Certificate in Quantitative Finance
Final Project

Bertrand Le Nézet

January 2014 Cohort

Abstract

This final project consists of two sub-projects: Basket Credit Default Swap (Section 1) and Interest Rate Derivatives (Section 2). Section 1 addresses the pricing of Basket Credit Default Swaps (a.k.a k^{th}-to-default Basket Credit Default Swap). We will see that correlation plays an important role in the pricing of these products (these products are generally traded by the correlation desk); we will therefore deep dive in the estimation of the correlation matrix and other input parameters before covering the sampling by copula (Gaussian copula and Student's t copula). Section 2 addresses the pricing of various IBOR[1] fixings linked derivatives (Bonds, Caps/Floors and swaptions) using the Heath–Jarrow–Morton framework. This framework relies on the estimation of historical forward rates volatility in order to simulate the evolution of the forward rates in the future. We will describe the principal component analysis (PCA) which will help us to identify the main factors which influence the volatility forward curve before covering the Monte Carlo simulation which projects the forward curve, compute the IBOR rates and price interest derivatives instruments.

Keywords: Basket Default Swaps, Monte Carlo method, Gaussian copula, Student t copula, Heath–Jarrow–Morton, Principal Component Analysis, Zero Coupon Bonds, Caps, Floors, Swaptions

[1] Interbank Offered Rate (IBOR) are daily reference rates based on the averaged interest rates at which banks offer to lend unsecured funds to other banks in interbank markets. LIBOR is the most famous reference but there are others (EURIBOR, SIBOR, etc.)

Contents

1 Basket Credit Default Swap

1.1 Introduction

A basket default swap is like an insurance contract that offers protection against the event of the k^{th} default on a basket of n ($n \geq k$) reference names. It is similar to a plain vanilla credit default swap but the credit event to insure against is the event of the k^{th} default. The spread (premium) is paid as an insurance fee until maturity or the event of the k^{th} default in return for a compensation for the loss. If the k^{th} default occurs, protection payment is triggered. We denote by $s^{k^{th}}$ the fair spread in a k^{th}-to-default swap, i.e, the spread making the value of this swap today equal to zero (Default leg = Premium leg). The most common type of basket default swaps is the first-to-default swap (FTDS), i.e $k = 1$ where the seller compensates the buyer for the loss on the reference name which defaults first.

We will study baskets with 5 reference names of 5 years maturity in order to keep the calculation fast; however the code that has been implemented can handle n references names.The appropriate default correlation and other input parameters are estimated from historical credit curves (survival probabilities and hazard rates; historical credit spreads are used as input to bootstrap historical credit curves).The fair spread of the k^{th}-to-default swap is calculated as an expectation over the joint distribution of default times and will be computed by sampling from copula (using pseudo-random and low-discrepancy numbers). We will explore two types of elliptical (Gaussian and Student's t copulas) and compare the results we obtain. We use a LIBOR curve as discount curve for the historical credit curve bootstrapping (used to estimate input parameters), the hazard rates bootstrapping and the basket spread calculation.

1.2 Model and hypothesis

1.2.1 Poisson process

Definition 1. Default indicator function

To model the arrival of a credit event we need to model an unknown random point in time $\tau \in \mathbb{R}_+$. We define the default indicator function (indicator function of the default event) to model default risk

$$I_{Default}(t) = 1_{\{\tau > T\}} = \begin{cases} 1 & \text{if } \tau < T \\ 0 & \text{if } \tau \geq T \end{cases}$$

Definition 2. Default survival function

We define also the survival indicator function $I_{Survival}(t) = 1 - I_{Default}(t)$

$$I_{Survival}(t) = 1_{\{\tau > T\}} = \begin{cases} 1 & \text{if } \tau > T \\ 0 & \text{if } \tau \leq T \end{cases}$$

Definition 3. Poisson process

We use a Poisson process characterized by its parameter λ(intensity of the process) to describe the default time of a company. The default time can be viewed as the first jump of a Poisson process. Based on the nature of the intensity function, the Poisson process can be classified as time homogeneous Poisson process or time inhomogeneous Poisson process.

- Poisson processes have no memory
- The inter-arrival times of Poisson processes $(\tau_{n+1} - \tau_n)$ are exponentially distributed
- Two or more jumps at exactly the same time have probability zero

Definition 4. Homogeneous Poisson process

A homogeneous Poisson process is defined as a process with stationary independent increments and initial value of zero. The times between two consecutive jumps are independently and identically distributed as an exponential random variable with mean $\frac{1}{\lambda}$ where λ is constant in time. If we define the default time τ as the first jump, the probability of an event can be expressed as:

$$P\left\{\tau \in [t, t+dt] | \tau \geq t\right\} = \lambda dt$$

Therefore, we can express the survival probability $P(t,T)$ between t and T as:

$$P(t,T) = \exp(-\lambda(T-t))$$

Definition 5. Inhomogeneous Poisson process

A inhomogeneous Poisson process is defined as a process with a non-negative deterministic and time varying intensity function $\lambda(t)$. In this case, we can derive the survival probability as:

$$P(t,T) = \exp\left(-\int_t^T \lambda(s)\, ds\right)$$

Remark. In this case the intensity (default hazard rate) depends on the time horizon T: we obtain a term structure of hazard rates when matching market prices.

1.2.2 Credit Default Swap pricing

There are two main families of model to price Credit Default Swap: the structural models and the reduced form models. Structural models are based on the complete knowledge of a detailed information set (e.g. Merton model 1974, Black and Cox 1976). In contrast, the reduced form approach (a.k.a. the "standard model" or J.P. Morgan approach in the literature) is based on the information set available in the market. In this section, we will describe the reduced form approach.

Definition 6. Mathematical setup

Let us suppose that there are N periods, indexed by $n = 1, \ldots, N$. Each period is of length Δt,expressed in years. Therefore, time intervals are:

$$\{(0, \Delta t), (\Delta t, 2\Delta t), \ldots, ((N-1)\Delta t, N\Delta t)\}$$

The end of period can be expressed as:

$$T_n = n\Delta t$$

Definition 7. Premium leg

We denote the N-period CDS spread as S_N,stated as an annualized percentage of the nominal value of the contract. We set the nominal value to 1. We assume that defaults occur only at the end of the period, so the premiums will be paid until the end of the period. Since the premium payments are made as long as the reference security survives, the expected present value of the premium leg is:

$$PL_N = S_N \sum_{n=1}^{N} D(0, T_n) P(T_n)(\Delta t_n)$$

where Δn is the year fraction corresponding to $T_{n-1} - T_n$ and $P(T_n)$ is the survival probability up to time T_n. This accounts for the expected present value of payments made from the buyer to the seller.

Definition 8. Default leg

The expected loss payment in period n is based on the probability of default in period n, conditional on no default in a prior period. This probability is given by the probability of surviving until period $n-1$ and then defaulting in period n $(P(T_{n-1}) - P(T_n))$. Therefore, the expected present value of loss payments is:

$$DL_N = (1-R) \sum_{n=1}^{N} D(0, T_n)(P(T_{n-1}) - P(T_n))$$

Definition 9. Fair spread

The fair pricing of the $N-$period CDS, i.e. the fair quote of the spread S_N, must be such that the expected present value of payments made by buyer and seller are equal, i.e. $PL_N = DL_N$. Thus we obtain

$$S_N = \frac{(1-R)\sum_{n=1}^{N} D(0,T_n)(P(T_{n-1}) - P(T_n))}{\sum_{n=1}^{N} D(0,T_n)P(T_n)(\Delta t_n)}$$

1.2.3 Bootstrapping survival probabilities and hazard rates

We have described previously how to calculate the spread S_n from the survival probabilities $P(T_n)$ with $n \in [1, N]$ and the discounting curve. The discounting curve is quoted in the market (bond markets, money market, etc.) but the survival probabilities are not quoted in the market. The participants in the Credit Default Swap market (Over-the-counter market) are quoting CDS spread instead (i.e. S_n are known but $P(T_n)$ are not known). This is a typical example of an inverse problem that we encounter often in quantitative finance. As in other similar situations, we will calculate the survival probabilities $P(T_n)$ with $n \in [1, N]$ recursively. i.e. We will start by find $P(T_1)$ which will help us to find $P(T_2)$ which will help us to find $P(T_3)$, etc. This is a bootstrapping process.

Definition 10. First step for survival probabilities

For $n = 1$, we obtain the following equation after a couple of algebraic manipulation:

$$\begin{aligned} PL_1 &= DL_1 \\ S_1 D(0,T_1)P(T_1)\Delta t_1 &= (1-R)D(0,T_1)(P(T_0) - (P(T_1)) \\ &\dots \\ D(0,T_1)P(T_1)(S_1\Delta t_1 + L) &= LD(0,T_1)P(T_0) \\ P(T_1) &= \frac{L}{L + \Delta t_1 S_1} \end{aligned}$$

with R the recovery rate and L the loss given default defined as $L = 1 - R$

Definition 11. Subsequent steps ($n = N$) for survival probabilities

The general expression is given by this formula

$$P(T_N) = \frac{\sum_{n=1}^{N-1} D(0,T_n)\left[(1-R)P(T_{n-1}) - ((1-R) + \Delta t_n S_N)P(T_n)\right]}{D(0,T_N)((1-R) + \Delta t_N S_N)} + \frac{P(T_{N-1})(1-R)}{((1-R) + \Delta t_N S_N)}$$

Remark. We can clearly see that$P(T_N)$ depends on $P(T_{N-1})$. This explains why we need to apply a bootstrapping process.

Definition. Hazard rates (non-cumulative)

We can calculate the non-cumulative hazard rate λ_n from the two survival probabilities $P(T_n)$ and $P(T_{n-1})$. We have seen earlier the relationship between the survival probability and the hazard rate in the case of an inhomogeneous Poisson process. If we assume the hazard rate function to be piecewise constant:

$$P(0,T_n) = \exp(-\int_0^{T_n} \lambda_s \, ds) = \exp(-\sum_{i=1}^{n} \lambda_i \Delta t_i)$$

From the previous equation, we can deduce that

$$\begin{aligned} P(0,T_n) &= \exp(-\sum_{i=1}^{n-1}\lambda_i \Delta t_i - \lambda_n \Delta t_n) \\ &= \exp(-\sum_{i=1}^{n-1}\lambda_i \Delta t_i) \times \exp(-\lambda_n \Delta t_n) \\ &= P(0,T_{n-1}) \times \exp(-\lambda_n \Delta t_n) \end{aligned}$$

This automatically give the expression that expresses the hazard rate as the "log" difference of two survival probabilities:

$$\lambda_n = -\frac{1}{\Delta t}\log(\frac{P(0,T_n)}{P(0,T_{n-1})}) \tag{1.1}$$

1.2.4 Marginal default times

The calculation of the basket spread that is part of the sampling from copula procedure[2] requires default times. The procedure will

- First generate a vector of correlated uniform variable $(u_1, \dots u_n)$ where n is the number of reference names in the basket CDS
- Then convert this vector of correlated uniform variables $(u_1, \dots u_n)$ into a vector of default times

This section explains how the exact default time can be calculated using the hazard rates (non-cumulative) term structure of a given reference name.

For a Poisson process, the probability distribution of the waiting time until the next event (i.e. default in our case) is an exponential distribution. We can therefore use the Poisson CDF ($F(\tau,\lambda)$)to convert τ to a probability $u \in [0,1]$:

$$u = F(\tau;\lambda_\tau) = \begin{cases} 1-\exp(-\lambda_\tau \tau) & \text{when} \tau \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

By inverting the above equation, we obtain:

$$\tau = F^{-1}(u;\lambda_\tau) = -\frac{\log(1-u)}{\lambda_\tau} \text{for u}\in[0,1]$$

We have assumed hazard rates to be piecewise constant when we bootstrapped the credit curve; therefore we don't have the exact value λ_τ (hazard rate $\lambda$for a given value of τ). We will therefore need to apply a procedure to estimate the exact default time τ for a given uniform variable u:

1. Find the year of default (i.e. find n so that the default occurs between $t_{n-1}$and t_n
2. Estimate the exact default time δt within the year of default

Definition 12. Year of default

We iterate through the term structure of hazard rates by summing up λ_i i until the inequality begins to hold:

$$\tau = \inf\left\{t > 0 : \log(1-u) \geq -\sum_{i=1}^{t}\lambda_i \Delta t_i\right\}$$

Where inf is the mathematical operator that represent the infinimum of a set

Remark. If the inequality holds after adding λ_n, then it means that $t_{n-1} < \tau < t_n$

[2]Sampling from copula will be described in detail later in this document

Definition 13. Exact default time

We define the default time τ as $\tau = t_{n-1} + \delta t$ where τ_n is the year of default and δt is the "year fraction" within the year of default. From the previous formula, we can calculate δt as:

$$\delta t = -\frac{1}{\lambda_n} \log(\frac{1-u}{P(0,t_{n-1})})$$

1.2.5 Join distribution

The most important problem in the pricing of basket credit default swaps is the modeling of the joint default times. The marginal distribution of a random variable X is defined by its CDF:

$$F(x) = \Pr(X \leq x)$$

The joint distribution function of two random variable $X_1$and $X_2$is:

$$F(x_1, x_2) = \Pr(X_1 \leq x_1, X_2 \leq x_2)$$

The joint distribution of n random variables $X_1, \ldots, X_n$ is:

$$F(x_1, \ldots, x_n) = \Pr(X_1 \leq x_1, \ldots, X_n \leq x_n)$$

We can model the default risk of our basket CDS if the joint default distribution function is known:

$$F(t_1, \ldots, t_n) = \Pr(\tau_1 \leq t_1, \ldots, \tau_n \leq t_n)$$

where $\tau_i \leq t_i$means a default event on the i^{th} reference name.

