answer**

b) Write down the expression for $S_1(T)$ and $S_2(T)$ in terms of their initial prices, drift, volatility parameters and the bivariate standard normal random variable $(Z_1, Z_2)$ with correlation $\rho$.

**Answer**

c) Write a Python function to compute the simulated price $V_N$ of this option. The input parameters shall include: the number of simulations $N$, the notional amount, reference prices $p_i{}'s$, time to option expiry and the market data required. The function shall return **both** the simulated price and the 95% confidence interval. (You may use the input parameters without validating them within the function. You may also pass the standard normal random number generator as an input parameter)

Consider the following example data.

$S_1(0) = 120, S_2(0) = 80, \mu_1 = 1.5\%, \mu_2 = 1.5\%, \ \sigma_1 = \sigma_2 = 20\%, r = 5\%, \ \rho = 70\%$

$T = 0.5, AMT = 10000, p_1 = 100$ and $p_2 = 100.$

d) Use your python function to compute the simulated price $V_N$ for $N = 200000$ simulations.

We would like to know the impact of the correlation parameter on the simulated price.

e) Plot a graph of $V_N$ against correlation $\rho$ for $\rho = -1.0, -0.9, \dots, 0.9, 1.0.$

## Question 3 (20 Points)

This question is about using Pandas to handle P&L calculations for the FX spot transactions in a file.

**Download the .csv files from Courseworks under the Assignment section, 'Exam 1'.**

The file 2023_FX_trans.csv consists of a list of FX spot transactions.

| | TradeID | Date | BuySellCcy1 | Ccy1 | Amt1 | BuySellCcy2 | Ccy2 | Amt2 |
|---|---|---|---|---|---|---|---|---|
| **0** | 200171 | 2023-02-20 | Sell | EUR | 25000000 | Buy | USD | 23650000 |
| **1** | 200172 | 2023-02-20 | Buy | EUR | 35000000 | Sell | USD | 33057500 |
| **2** | 200173 | 2023-02-20 | Buy | EUR | 15000000 | Sell | USD | 14155500 |
| **3** | 200174 | 2023-02-21 | Buy | EUR | 10000000 | Sell | USD | 9439000 |

**Header descriptions:**
Date: date on which the transaction is traded
BuySellCcy1: buy/sell flag for Ccy1
Ccy1: primary currency of the currency pair
Amt1: amount in Ccy1
BuySellCcy2: buy/sell flag for Ccy2
Ccy2: secondary currency of the currency pair
Amt2: amount in Ccy2

The file 2023_EURUSD.csv keeps the closing prices of the EURUSD rate for the trading period.

| | Date | Open | High | Low | Close |
|---|---|---|---|---|---|
| **0** | 2023-02-20 | 0.93594 | 0.93707 | 0.93410 | 0.93594 |
| **1** | 2023-02-21 | 0.93610 | 0.93942 | 0.93496 | 0.93610 |

a) Write a Python program that reads the given input files and displays a DataFrame that keeps the following values, one record for each date.

| Column name | Description |
|---|---|
| Pos_EUR | Sum of EUR amounts of the transactions on date T |
| Pos_USD | Sum of USD amounts of the transactions on date T |
| CumPos_EUR | Cumulated sum of EUR amounts of all transactions up to and including date T |
| CumPos_USD | Cumulated sum of USD amounts of all transactions up to and including date T |
| EoDValue_USD | End-of-day value of all cumulated currency positions on date T<br>EoDValue_USD(T) = CumPos_EUR(T) x (Closing price of EURUSD on date T) + CumPos_USD(T) |
| PnL_USD | Daily P/L in USD on date T<br>PnL_USD(T) = EoDValue_USD(T) - EoDValue_USD(T-1)<br>Daily P/L in USD on the first date in file is equal to EoDValue_USD on the first date |

Here we assume zero transaction cost and date T-1 means the previous business day.

The first few rows will look something like this.

| | Pos_EUR | Pos_USD | CumPos_EUR | CumPos_USD | EoD_Value_USD | PnL_USD |
|---|---|---|---|---|---|---|
| **Date** | | | | | | |
| **2023-02-20** | 25000000.0 | -23563000.0 | 25000000.0 | -23563000.0 | -164499.3250 | -164499.3250 |
| **2023-02-21** | -22000000.0 | 20874600.0 | 3000000.0 | -2688400.0 | 119900.0180 | 284399.3430 |

The trader told you that the booking FX spot rate is problematic in some of the transactions and asked you to find them out. One possible approach is to set the lowest rate L and highest rate H in this trading period as the lower bound and upper bound of the range of reasonable rates.

Note: here booking rate of a EURUSD spot transaction is equal to the USD amount divided by the EUR amount.

b) Write a program to display the transaction records where the booking FX spot rate is greater than H or lower than L.

--end of exam—
-- Happy Spring Break --