A multivariate joint distribution is a multidimensional analytical construction ($F(x_1, \ldots, x_n) = \Pr(X_1 \leq x_1, \ldots, X_n \leq x_n)$ that is hard to work with, especially in high dimension. The better approach to isolate dependence structure among multiple variables is a copula. Copula approach separates the joint distribution into two parts:

- the marginal distributions for each variable
- and their dependance structure

The joint distribution function of n uniform random variables $U_1, \ldots, U_n$is the copula function:

$$C(u_1, \ldots, u_n; \Sigma_\rho)$$

where the dependance structure is defined by a correlation matrix Σ_ρ:

$$\Sigma_\rho = \begin{pmatrix} 1 & \sigma_{i,j} & \sigma_{i,j} \\ \sigma_{i,j} & \ddots & \sigma_{i,j} \\ \sigma_{i,j} & \sigma_{i,j} & 1 \end{pmatrix}$$

Remark. Equicorrelation between all reference names is sometimes assumed, in this case we have a single parameter $\forall i \forall j, \sigma_{i,j} = \rho$ that defines the dependance structure.

Definition 14. Copula as a joint distribution

In order to obtain the expression for the copula function in terms of u_i, we transform a random variable by its own CDF. For the random variables $X_1, \ldots, X_n$ with a flexible choice of marginal distribution $F_1(x_1), \ldots, F_n(x_n)$, we have:

$$\begin{aligned} C(u_1, u_2, \ldots, u_n) &\equiv \Pr(U_1 \leq u_1, \ldots, U_n \leq u_n) \\ &= \Pr(F_1(X_1), \ldots, F_n(X_n)) \\ &= \Pr(X_1 \leq x_1, \ldots, X_n \leq x_n) \\ &\equiv F(x_1, \ldots, x_n) \end{aligned}$$

Definition 15. Sklar theorem

Sklar proved that for every multivariate CDF F with marginals F_i, there exist some copula function C such that:

$$F(x_1, x_2, \ldots, x_{n)} \equiv C(u_1, u_2, \ldots, u_n)$$

and if the joint distribution is continuous then the copula function C is unique.

Remark. Sklar theorem shows that we can obtain and rely on the copula function to model dependence structure

Definition 16. Multivariate Gaussian copula

The multivariate Gaussian Copula function can be expressed as

$$C(u_1, u_2, \ldots, u_n) = \Phi_n(\Phi^{-1}(u_1), \ldots, \Phi^{-1}(u_n); \Sigma)$$

where $\Phi_n$is the CDF for the multivariate standard normal distribution. There is no closed-form solution for this copula function

Definition 17. Multivariate Student's t copula

The multivariate Student's t Copula function can be expressed as

$$C(u_1, u_2, \ldots, u_n) = T_\nu(T_\nu^{-1}(u_1), \ldots, T_\nu^{-1}(u_n); \Sigma)$$

Student's t copula density function is

$$c(U; \nu, \Sigma) = \frac{1}{\sqrt{|\Sigma|}} \frac{\Gamma(\frac{\nu+n}{2})}{\Gamma(\frac{\nu}{2})} \left(\frac{\Gamma(\frac{v}{2})}{\Gamma(\frac{\nu+1}{2})} \right)^n \frac{\left(1 + \frac{T_\nu^{-1}(U') \sum^{-1} T_\nu^{-1}(U)}{\nu}\right)^{-\frac{\nu+n}{2}}}{\prod_{i=1}^n \left(1 + \frac{T_\nu^{-1}(u_i^2)}{\nu}\right)^{-\frac{\nu+1}{2}}}$$

where $U = (u_1, \ldots, u_n)$

1.2.6 Sampling from copula

Sampling from copula is a method to structure Monte-Carlo simulation. For each simulation, do the following:

- Generate a vector $(z_1, \ldots, z_n)$ of n uncorrelated random normal variables (n is the number of reference name is the basket CDS)
- Impose correlation on the vector $(z_1, \ldots, z_n)$ using a correlation matrix and obtain the vector $(x_1, \ldots, x_n)$
- Use the Normal / Student's t CDF to transform the vector $(x_1, \ldots, x_n)$ to a vector $(u_1, \ldots, u_n)$ of variables uniformly distributed over $[0, 1]$
- Convert the vector $(u_1, \ldots, u_n)$ to a vector of default times $(\tau_1, \ldots, \tau_n)$ using the hazard rate term structure of each reference name (i.e. use the hazard rates term structure of the reference name n to convert u_n into τ_n)

The benefit of the copula method is the separation between

- the dependance structure and
- marginal distribution for the default time τ_i

1.2.7 Correlation matrix for Gaussian copula

Gaussian copula require linear correlation. Linear correlation is always estimated from normal variables. The following steps explain how to estimate the correlation matrix that can be used in sampling from Gaussian copula:

1. Transform data $\mathbf{X}$ into uniform $\mathbf{U} = \hat{\mathbf{F}}(\mathbf{X})$, where $\mathbf{U}$ is properly scaled $\mathbf{U} \in [0, 1]$and $\hat{\mathbf{F}}(\mathbf{X})$ is the empirical cumulative density function (CDF)
2. Apply inverse Normal CDF $\mathbf{Z} = \mathbf{\Phi^{-1}}(\mathbf{U})$
3. Calculate the linear correlation (Pearson correlation) matrix $\hat{\Sigma} = \text{corr}(\mathbf{Z})$

Remark. Correlating data by a linear measure (Pearson) assumes the data is Normally distributed

1.2.8 Correlation matrix for Student's t copula

Student's t copula requires linear correlation (this is true for all elliptical copulas). The following steps explain how to estimate the correlation matrix that can be used in sampling from Student's t copula:

1. Transform data matrix$\mathbf{X}$ into uniform using$\mathbf{U} = \hat{\mathbf{F}}(\mathbf{X})$, where $\hat{\mathbf{F}}(\mathbf{X})$ is the empirical cumulative density function (CDF)
2. Calculate the correlation matrix $\hat{\sum}_K$ by applying Kendall's tau (rank measure) to U
3. Calculate the linear correlation matrix $\hat{\Sigma}$by applying the formula $\rho = \sin(\frac{\pi}{2}\rho_K)$ on each of the element ρ_K of the matrix$\hat{\sum}_K$

1.2.9 Maximum likelihood estimation for Student's t copula

The maximum likelihood estimation (MLE) for Student's t copula done by summing up contributions of daily/weekly observations to the log-likelihood. We maximize the following quantity with regards to ν:

$$\arg\max_\nu \left\{ \sum_{t=1}^{T} \log c(U_t^{Hist}; \nu, \Sigma) \right\}$$

where the function $c(U; \nu, \Sigma)$ is the Student's t copula density function that we have seen earlier. The MLE is a two-step procedure that is repeated for each value of ν from 1 to 25:

1. Calculate the log of Student's t copula density function using the input of $T_\nu^{-1}(U')$, where U' is a row vector of $1 \times n$, a set of observations for n reference names
2. Add to the sum of the log-likelihood for the previous days. Continue until $T = \text{NumberObservations}$

1.2.10 Sampling from Gaussian copula

The detailed procedure for sampling from Gaussian copula is as follows:

- Compute decomposition of correlation matrix $\hat{\sum}_{Gaussian} = AA'$ using Cholesky (if matrix is positive definite)
- For each simulation, do the following:
  - Generate a vector $Z = (z_1, \ldots, z_n)$ of n uncorrelated random normal variables (n is the number of reference name is the basket CDS)
  - Impose correlation on the vector $Z = (z_1, \ldots, z_n)$ using the correlation matrix A and obtain the vector $X = AZ = (x_1, \ldots, x_n)$
  - Use the Normal CDF to transform the vector $X = (x_1, \ldots, x_n)$ to a vector $U = (u_1, \ldots, u_n)$ of variables uniformly distributed over $[0, 1]$

  - Convert the vector $U = (u_1, \ldots, u_n)$ to a vector of default times $\tau = (\tau_1, \ldots, \tau_n)$ using the hazard rate term structure of each reference name (i.e. use the hazard rates term structure of the reference name n to convert u_n into τ_n)
  - Calculate the default leg
  - Calculate the premium leg

- Obtain the expectation of the default leg $< DL >$ by averaging the default leg values calculated across all simulations
- Obtain the expectation of the premium leg $< PL >$ by averaging the premium leg values calculated across all simulations
- Calculate the expectation of the fair spread as

$$\mathbb{E}^{\mathbb{Q}}\left[s^k\right] = \frac{< DL >}{< PL >}$$

1.2.11 Sampling from Student t copula

The detailed procedure for sampling from Student's t copula is as follows:

- Compute decomposition of correlation matrix $\hat{\sum}_{Studentt} = AA'$ using Cholesky (if matrix is positive definite)
- For each simulation, do the following:
  - Generate a vector $Z = (z_1, \ldots, z_n)$ of n uncorrelated random normal variables (n is the number of reference name is the basket CDS)
  - Draw an independent chi-squared random variable $s \sim \chi^2_\nu$
  - Compute the n-dimensional vector Student's t vector $Y = \frac{Z}{\sqrt{\frac{s}{\nu}}}$
  - Impose correlation on the vector $Z = (z_1, \ldots, z_n)$ using the correlation matrix A and obtain the vector $X = AY = (x_1, \ldots, x_n)$
  - Use the Student's t CDF to transform the vector $X = (x_1, \ldots, x_n)$ to a vector $U = (u_1, \ldots, u_n)$ of variables uniformly distributed over $[0, 1]$
  - Convert the vector $U = (u_1, \ldots, u_n)$ to a vector of default times $\tau = (\tau_1, \ldots, \tau_n)$ using the hazard rate term structure of each reference name (i.e. use the hazard rates term structure of the reference name n to convert u_n into τ_n)
  - Calculate the default leg
  - Calculate the premium leg
- Obtain the expectation of the default leg $< DL >$ by averaging the default leg values calculated across all simulations
- Obtain the expectation of the premium leg $< PL >$ by averaging the premium leg values calculated across all simulations
- Calculate the expectation of the fair spread as

$$\mathbb{E}^{\mathbb{Q}}\left[s^k\right] = \frac{< DL >}{< PL >}$$

1.2.12 Spread Calculation

Theory

The par spread of the k^{th}-to-default swap is derived by equating the default leg to the premium leg under the risk-neutral measure:

$$\begin{aligned} < DL > &= (1-R) \times NP \sum_{i=1}^{m} Z(0,t_i)(F_k(t_i) - F_k(t_{i-1})) \\ < PL > &= s^k \times NP \times \Delta t \sum Z(0,t_i)(1 - F_k(t_i)) \end{aligned}$$

where m is the number of periods

We can therefore calculate the fair spread of the k^{th}-to-default basket CDSs^k as:

$$s^k = \frac{< DL >}{< PL >} = \frac{(1-R) \times NP \sum_{i=1}^{m} Z(0,t_i)(F_k(t_i) - F_k(t_{i-1}))}{NP \times \Delta t \sum_{i=1}^{m} Z(0,t_i)(1 - F_k(t_i))}$$

However the joint distribution for k^{th}-to-default time across all reference names $\tau_k \sim F_k(t_1, t_2, \ldots, t_n)$ is unknown.

The alternative to calculate the fair spread is to use the Loss function as an expectation over the join distribution $L_k = \mathbb{E}\left[F_k(t)\right]$

The fair spread can then be calculated as

$$s^k = \frac{(1-R) \times \sum_{i=1}^{m} Z(0,t_i)(L_i - L_{i-1})}{NP \times \Delta t \sum_{i=1}^{m} Z(0,t_i)(NP - L_i)}$$

The fair spread calculation is carried out multiple times and averaged at the end (Monte-Carlo method).

Practical implementation

I have taken the following assumptions for the pricing of k^{th}-to-default basket CDS:

- Notional principal NP is equal to 1
- Discrete-time annual payments for the premium leg
- Continuous-time payment for the default leg
- Reference entities that have defaulted before k^{th} default are removed from the basket; each default will reduce the value of the portfolio by $\frac{1}{n} \times NP$ (n being the number reference names in the basket)

1.3 Main result

1.3.1 Correlation matrix for Gaussian copula

One of the most important step in pricing Basket Credit Default Swap is the estimation of the correlation matrix. The correlation matrix is generally obtained through historical sampling. Several ways to estimate the Gaussian correlation matrix have been researched:

- Difference of default probabilities vs. log difference of default probabilities
- Hazard rates vs. difference of hazard rates vs. log difference of hazard rates
- Stock log returns
- Weekly observations vs. daily observations

Credit entity	Ticker	Tier	Doc Clause	Rating	Sector
Bristol-Myers Squibb	BMY	Senior Unsecured	XR	A	Healthcare
Thomson Reuters	TRI	Senior Unsecured	XR	BBB	Consumer Services
Hewlett-Packard	HPQ	Senior Unsecured	XR	BBB	Technology
Int. Business Machines	IBM	Senior Unsecured	XR	AA	Technology
Pfizer	PFE	Senior Unsecured	XR	AA	Healthcare

Table 1: Credit entities used for Basket CDS Pricing

Methodology

Historical credit spreads have been obtained from Markit for year 2011 until today[3]. We are looking at 5 credit entities which are all corporates head-quartered in the United States. The below table (Table 1)[4] gives more details on the CDS contracts we have worked with. The restructuring clause (Doc Clause) stipulates whether restructuring is considered as a credit event or not. "XR" means "No restructuring". All CDS have been chosen with the restructuring clause "XR" because these contracts are considered as the most liquid in the United States.

Historical yield curves (discount factors) have been obtained from Bloomberg for year 2011 until today. I have only snapped one curve per week (Fridays) to reduce the amount of market data required. This seems to be a reasonable simplification considering the movements that can be observed in the yield curve over short period of time (one week) and considering the sensitivity of credit curves to interest rates.

Historical credit curves (probability of default, non-cumulative hazard rates) are bootstrapped using the historical credit spreads and the historical yield curves. The "standard model" (described earlier) is used bootstrap the credit curve. The CDS payment frequency is assumed to be 1 (annual payments). Linear correlations (Pearson correlation) is then estimated on changes (difference or log difference) in the variable (default probability or hazard rates) or on stock log returns. 5Y maturity point has been chosen as reference point for default probability and hazard rates (i.e. hazard rates between 4Y and 5Y) because the 5Y CDS are generally the most liquid contracts.

The process to estimate the correlation matrix for Gaussian copula is the one summarized in the "Model and hypothesis" section.

Historical stock prices (closing prices) have been obtained from Yahoo Finance.

Remark. I have tried several option to obtain the empirical CDF $\hat{\mathbf{F}}(\mathbf{X})$: the function density() which computes kernel density estimates combine with a trapezoid integration and the function ecdf() which returns a empirical cumulative distribution step function. The below smoothing test (Figure 1.1) shows that the function ecdf() works best in our case.

Default Probabilities

We have worked with 2 data sets of historical market data:

Rate set A 1 year of daily CDS spreads between 29/04/2013 to 30/04/2014

Rate set B 2 years of daily CDS spreads between 30/04/2012 to 30/04/2014

Daily 5Y default probabilities are obtained through the bootstrapping process. Differences (or log differences) are calculated for each day (except for the first one), finally this data is transformed to standard normal data using the empirical CDF (Pearson correlation takes normally distributed data as input).

Remark. It occurs that due to extreme market movements, some differences (or log differences) are so large (outliers) that their CDF is calculated as 1 when converting data to uniform (i.e. conversion from X to U) using the empirical CDF $\hat{\mathbf{F}}$. These "1" are then converted to Infinity when we apply the inverse standard normal (i.e $\mathbf{Z} = \mathbf{\Phi^{-1}}(\mathbf{U})$). These infinite values cause issues while calculating the Pearson correlation; I have excluded them. Since we are interested in the base correlation (long term correlation),

[3] July 2014

[4] Market data as of 30/04/2014

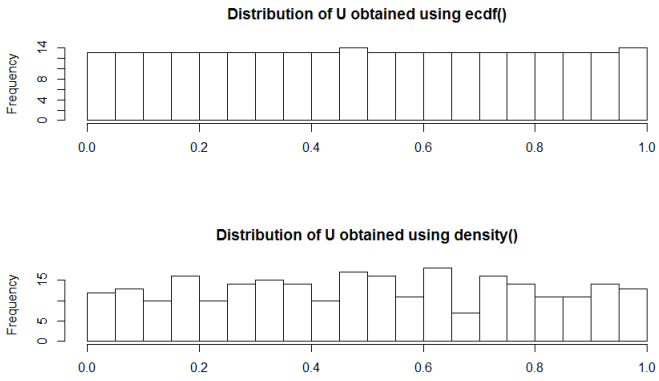

Figure 1.1: Two ways to transform data to uniform pseudo-samples - $\hat{\mathbf{F}}(\mathbf{X})$

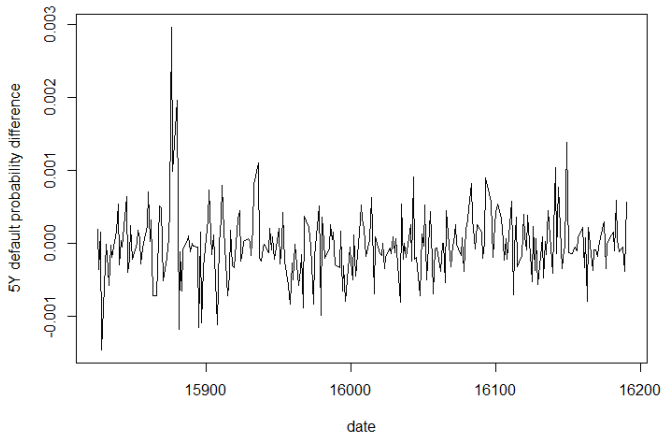

Figure 1.2: Bristol-Myers Squibb - 5Y default probability difference (Date origin in R is 01/01/1970)

it seems acceptable to exclude a few extreme values while estimating the correlation matrix. The figure (Figure 1.2) shows one of the extreme market movements for Bristol-Myers Squibb on 20/06/2013 (15876 on the diagram).

As we have seen earlier, we have to convert our data to normal before we can calculate the linear correlation. We will get better results if our original data follows a distribution that is close to the normal distribution. The figure (Figure 1.3) plots the kernel density of differences and log differences of 5Y default probabilities. We can observe that both distributions are relatively close from the normal distribution.

The correlation matrices estimated using 5Y Default Probability differences and the 5Y Default Probability log differences are very close (See **Rate set A** and **Rate set B** in Table 2). The correlations we obtain are all between -5% and 50%; must of them are positive or close to zero. The two highest correlations we observe are for the pairs Bristol Myers Squibb/Pfizer and HP/IBM. This makes senses these companies belong to the same industry sector: Healthcare for Bristol Myers Squibb/Pfizer and Technology for HP/IBM. I have deliberately chosen a fifth corporate that belong to a third industry sector (Thomson Reuters belong to Consumer Services). We can see that Thomson Reuters has very small correlation with the 4 other corporates.

Correlations observed over 1 year (**Rate set A**) and over 2 years (**Rate set B**) are also quite similar. In order to test the correlation stability, I have calculated the rolling correlation over 60 days (Figure 1.3). We can observe that the 60 days correlation is not very stable over time.

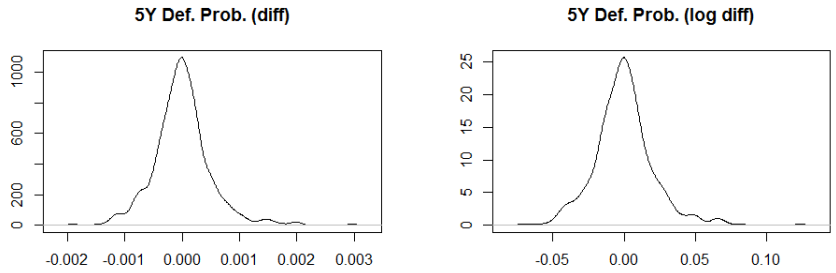

Figure 1.3: Distribution of 5Y Default Probabilities difference vs log difference

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	-0.93	27.81	30.00	48.60
TRI	-0.93	100	2.55	-3.72	-0.43
HPQ	27.81	2.55	100	45.43	35.96
IBM	30.00	-3.72	45.43	100	32.11
PFE	48.60	-0.43	35.96	32.11	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	-1.20	27.04	28.80	47.60
TRI	-1.20	100	3.62	-3.41	-0.45
HPQ	27.04	3.62	100	44.38	35.60
IBM	28.80	-3.41	44.38	100	31.97
PFE	47.60	-0.45	35.60	31.97	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	3.35	30.38	30.75	47.68
TRI	3.35	100	1.42	-0.32	2.30
HPQ	30.38	1.42	100	42.97	39.66
IBM	30.75	-0.32	42.97	100	38.28
PFE	47.68	2.30	39.66	38.28	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	2.78	29.20	29.02	46.78
TRI	2.78	100	-0.01	1.09	1.49
HPQ	29.20	-0.01	100	42.01	38.25
IBM	29.02	1.09	42.01	100	38.67
PFE	46.78	1.49	38.25	38.67	100

Table 2: 5Y Def. Probability difference (left) and log difference (right) for Rate set A (top) and B (bottom)

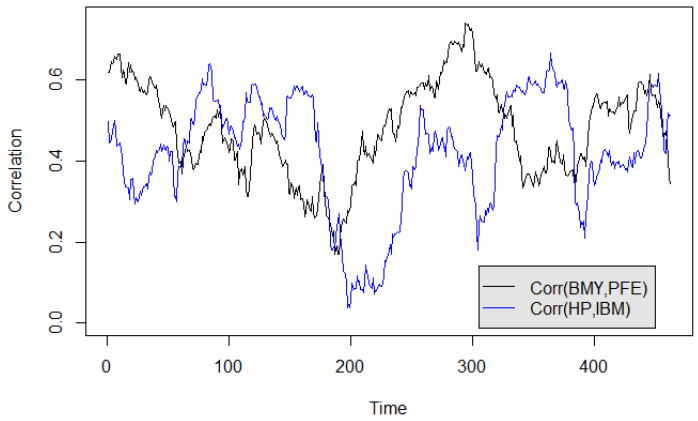

Figure 1.4: 60 days rolling correlation for Corr(BMY,PFE) and Corr(HP,IBM)

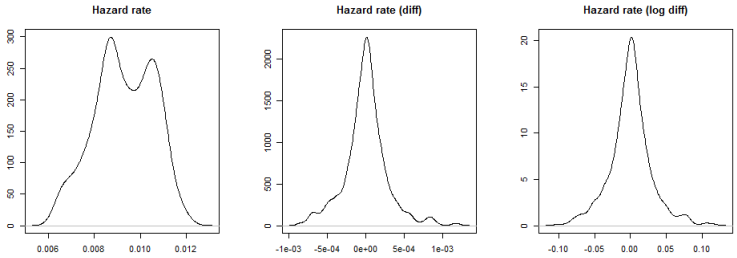

Figure 1.5: Hazard rate, hazard rates differences and log differences

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	-3.07	21.08	14.13	25.18
TRI	-3.07	100	-3.46	0.04	-7.75
HPQ	21.08	-3.46	100	18.63	23.84
IBM	14.13	0.04	18.63	100	6.78
PFE	25.18	-7.75	23.84	6.78	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	-3.36	21.30	14.02	25.48
TRI	-3.36	100	-3.92	0.39	-6.18
HPQ	21.30	-3.92	100	19.02	24.69
IBM	14.02	0.39	19.02	100	8.37
PFE	25.48	-6.18	24.69	8.37	100

Table 3: Gaussian correlation matrix for differences (left) and log differences (right)

Hazard rates

We know that hazard rates are already log difference with regards of probability of default (1.1); however when we plot their kernel density (Figure 1.5) we can see that the data is not normally distributed (bimodal distribution). However hazard rates differences and log differences are close to a normal distribution. It is therefore suggested to work with differences (or the log differences) of hazard rates. Correlation matrix for hazard rates differences and log differences are again very similar (Table 3). Finally, the 60 days rolling correlation gives a result similar to what we observed with default probabilities; correlation is not very stable over time.

Stock log returns

I have also studied how a correlation matrix estimated from stocks log returns could be used as proxy. Without surprise, log returns are almost normally distributed (Figure 1.6). The correlation matrix obtained from stock log returns is pretty close from the correlation matrix obtained from 5Y default probabilities log differences (Table 4). Again, the highest correlations are observed for Bristol Myers Squibb/Pfizer and HP/IBM. At least in this case, we can conclude that stock log returns are a good proxy for correlation matrix estimation. Finally, I tested at the stability of the correlation estimated from stocks log returns by calculating the 60 days rolling correlation (Figure 1.7). Correlations of stock log returns are still pretty unstable but slightly more stable that the correlation of 5Y default probability log difference.

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	-1.20	27.04	28.80	47.60
TRI	-1.20	100	3.62	-3.41	-0.45
HPQ	27.04	3.62	100	44.38	35.60
IBM	28.80	-3.41	44.38	100	31.97
PFE	47.60	-0.45	35.60	31.97	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	2.78	29.20	29.02	46.78
TRI	2.78	100	-0.01	1.09	1.49
HPQ	29.20	-0.01	100	42.01	38.25
IBM	29.02	1.09	42.01	100	38.67
PFE	46.78	1.49	38.25	38.67	100

Table 4: Correlation matrix from stock log return vs. default probability log diff (1 year)

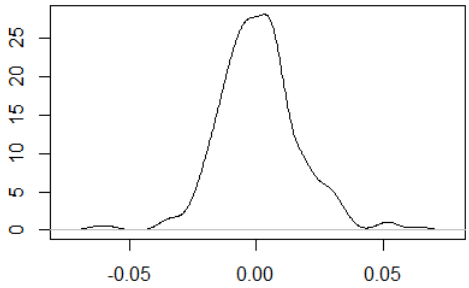

Figure 1.6: Bristol-Myers Squibb - Stock log returns

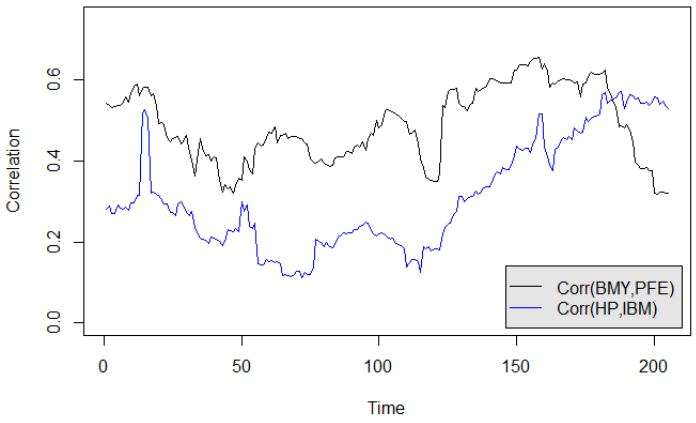

Figure 1.7: 60 days rolling correlation for Corr(BMY,PFE) and Corr(HP,IBM)

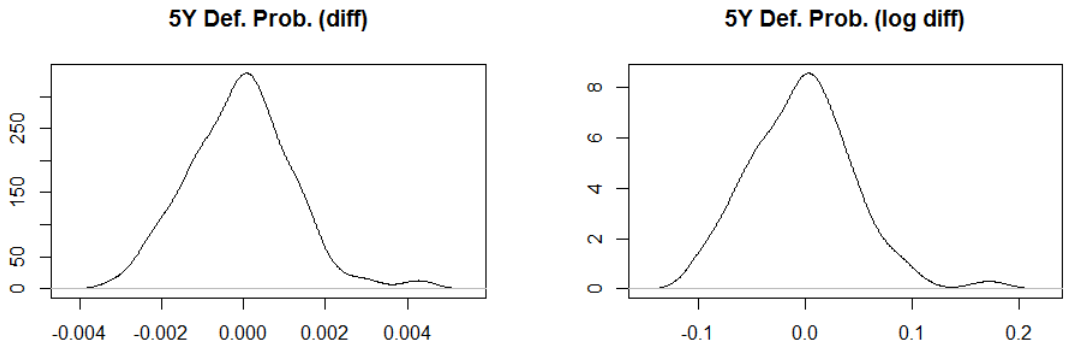

Figure 1.8: 5Y Default Probability difference vs. log difference (Weekly observations)

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	21.82	33.05	47.18	57.12
TRI	21.82	100	11.65	13.98	23.92
HPQ	33.05	11.65	100	61.60	47.65
IBM	47.18	13.98	61.60	100	63.49
PFE	57.12	23.92	47.65	63.49	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	2.78	29.20	29.02	46.78
TRI	2.78	100	-0.01	1.09	1.49
HPQ	29.20	-0.01	100	42.01	38.25
IBM	29.02	1.09	42.01	100	38.67
PFE	46.78	1.49	38.25	38.67	100

Table 5: Weekly observations log diff vs. daily observation log difference (2 years)

Weekly observations

I have snapped weekly CDS spreads data (Fridays) between 29/04/2011 and 30/04/2014 and bootstrap the credit curve on these dates. The differences and the log differences are also normally distributed (Figure 1.8). The estimated correlation matrix keeps some similarity with the one calculated with daily observations. For instance, the largest correlations are still observed for are observed for the pairs Bristol Myers Squibb/Pfizer and HP/IBM. But there are also differences: we start to see correlations between Thomson Reuters and the other reference names whereas these correlation were always very close to zero with daily observations. We can observe that the "weekly" correlations are higher than the "daily" correlations on average. Finally, I tested the stability of these correlations over time by plotting the 60 weeks rolling correlations. "Weekly" correlations appear to be more stable over "time"; they probably reflect better base correlation. We will later compare the basket CDS pricing results using daily and weekly data for 5Y default probability log differences.

1.3.2 Correlation matrix and degree of freedom for Student T copula

We used the same input data as for the correlation matrix for Gaussian copula. The process to estimate the correlation matrix for Student's t copula is the one summarized in the "Model and hypothesis" section. The correlations observed for the pairs Bristol Myers Squibb/Pfizer and HP/IBM are still among the largest (See Table 6 and Table 7).

I have applied the maximum likelihood estimation (MLE) procedure to calculate the degree of freedom.

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	2.92	20.59	19.61	31.32
TRI	2.92	100	-0.34	0.61	0.99
HPQ	20.59	-0.34	100	30.56	27.15
IBM	19.61	0.61	30.56	100	25.38
PFE	31.32	0.99	27.15	25.38	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	4.59	31.78	30.32	47.24
TRI	4.59	100	-0.54	0.96	1.55
HPQ	31.78	-0.54	100	46.18	41.37
IBM	30.32	0.96	46.18	100	38.82
PFE	47.24	1.55	41.37	38.82	100

Table 6: Daily observations - Kendall Tau (left) and Student T (right) correlation matrix

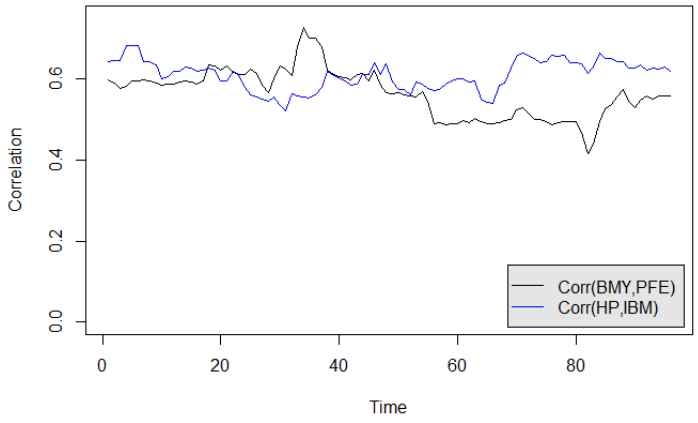

Figure 1.9: 60 days rolling correlation for Corr(BMY,PFE) and Corr(HP,IBM) - weekly data

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	15.77	24.81	34.75	43.43
TRI	15.77	100	7.75	11.78	18.08
HPQ	24.81	7.75	100	44.02	34.52
IBM	34.75	11.78	44.02	100	50.08
PFE	43.43	18.08	34.52	50.08	100

%	BMY	TRI	HPQ	IBM	PFE
BMY	100	24.53	37.99	51.91	63.05
TRI	24.53	100	12.15	18.40	28.02
HPQ	37.99	12.15	100	63.76	51.61
IBM	51.91	18.40	63.76	100	70.80
PFE	63.05	28.02	51.61	70.80	100

Table 7: Weekly observations - Kendall Tau (left) and Student T (right) correlation matrix

The procedure is described in the "Model and hypothesis" section. The degree of freedom has been calculated for daily observations and weekly observations; we obtain the below results:

Degree of Freedom	
Daily data	Weekly data
13.78	10.76

1.3.3 Spread calculation

Sampling from Gaussian copula and Student's t copula follows the procedure described in the "Model and hypothesis" section. Spread calculated is also detailed in the "Model and hypothesis" section. The following results have been obtained by running 1,000,000 simulations.

I have priced the k^{th}-to-default basket using:

1. the correlation matrix estimated from the weekly observations
2. and using the correlation matrix estimated from the daily observations.
   The fair spread of the k^{th}-to-default basket decreases as k increases. As k increases, the risk of default of the basket (i.e. risk of having at least k reference names default within the basket) decreases. It is therefore natural that the "insurance" premium (i.e basket fair spread) decreases. The spread of the first-to-default basket is higher with the Gaussian copula than with the Student's t copula. On the other hand, the spread of the k^{th}-to-default for $k > 1$ are smaller with the Gaussian copula than with the Student's t copula. The t-distribution (when $\nu < 25$) has heavier tails than the normal distribution; it means that it is more prone to producing values that fall far from its mean. I believe this explains why the the spread of $k^{th}$to-default for $k > 1$ are higher with the Student's t copula than with the Gaussian copula.

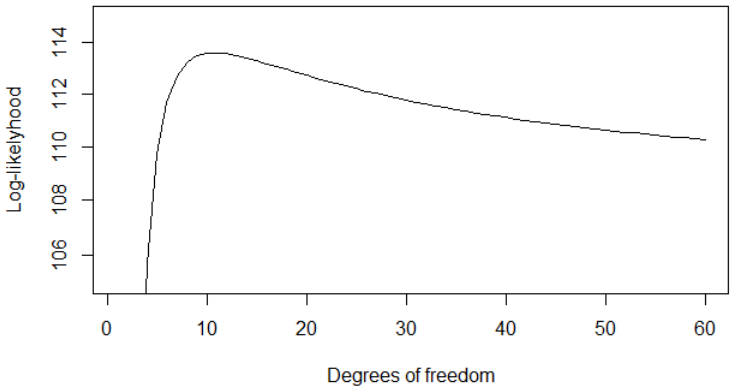

Figure 1.10: Log-Likelihood function for Student's t copula vs degree of freedom parameter (weekly data)

Remark. The first-to-default basket spreads (Gaussian and Student's t copulas) computed with the "weekly observations" correlation matrix are lower than the first-to-default basket spreads computed using "daily observations" correlation matrix. On the other hand, the k^{th}-to-default spreads with $k \in \{2, 3, 4, 5\}$ computed with the "weekly observations" are higher than the k^{th}-to-default spreads computed using "daily observations" correlation matrix. This is because, on average the "weekly" correlation matrix's elements are higher than the elements of the "daily" correlation matrix. More on this in the risk and sensitivity analysis (correlation impact).

	Daily observations		Weekly observations	
	Gaussian Copula	Student's T Copula	Gaussian Copula	Student's T Copula
$k^{th}$to default	Fair spread (bp)	Fair spread (bp)	Fair spread (bp)	Fair spread (bp)
1	43.6619	41.7913	39.8578	37.8018
2	6.5649	7.5139	8.4065	9.1998
3	1.1302	1.6419	2.2866	2.9420
4	0.1506	0.2943	0.5283	0.8390
5	0.0067	0.0217	0.0616	0.1436

1.3.4 Risk and sensitivity analysis

This section explores the impacts of the model input parameters on the k^{th}-to-default fair spread. We are moving one parameter at a time in order to clearly identify its impact; also we simplify the overall setup by taking the following assumptions:

- Homogeneous basket of 5 reference names with constant CDS spreads (flat term structure)
- Equicorrelation between all reference names
- Constant interest rate

Impact of credit quality of individual reference names

- Equicorrelation = 30%
- Interest rate = 1%
- Degree of freedom (for Student's t): 10

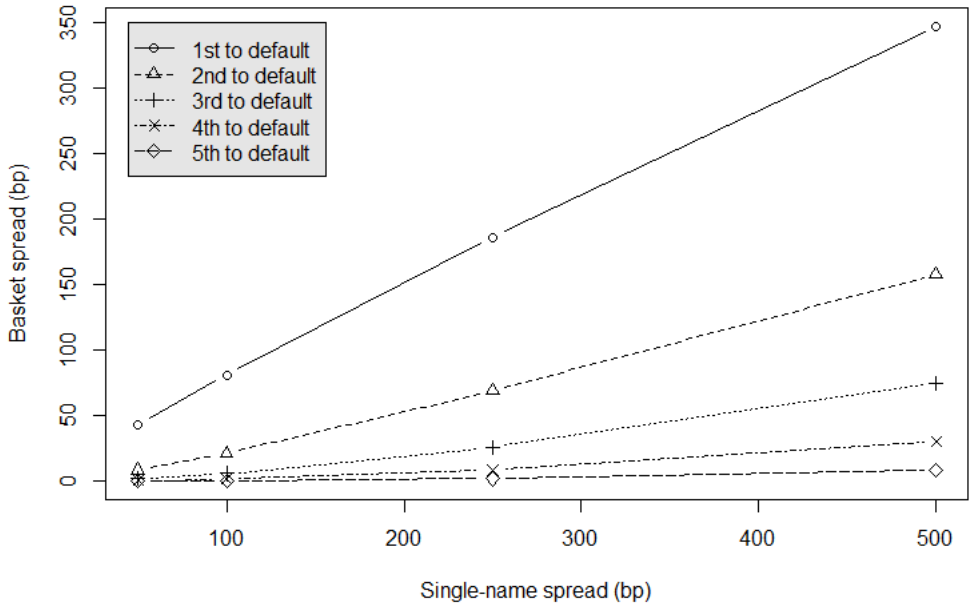

Figure 1.11: Impact of credit quality on k^{th}-to-default fair spread (Gaussian copula)

- 1,000,000 simulations

The decrease of credit quality of the individual reference names (i.e increase of their CDS spread) does not have the impact in all k^{th}-to-default fair spread (Figure 1.11). The bigger k is, the bigger is the impact of the decrease of credit quality on the spread $s^k$is (Figure 1.11). For instance the increase of the CDS spread from 50 basis points to 500 basis points multiplies the $1^{st}$to default spread by 8 ($\frac{347.4121}{42.6791} \sim 8$) but it multiplies by 45 ($\frac{74.6040}{1.6561} \sim 45$) the 3^{rd} to default spread.

	Gaussian Copula			
$k^{th}$to default	50bp	100bp	250bp	500bp
1	42.6791	80.9760	186.3073	347.4121
2	8.1186	21.0202	68.7467	157.8954
3	1.6561	5.6255	25.8877	74.6040
4	0.2917	1.2642	8.1271	29.9927
5	0.0307	0.1759	1.6065	7.8849

Student T Copula			
50bp	100bp	250bp	500bp
39.7199	76.4019	180.1036	340.5270
9.4424	22.3590	69.1917	157.6017
2.5171	6.9085	27.1378	75.1341
0.6065	1.8239	9.0312	30.5448
0.0942	0.3233	1.9560	8.1306

Impact of correlation on the fair spread

- CDS spread = 100 basis points
- Interest rate = 1%
- Degree of freedom (for Student's t): 10
- 1,000,000 simulations

The plot (Figure 1.12) shows that the impact of correlation on the k^{th}-to-defaut basket spread. Its highlights that correlation does not have the same impact on all k^{th}-to-defaut spread.

We can observe that first-to-default spread of a basket with correlation is always lower than the CDS spread of any reference name: this is related to correlation. In a basket of uncorrelated reference names (i.e. equicorrelation = 0%), most of time, when one reference name defaults, it defaults alone (i.e. no other reference name defaults within the 5 years of the contract). Intersections (several reference names defaulting in the same simulation) are very rare. In this case, the first-to-default spread is close to the CDS spread of any of the reference name (i.e. the risk of single-name CDS on the reference A and the risk on a basket CDS composed of 5 reference names which have the same credit curve as A are very close). In a basket of correlated reference names, things are different. Intersections (i.e. more than one reference name defaults within the 5 years of the contract) are going to increase; as a consequence, there will be less simulations where only one default can be observed (but there will be more simulations where more than one default can be observed). This explains why the spread of the first-to-default will be lower than the spread of any CDS when the reference names are correlated. The higher the correlation the lower the spread of the first-to-default basket CDS.
The spread of $3^{rd}/4^{th}/5^{th}$-to-default basket increases with correlation because as we have seen earlier, the number of simulations with multiple defaults increase with correlation.
Finally, the spread of the 2^{nd}-to-default basket increases with correlation until $\sim$ 70% and then decrease slightly between 70% and 100%. The 2^{nd}-to-default combines the effect that we described for the first-to-default and for the $3^{rd}/4^{th}/5^{th}$-to-default. The 2^{nd}-to-default spread start to increase like the$3^{rd}/4^{th}/5^{th}$-to-default until correlation$\sim$ 70%. When correlation goes beyond 70%, there are less and less simulations with only 2 defaults within the life of the contract; therefore the spread of the second-to-default spread starts to decrease.

	Gaussian Copula			
$k^{th}$to default	0	0.3	0.6	0.99
1	99.6695	80.9760	60.9459	24.7272
2	13.1045	21.0202	25.1358	21.7586
3	1.0561	5.6255	11.7798	19.7831
4	0.0434	1.2642	5.1764	17.9658
5	0.0002	0.1759	1.7027	15.7702

Student T Copula			
0	0.3	0.6	0.99
93.8157	76.3525	58.2076	24.4443
16.0760	22.2168	25.3058	21.6490
2.1289	6.9054	12.5474	19.7686
0.1809	1.8475	5.8387	18.0612
0.0085	0.3126	2.1023	15.9467

Impact of the recovery rate on the fair spread

- CDS spread = 100 basis points
- Equicorrelation = 30%
- Interest rate = 1%
- Degree of freedom (for Student's t): 10
- 1,000,000 simulations

This plot (Figure 1.13) shows that all k^{th}-to-default spread decreases when the recover rate increases. When the recovery rate $R \to 1$, we have the loss given default $L = 1 - R \to 0$; as a consequence have the default leg $DL \to 0$; therefore $s^k \to 0$. We can also observe that the increase in the recovery rate has the same impact in percentage on all k^{th}-to-default spreads.

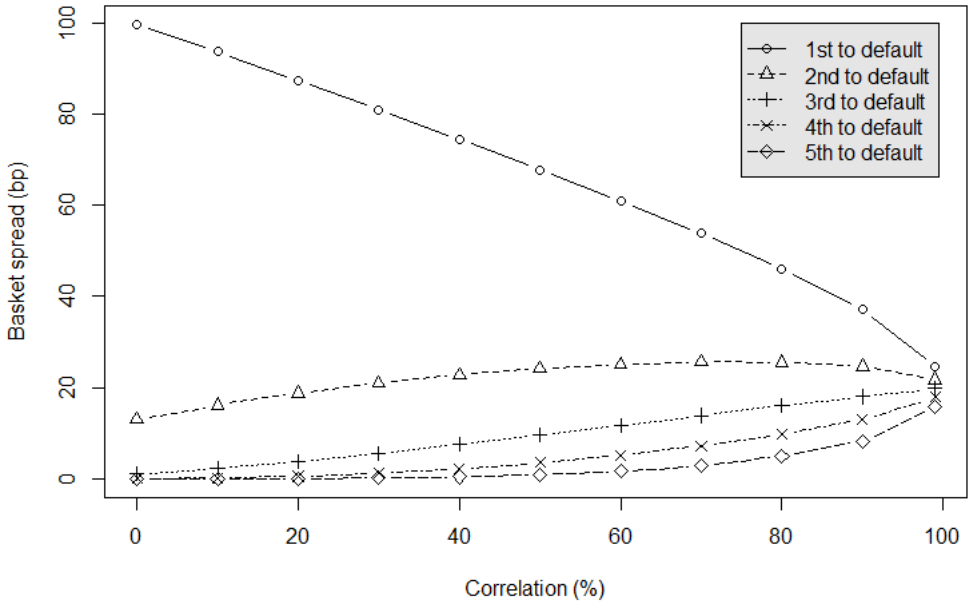

Figure 1.12: Impact of correlation on k^{th}-to-default fair spread (Gaussian copula)

	Gaussian Copula			
$k^{th}$to default	0.30	0.40	0.50	0.60
1	94.4720	80.9760	67.4800	53.9840
2	24.5236	21.0202	17.5168	14.0134
3	6.5631	5.6255	4.6879	3.7503
4	1.4749	1.2642	1.0535	0.8428
5	0.2052	0.1759	0.1466	0.1173

Student T Copula			
0.30	0.40	0.50	0.60
89.2530	76.3525	63.5134	50.9287
26.2039	22.2168	18.5329	14.8813
8.1251	6.9054	5.7410	4.6235
2.1734	1.8475	1.5155	1.2343
0.3709	0.3126	0.2543	0.2155

Impact of the discount curve on the fair spread

- CDS spread = 100 basis points
- Equicorrelation = 30%
- Degree of freedom (for Student's t): 10
- 1,000,000 simulations

For this sensitivity analysis, we assume a constant interest rate throughout the yield curve. We can observe that the increase in the the interest rate does not have the same impact on all k^{th}-to-default spreads. The first-to-default and second-to-default spreads increase with interest rates but the $3^{rd}/4^{th}/5^{th}$-to-default spreads decrease when interest rate increase.

	Gaussian Copula			
$k^{th}$to default	1%	2%	3%	4%
1	80.9502	81.4333	81.9212	82.4140
2	20.9834	21.0026	21.0217	21.0406
3	5.6333	5.6216	5.6097	5.5976
4	1.2559	1.2511	1.2461	1.2412
5	0.1704	0.1695	0.1685	0.1676

Student T Copula			
1%	2%	3%	4%
76.2095	76.8693	77.8725	78.4327
22.4999	22.2040	22.5190	22.4182
6.8674	6.7687	6.8970	6.8895
1.8482	1.8547	1.7479	1.8930
0.3160	0.2774	0.3041	0.3147

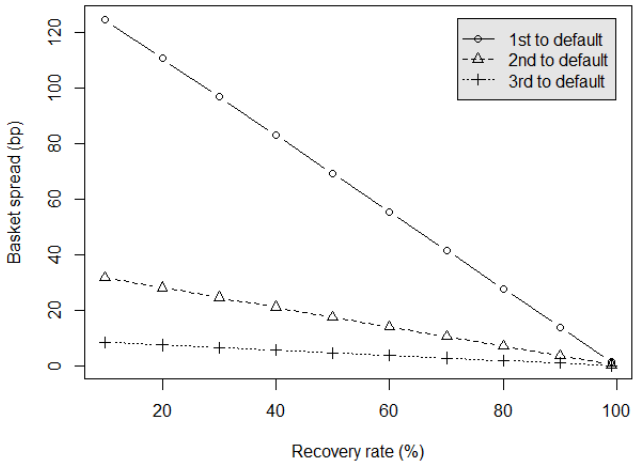

Figure 1.13: Impact of the recovery rate

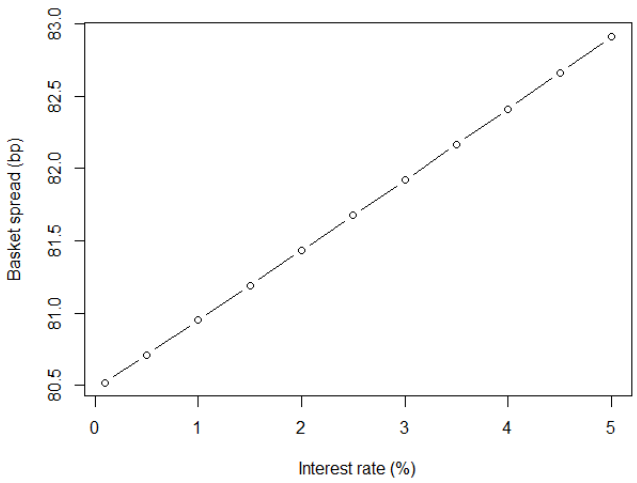

Figure 1.14: Impact of yield curve on first-to-default basket spread

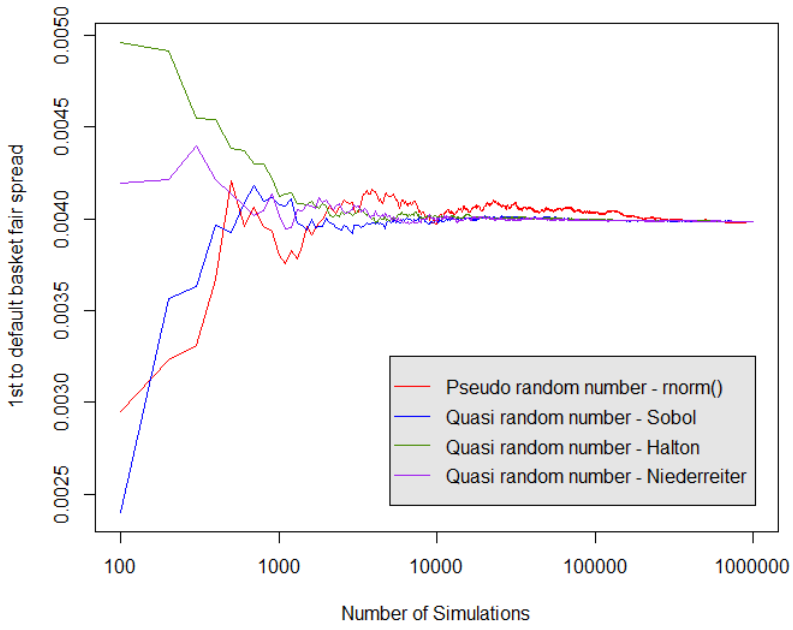

Figure 1.15: Pseudo-random vs. Sobol vs. Niederreiter vs. Halton convergence diagram for a first-to-default Basket CDS

1.4 Convergence and stability

I have run 1,000,000 simulations using pseudo-random numbers (rnorm native function in R) and low-discrepancy numbers (Sobol, Niederreiter and Halton). With this data, i obtain a convergence diagram that compares the convergence of these 3 types of numbers. The simulation that uses pseudo-random number is the slowest to converge: we can still observe significant variation in spread after 300,000 simulations. On the other hand, simulations that use Sobol and Niederreiter numbers seem to stabilize after 100,000 simulations.

- Pseudo-random number generators can be used as number pipelines.

  These generators have no built-in mechanism for sampling efficiency (each draw has no knowledge of the previous draw). They typically lead to clusters and gaps which may or may not be desirable. They emulate randomness.

- Low-discrepancy number generators

  These algorithms are designed to take advantage of our knowledge of the dimensionality and ordering of importance of the given simulation problem at hand. It is an holistic scheme aiming for as little randomness as possible.

Monte Carlo simulation using low-discrepancy numbers are, in theory, expected to converge as

$$\sim c(d)\frac{(\log N)^d}{N}$$

whereas the convergence using pseudo-random numbers is, expected to be

$$\sim \frac{1}{\sqrt{N}}$$

1.5 Numerical implementation

1.5.1 Highlights

The highlights of this implementation are:

- Integration with NAG Fortran library

  I have used the NAG Fortran library 64 bit[5] (FLW6I24DCL) because it can be integrated smoothly with R 64 bit. I have created a R function which first initialize a low-discrepancy number generator (NAG routine G05YLF) and then "draw" n numbers for each dimension (NAG routine G05YJF).

- The core pricing functions BasketCDSPricing_GaussianCopula() and BasketCDSPricing_StudentTCopula() are generic enough to price baskets with n reference names

- Pricing functions return all k^{th}-to-default basket spreads together with the single-name CDS spread (can be used for verification) and the results of the simulations which can be used to produce convergence diagrams

The improvement opportunities in this implementation are:

- Usage of NAG routines (G02AAF) to calculate the nearest correlation matrix (when the correlation matrix is not positive definite)

- Credit curves could be bootstrapped assuming quarterly payments for CDS (bootstrapping is performed assuming annual payments)

- Basket CDS could be priced assuming quarterly payments

- Premium leg is calculated with annual payment. Market practice is quarterly payments

1.5.2 Algorithm

The algorithm implemented for this project (k^{th}-to-default basket pricing) an be summarized as follows:

1. Load historical credit spreads and historical LIBOR curves - *market_data_loading.R*
2. Bootstrap historical credit curve - *credit_curve_bootstrapping_functions.R*
3. Calculate correlation matrix (Gaussian and Student's t copula) and degree of freedom - *default_probability_correlatio*
4. Calculate correlation matrix from Stock log returns - *stock_log_return_correlation_matrix.R*
5. Calculate marginal hazard rates - *calculate_marginal_hazard_rates.R*
6. Price k^{th}-to-default spread using sample from Gaussian copula - *sampling_from_gaussian_copula.R*
7. Price k^{th}-to-default spread using sample from Student's copula - *sampling_from_student_t_copula.R*

1.5.3 Market data

Two types of market data have been required for this implementation:

- Credit default swap spreads
- Discounting curve

[5] This free trial license has been graciously given by the Numerical Algorithms Group

Credit default swap spreads

Historical CDS spreads for the following entities have been obtained from Markit:

Ticker	RedCode	Tier	Currency	DocClause	Sector	Region
BMY	1C1134	SNRFOR	USD	XR	Healthcare	N.Amer
TRI	8GD65J	SNRFOR	USD	XR	Consumer Services	N.Amer
HPQ	46AA59	SNRFOR	USD	XR	Technology	N.Amer
IBM	49EB20	SNRFOR	USD	XR	Technology	N.Amer
PFE	7I8789	SNRFOR	USD	XR	Healthcare	N.Amer

Discounting curve

Historical discounting curves have been obtained from Bloomberg using the following Bloomberg Tickers:

Bloomberg Ticker	Description
S0023D 1Y BLC2	Discount factor (LIBOR curve) for 1Y
S0023D 2Y BLC2	Discount factor (LIBOR curve) for 2Y
S0023D 3Y BLC2	Discount factor (LIBOR curve) for 3Y
S0023D 4Y BLC2	Discount factor (LIBOR curve) for 4Y
S0023D 5Y BLC2	Discount factor (LIBOR curve) for 5Y

1.5.4 Instructions and package description

This program has been written using R (version 3.1.0, 64-bit) and Rstudio as IDE. In order to launch this program, please follow this procedure:

1. Copy the whole folder "creditbasket" to your local computer (e.g. C:\temp\creditbasket)
2. Open the file "creditbasket\src\master_program.R" (e.g. C:\temp\creditbasket\src\master_program.R)
3. Update the value of the variable **current_dir** to match your environment (e.g. current_dir = "C:/temp/creditbasket/src")
4. Launch R
5. type: source ("C:/temp/creditbasket/src/master_program.R") (replace the path to master_program.R to match your environment)
6. The program will start.

Folder	File	Description
.\creditbasket		Top folder
.\creditbasket\src		Folder containing source code (R)
	master_program.R	Main program that calls all the other programs
	class_definition.R	Defines several classes which are used throughout the project
	install_register_packages.R	Installs and registers external packages which are required
	market_data_functions.R	Defines the function that loads the historical CDS spreads and the historical discount curve
	market_data_loading.R	Loads historical CDS spreads and the historical discount curve
	credit_curve_bootstrapping_functions.R	Defines the functions that bootstrap credit curve
	default_probability_correlation_matrix.R	Estimate Gaussian correlation matrix, Student's T correlation matrix and degree of freedom
	estimate_marginal_hazard_rates.R	Calculate marginal hazard rates
	nag_library_wrapper.R	Defines a function that allows R to call the NAG Fortran 64 bit library
	gaussian_copula_functions.R	Defines main pricing functions for Gaussian copula: Monte Carlo loop and kth-to-default basket spread pricing
	student_t_copula_functions.R	Defines main pricing functions for Student's T copula: Monte Carlo loop and kth-to-default basket spread pricing
	sampling_from_gaussian_copula.R	Calls above mentioned functions; display kth-to-default basket spread and plots convergence diagrams
	sampling_from_student_t_copula.R	Calls above mentionned functions; display kth-to-default basket spread and plots convergence diagrams
.\creditbasket\data		Folder containing market data files. Various market data files for CDS spreads, stock returns and historical yield curve

2 Interest Rate Derivatives

2.1 Introduction

The Heath–Jarrow–Morton (HJM) framework is a framework that model the evolution of the yield curve by representing the evolution of the instantaneous forward rates. This model imposes the no-arbitrage restriction: the drift can't be random and depends on the volatility of forward rates. In HJM, volatility of forward rates is estimated from historical interest rate data (historical forward rates in our case). We will show how the principal component analysis is conducted (estimation and analysis of covariance matrices) to identify the major factors which influence the evolution of the yield curve before describing the projection of the forward curve and the pricing of various interest rate derivatives instrument using Monte Carlo simulation (using pseudo-random and low-discrepancy numbers). We use an OIS[6] spot curve as discounting curve in order to reflect the cost of funding.

2.2 Model, hypothesis

2.2.1 Forward rate notation

Definition 18. Forward rate

$F(t;T,S)$ is the complete notation for the forward rate available at time t for the period starting a time T (re-set or expiry time of the forward) and ending at time S (maturity or the loan/deposit or cash-flow payment time)

Definition 19. Instantaneous forward rate

The instantaneous forward rate $f(t,T)$, as seen from some time t, start at time T and applies over an instant $\Delta t \to 0$.

$$F(t;T,S) \to f(t,T) \text{ when } S \to T^+$$

It is a number from a continuous process:

$$\begin{aligned} f(t,T) &= -\lim_{\Delta t \to 0} \frac{\log Z(t;T+\Delta t) - \log Z(t;T)}{\Delta t} \\ &= -\frac{\partial}{\partial T} \log\ Z(t;T) \end{aligned}$$

Remark. Forward rates are bootstrapped using actual or synthetic zero-coupon bonds $Z(t;T)$

2.2.2 Heath–Jarrow–Morton

Definition 20. Single-factor HJM model

The stochastic differential equation for an instantaneous forward rate is written as:

$$df(t,T) = m(t,T)dt + \nu(t,T)dX$$

where $m(t,T)$ is the risk-neutral drift represented by no arbitrage condition:

$$m(t,T) = \nu(t,T) \int_t^T \nu(t,s)\ ds$$

Remark. The risk-neutral drift depends on volatility

[6] Overnight indexed swap

Remark. A single-factor model for a spot interest rate can't capture the richness of the yield curves observed in practice. Generally speaking, a spot rate model with one source of randomness may be good at reflecting the overall level of a yield curve but fails when it comes to shifts in the yield curve that are substantially different at different maturities.

Definition. Multi-factor HJM model
The N-dimensional stochastic differential equation (multi-factor HJM) is as follows:

$$df(t,T) = m(t,T)dt + \sum_{i=1}^{N} \nu_i(t,T)dX_i$$

where $m(t,T)$ is the risk-neutral drift represented by no arbitrage condition:

$$m(t,T) = \sum_{i=1}^{N} \nu_i(t,T) \int_t^T \nu_i(t,s)\, ds$$

Remark. Multi-factor means adding sources of randomness dX_i. Factors must be uncorrelated, otherwise their impact would be overlapping and the model would require cross-terms and estimation of correlations $\mathbb{E}\left[dX_i dX_j\right] = \rho dt$

2.2.3 Musiela parametrization

It is useful to model the volatility of the forward rate for various maturities (6 months, one year, two years, etc.) and not for each maturity dates (1 Jan 2014, 1 Jun 2014, etc.). Therefore we need to re-parametrize the volatility, the drift and the stochastic differential equation as function of tenor time $\tau = T - t$. We call $\overline{\nu}$ the volatility function expressed as a function of tenor time τ:

$$\nu(t,T) = \overline{\nu}(t,T-t) = \overline{\nu}(t,\tau)$$

Definition 21. Single-factor HJM model (Musiela parametrization)
The stochastic differential equation for an instantaneous forward rate (expressed as a function of tenor time) is written as:

$$d\overline{f}(t,\tau) = \overline{m}(t,\tau)dt + \overline{\nu}(t,\tau)dX$$

where $\overline{m}(t,\tau)$ is the risk-neutral drift (expressed as a function of tenor time):

$$\overline{m}(t,\tau) = \overline{\nu}(t,\tau) \int_0^\tau \overline{\nu}(t,s)\, ds + \frac{\partial}{\partial \tau}\overline{f}(t,\tau)$$

2.2.4 Multi-factor Musiela parametrization

Definition 22. Multi-factor HJM model (Musiela parametrization)
The N-dimensional stochastic differential equation is as follows:

$$d\overline{f}(t,\tau) = \overline{m}(t,\tau)dt + \sum_{i=1}^{k} \overline{\nu}_i(t,\tau)dX_i$$

where $\overline{m}(t,\tau)$ is the risk-neutral drift (expressed as a function of tenor time):

$$\overline{m}(t,\tau) = \sum_{i=1}^{k} \overline{\nu}_i(t,\tau) \int_0^\tau \overline{\nu}_i(t,s)\, ds + \frac{\partial \overline{f}(t,\tau)}{\partial \tau}$$

Remark. The model works with $k < N$ volatility functions that determine the diffusion of k independent factors represented by their uncorrelated Brownian Motions dX_i

2.2.5 Principal component analysis

Covariance matrix

The covariance matrix is obtained by following these steps:

1. Gather a $n \times m$ matrix of historical forward rates (n is the number of observations, m is the number of tenor/maturities)
2. Calculate an $(n-1) \times m$ matrix of daily differences (log differences since we are using data in a low-rate regime - market data from June 2011 to May 2014)
3. Calculate the covariance of the matrix of daily changes expressed in percentages Σ_{daily}. $\Sigma_{daily}$is a $(m \times m)$ matrix
4. Obtain the covariance matrix Σ of (annualized and expressed in decimal) as follows:

$$\Sigma = \frac{252}{100 \times 100} \Sigma_{daily}$$

Principal component analysis idea

We want to understand the factors that drive the evolution of the forward curve. We can identify these factors by decomposing the covariance matrix of differences Σ (i.e compute the eigenvalues and eigenvectors of the covariance matrix).

$$\Sigma = V \Lambda V^{-1}$$

where Λ is the matrix with eigenvalues along its diagonal and zeros elsewhere

$$\Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

and V is a matrix with eigenvectors in its column

$$V = \left(e^{(1)}, \ldots, e^{(n)} \right)$$

The eigenvector $e^{(1)}$ is associated to the eigenvalue λ_1, the eigenvector $e^{(2)}$ is associated to the eigenvalue λ_2, etc. The eigenvectors are orthogonal projections of the changes in data in directions of minimum variance. There are several ways to decompose matrices. We have chosen the Jacobi transformation which is not very computationally efficient but has the advantage of keeping the structure of the original covariance matrix.

We select k eigenvectors that correspond to the largest eigenvalues to serve as volatility functions:

$$\overline{\nu}_i(t,T) = \sqrt{\lambda_i} e^{(i)}$$

Remark. The first entry in any vector $e^{(i)}$ represents the movement of the one-month forward rate ($\tau = 0.08$), the second entry represents the movement in the six-month forward rate ($\tau = 0.5$), the third entry represents the movement of the one-year forward rate ($\tau = 1$) and so on.

Jacobi transformation

Jacobi transformation is a tractable numerical method to obtain eigenvalues and eigenvectors (diagonalization). The main idea is to eliminate largest off-diagonal element by rotating the matrix at each iteration. The following steps describe the algorithm:

1. Calculate the sum of the square of the elements of the upper triangle of the input matrix Σ

2. Continue to next step if this sum is more than the expected tolerance; otherwise stop recursive process and obtain the eigenvalues from the diagonal of Σ
3. Identify the largest off-diagonal element $\sigma_{p,q}$ of the matrix Σ
4. Calculate the rotation matrix angle ϕ required to eliminate $\sigma_{p,q}$
   $\tan(2\phi) = \frac{2\sigma_{p,q}}{\sigma_{q,q}-\sigma_{p,p}}$
   Therefore we obtain: $\phi = \frac{1}{2}\arctan\left(\frac{2\sigma_{p,q}}{\sigma_{q,q}-\sigma_{p,p}}\right)$
5. Calculate the rotation matrix $P_{p,q}$ as

$$P_{p,q} = \begin{pmatrix} 1 & & & & & & \\ & \ddots & & & & 0 & \\ & & \cos(\phi) & \cdots & \sin(\phi) & & \\ & & \vdots & \ddots & \vdots & & \\ & & -\sin(\phi) & \cdots & \cos(\phi) & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{pmatrix}$$

6. Calculate the matrix $\Sigma^R = (P_{p,q})^T \times \Sigma \times P_{p,q}$
7. Go back to step 1 and take Σ^Ras input (i.e. $\Sigma = \Sigma^R$)

Remark. This method is computationally inefficient as each rotation destroys zeros obtained at previous step; however it achieves convergence. The covariance matrix Σ becomes a diagonal matrix with eigenvalues in its diagonal:

$$\Sigma \implies \Lambda = \begin{pmatrix} \lambda_1 & & \\ & \ddots & \\ & & \lambda_n \end{pmatrix}$$

Eigenvectors are recovered by iteratively multiplying the rotation matrix $P_{p,q}$:

$$V = P_0 \times P_1 \times \ldots \times P_m$$

where m is the number of iterations required to reach the expected tolerance

Volatility fitting

As we have seen, HJM drift calculation requires an integration over the volatility. In order to perform an integration, we need to fit these volatility numbers to a function. We are performing polynomial fitting by cubic spline. It means finding a set of $(\beta_0, \beta_1, \beta_2, \beta_3)$ for each power ofτ so that the function $\overline{\nu}(t,\tau) = \beta_0 + \beta_1\tau + \beta_2\tau^2 + \beta_3\tau^3$approximates the values $\overline{\nu}_i(t,\tau)$.

Remark. There are alternatives to to cubic spline fitting: multi-spline / piecewise cubic spline (fit parts of the curve with different splines) or a logarithmic function $a + b\log\tau$ for the first component

2.2.6 Zero-Coupon Bond pricing

Definition 23. Zero coupon bond price in HJM

Definition. The ZCB price comes from the solution for $dZ = r(t)dt$ under the risk-neutral measure:

$$\begin{aligned} Z(t;T) &= \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_t^T r(s)\,ds\right)\right] \\ &= \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_t^T f(t,s)\,ds\right)\right] \\ &= \mathbb{E}^{\mathbb{Q}}\left[\exp\left(-\int_0^\tau \overline{f}(t,s)\,ds\right)\right] \end{aligned}$$

2.2.7 LIBOR pricing

Definition 24. Continuously compounded LIBOR rate

LIBOR is equal to the "fixed" yield of a future-starting fairly priced ZCB. The LIBOR rate $L(t;T)$ that resets at time t and is paid until maturity time T is expressed as:

$$L(t;T) = -\frac{\log Z(t;T)}{T-t}$$

Definition 25. Continuously compounded LIBOR rate in HJM (Musiela parametrization)

Since we know $Z(t;T) = \exp(-\int_t^T f(t,s)\,ds)$, we can derive the following:

$$\begin{aligned} L(t;T) &= -\frac{\log Z(t;T)}{T-t} \\ &= -\frac{\log(\exp(-\int_t^T f(t,s)\,ds))}{T-t} \\ &= \frac{1}{T-t}\int_t^T f(t,s)\,ds \\ &= \frac{1}{\tau}\int_0^\tau \overline{f}(t,s)\,ds \end{aligned}$$

where $\tau = T - t$

Definition 26. Simple annualized LIBOR rate

The simple annualized LIBOR rate (i.e the standard LIBOR rate) can be expressed as:

$$L^{'} = m(e^{\frac{L}{m}} - 1)$$

where m is the compounding frequency

Remark. In our context, we work with 3M LIBOR. Therefore, we have $L^{3M} = 4(e^{\frac{L}{4}} - 1)$

2.2.8 Cap/Floor pricing

Definition 27. Caplet price

A caplet is an interest option that pays a cash-flow based on the value of LIBOR at a reset time T_i. The cash-flow is paid for the period from reset (T_i) to maturity (T_{i+1}) in arrears. This cash-flow discounted to the current time can be expressed as:

$$Z(0;T_{i+1}) \times \max\left[L(T_i,T_{i+1}) - K, 0\right] \times \tau \times N = Z(0;T_{i+1}) \times \left(L(T_i,T_{i+1}) - K\right)^+ \times \tau \times N$$

where

- $L(T_i, T_{i+1})$ is the forward LIBOR rate (simple annualized) that resets at T_i and matures at T_{i+1}
- $\tau = T_{i+1} - T_i$ is the year fraction that converts the annualized rate to the period
- N is the notional (we will scale it to 1 going forward)

Definition 28. Cap price
A $\text{Cap}(t; T; K)$ starts at time t (beginning of first period) and matures at T. It can be decomposed into n caplets of shorter maturities:

$$\text{Cap}(t; T; K) = \sum_{i=1}^{n} \text{Cpl}_i(T_i, T_{i+1}, K)$$

Remark. If we assume 3M maturity caplets, we have $n = \frac{T-t}{4}$

Therefore, we can express the cap discounted payoff (price) as:

$$\text{Cap}(t; T; K) = \sum_{i=1}^{n} Z(0, T_{i+1}) \times (L(T_i, T_{i+1}) - K)^+ \times \tau \times N$$

Remark. Buying a cap allows protection from increase in LIBOR rate:

$$L - (L - K)^+ = \min(L, K)$$

Definition 29. Floor price
The price of a floor can be obtained from the price of a cap using this cap/floor relationship formula:

$$\text{Cap}(t; T; K) - \text{Floor}(t; T; K) = N \sum_{i=1}^{n} L(T_i, T_{i+1}) \times \tau \times Z(0, T_{i+1})$$

Definition 30. Floorlet price
The price of a floorlet can also obtained from the price of a caplet using this caplet/floorlet relationship formula:

$$\text{Caplet}(T_i; T_{i+1}) - \text{Floorlet}(T_i; T_{i+1}) = L(T_i, T_{i+1}) \times \tau \times Z(0, T_{i+1}) \times N$$

2.2.9 Swaption pricing

Definition 31. Vanilla payer swaption price
We can write the discounted payoff of a payer swaption by considering the value of the underlying payer IRS at its first reset date t (it is also maturity of the option embedded in the swaption):

$$N \sum_{i=1}^{n} (L(T_i, T_{i+1}) - K) \times \tau \times Z(t, T_{i+1})$$

The option will be exercised only if the above value is positive. In order to obtain the swaption payoff at time t, we need to apply the positive-part operator. Discounting this payoff from time t to the current time, we obtain:

$$\text{PayerSwaption}(t; T; K) = N \times Z(0, t) \left(\sum_{i=1}^{n} (L(T_i, T_{i+1}) - K) \times \tau \times Z(t, T_{i+1}) \right)^+$$

2.2.10 Black76

The market quotes Caps/Floors and Swaptions in terms of implied volatility. The below formula will help us to convert the prices we obtain via HJM as Black76 implied volatility.

Definition 32. Caplet/Floorlet pricing with Black76
The price of a caplet using Black76 can be expressed as:

$$\text{Cpl}^{Bl}(0, T_i, T_{i+1}, K, \zeta_i) = Z(0, T_{i+1}) \times \tau_i \times \text{Black76}(K, F_i(0), \zeta_i)$$

where $\text{Black76}(K, F_i(0), \zeta_i)$ is the Black formula defined as:

$$\begin{aligned} \text{Black76}(K, F_i(0), \zeta_i) &= \mathbb{E}^{\mathbb{Q}}\left[F_i(t) - K\right]^+ \\ &= F_i(0)\Phi(d_1) - K\Phi(d_2) \end{aligned}$$

with

$$\begin{aligned} d_1 &= \frac{\log \frac{F}{K} + \frac{\zeta^2}{2}}{\zeta} \\ d_2 &= \frac{\log \frac{F}{K} - \frac{\zeta^2}{2}}{\zeta} \end{aligned}$$

where ζ^2 is defined as $\zeta^2 = \zeta_i^2 T_i$

Remark. The cap price is simply calculated as the sum of its caplet prices as we have seen in definition 28.

Definition 33. Cap/Floor implied volatility with Black76
We find the implied volatility using a simple root finding procedure (Bisection or Newton-Raphson methods) applied on the cap price formula.

Remark. We assume that volatility is constant throughout the cap; i.e. all caplets are priced with the same volatility

2.3 Main result

2.3.1 Principal Component Analysis

I have worked with 3 years of GBP historical forward curves obtained from Bank of England (Bank Liability Curve constructed from LIBOR-linked instruments between 01/06/2011 and 30/05/2014). For the principal component analysis, I have taken the one-month rate from the short-end file (one-month to five-year rates) and all the rates from the long-end file (6 months, 1 year, ... 25 years). The process to calculate the covariance matrix is the one summarized in the "Model and hypothesis" section (data must re-annualized and converted from percentages to decimals). The plot (Figure 2.1) shows the yield curve at different time between June 2011 and May 2014. We can observe a certain homogeneity between these various curves.

Using the Jacobi transformation on the covariance matrix Σ, we obtain the eigenvalues and eigenvectors of Σ. I have sorted the eigenvalues and calculated their weight and cumulative weight to identify how many factors i should consider to cover at least 95% of the movement of the curve.

$$\text{Weight}(\lambda_k) = \frac{\lambda_k}{\sum_{i=1}^{N} \lambda_i}$$

$$\text{CumulativeWeight}(\lambda_k) = \frac{\sum_{i=1}^{k} \lambda_i}{\sum_{i=1}^{N} \lambda_i}$$

where N is the dimension of the covariance matrix (i.e. the number of eigenvalues).

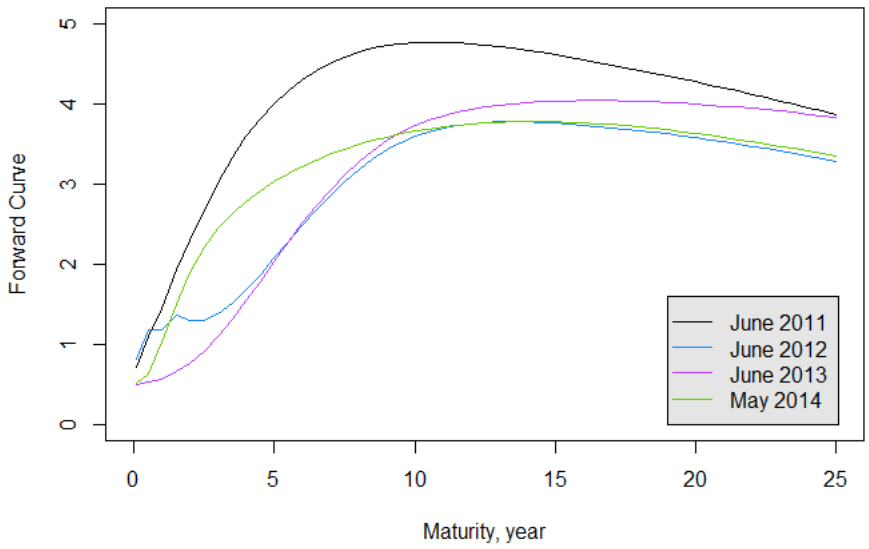

Figure 2.1: GBP Forward rate term structure between June 2011 and May 2014

I have decided to take 5 factors into account: all combined, they explain 97.26% of the movement of the curve. The below table gives the eigenvalue, the tenor τ, the weight and the cumulative weight for each of these principal components (factors).

Remark. τ is given as indication. Orthogonal factors don't necessarily indicative come from the behavior of the curve at a particular tenor/maturity.

Principal component	λ	τ	Weight (%)	Cumul. Weight (%)
PC1	0.0002959422	2	68.18	68.18
PC2	0.00007297595	0.5	16.81	84.99
PC3	0.00003116608	25	7.18	92.18
PC4	0.00001409952	3.5	3.24	95.42
PC5	0.000007983922	1	1.84	97.26

The plot (Figure 2.2) shows the eigenvectors of the 5 principal components that i have selected. The plot (Figure 2.3) shows the volatility $\overline{\nu}_i(t,T) = \sqrt{\lambda_i}e^{(i)}$ and their fitted volatility functions. The below table gives the parameters $(\beta_0, \beta_1, \beta_2, \beta_3)$ for these functions:

Principal component	β_0	β_1	β_2	β_3
PC1	0.00156	0	0	0
PC2	0.00249	-0.00065	0.00004	0
PC3	0.00126	-0.00061	0.00005	0
PC4	0.00033	-0.00007	0	0
PC5	-0.00014	-0.00004	0.00001	0

As expected, we can observe a "parallel shift" for the first principal component (PC1); it defines the general shape of the yield curve and accounts for 68% of the changes in the yield curve. Finally, we can observe that the fitted functions are crossing the x axis (i.e. $f(x) = 0$) pretty much for the same value as the actual volatility function. This is regarded as a criterion for "good fitting". The multivariate linear regression (power of τ) that we performed to find these fitted curves introduce model risk. Adding more term (higher powers of τ) will increase in-sample accuracy of fit but might represent an over-fitting.

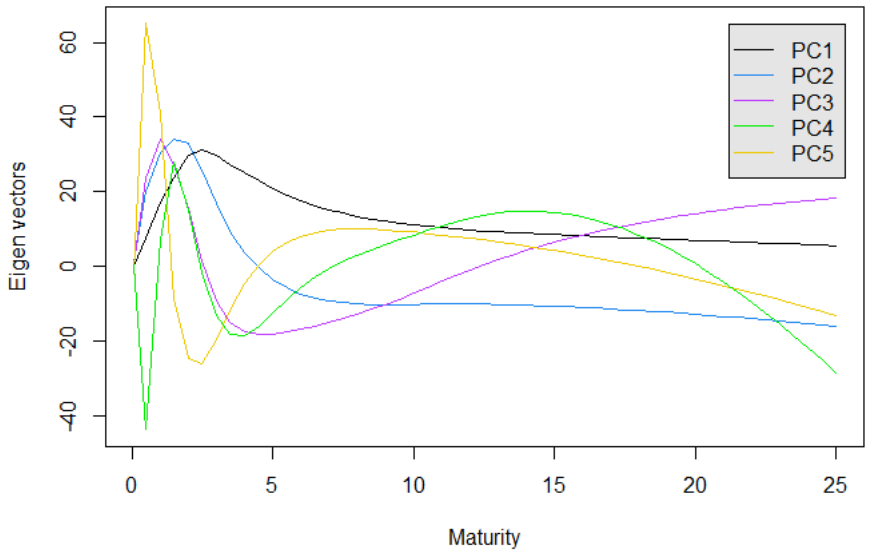

Figure 2.2: Eigenvectors for the 5 principal components

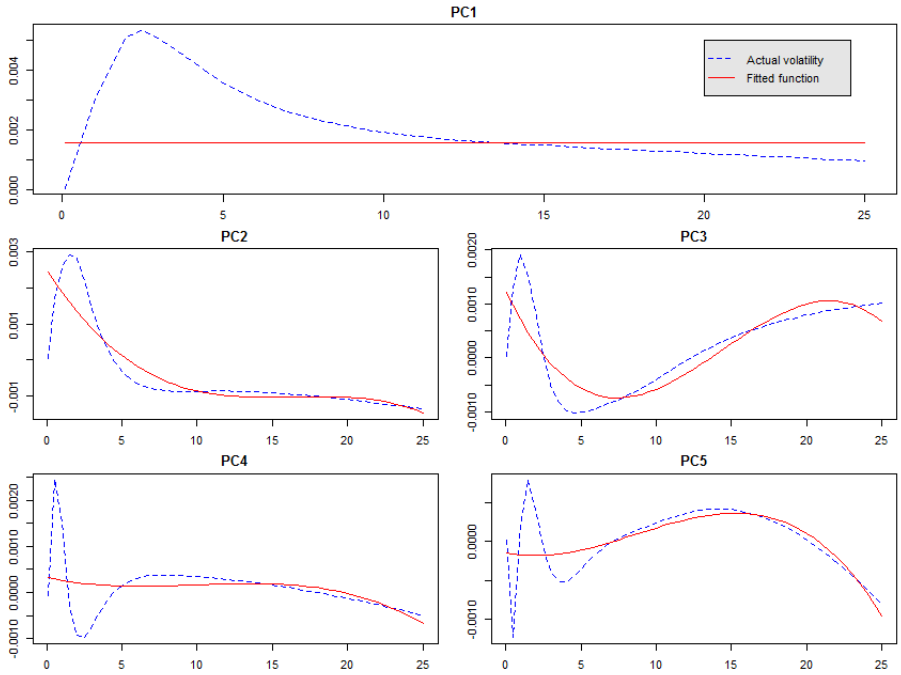

Figure 2.3: Actual volatility vs. Cubic spline function

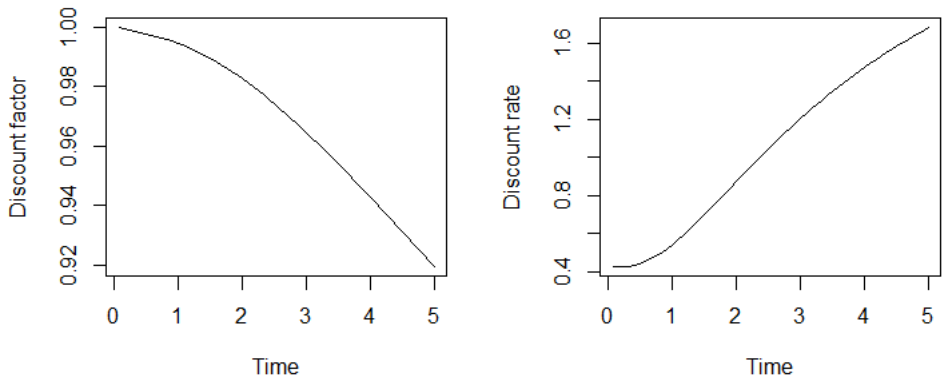

Figure 2.4: GBP OIS Spot curve

2.3.2 OIS Discounting

Discounting is a key aspect of pricing because interest rate derivatives are decomposable into cash-flows. This implementation uses dual-curve pricing: one forward/forecast curve (that gauges an expectation of what short rate will be in the future) and one discounting curve. I have used the GBP OIS spot curve obtained from Bank Of England (UK OIS spot curve) which gives us discount rates (bootstrapped from overnight index swap rates) for a wide range of tenors/maturities since 2009. Bank of England publishes OIS discount rates from 1 month up to to 5 years (every month in between). Liquidity in OIS contracts beyond 5 years is limited; therefore Bank of England only publishes OIS curve up to 5 years. OIS curves over 5 years generally use OIS spread over LIBOR.

2.3.3 Forward rates projection matrix

Pricing of various interest rate derivatives instrument within the HJM framework is performed through Monte Carlo simulations. For each simulation, we use the HJM multi-factor stochastic differential equation to generate a matrix that describes one possible evolution of the forward curve starting from a given forward curve (the forward curve on the valuation date). This matrix has 60 columns (one column for each tenor from one-month to 5 years - every month) and $n \times 100$ rows (where n is the number of years for which we need to project the forward curve). For instance, a one-year forward starting cap will require a projection up to 1 year (i.e. 100 rows of projection). The time-step between each of this rows is $\delta t = 0.01$. The below matrix illustrates how the matrix is constructed. The volatility of the HJM stochastic differential equation is calculated as a sum of the volatility functions of each factor multiplied by the Brownian motion of this factor. The drift calculation requires an integration over the volatility functions (not an issue since we are working with fitted cubic spline) and the calculation of the derivative $\frac{\partial \overline{f}(t,\tau)}{\partial \tau}$. This derivative is generally calculated as a forward difference on the previous time-step, i.e $\frac{\partial \overline{f}(t,\tau)}{\partial \tau} = \frac{\overline{f}(t-\delta t,\tau+\delta\tau)-\overline{f}(t-\delta t,\tau)}{\delta\tau}$; except for the last column of the projection matrix (we can't go forward), in this case, we calculate this derivative as a backward difference: $\frac{\partial \overline{f}(t,\tau)}{\partial \tau} = \frac{\overline{f}(t-\delta t,\tau)-\overline{f}(t-\delta t,\tau-\delta\tau)}{\delta\tau}$. As we have seen previously, the HMJ multi-factor stochastic differential equation (Musiela parametrization) is:

$$d\overline{f}(t,\tau) = \overline{m}(t,T)dt + \sum_{i=1}^{k} \overline{\nu}_i(t,\tau)dX_i$$

$$\overline{m}(t,T) = \sum_{i=1}^{k} \overline{\nu}_i(t,\tau) \int_t^T \overline{\nu}_i(t,s)\, ds + \frac{\partial \overline{f}(t,\tau)}{\partial \tau}$$

	$\frac{1}{12}$	$\frac{2}{12}$	$\frac{3}{12}$	$\frac{4}{12}$	$\cdots$	$\frac{60}{12}$
0						
0.01						
0.02			$\overline{f}(0.02, \frac{3}{12})$			
0.03						
$\vdots$						
4						

Once the matrix is entirely populated, we perform the calculations required (calculation of Bond price, LIBOR rates, Cap, Swaptions, etc.) and store the results. We start then another simulation unless we have reached the number of simulations required; in this case, we average all the results obtained for each simulation to obtain the expectation (price) of the instruments we are pricing.

2.3.4 Zero Coupon Bond pricing

Zero Coupon Bonds have been priced using the formula described in the "Model and hypothesis" section. The one-month rate $\overline{f}(t, \tau = 0.08)$ is taken as proxy for the spot rate $r(t)$. These are bond prices obtained after running 30,000 simulations. As we can expect, a ZC bond starting today with 2 years maturity $Z(0;2)$ has more value than a bond starting today with 5 years maturity $Z(0;5)$.

$Z(t;T)$	T				
t	1	2	3	4	5
0	0.9923	0.9770	0.9555	0.9304	0.9037
1	NA	0.9844	0.9627	0.9375	0.9105

Remark. $Z(0;1) \times Z(1;2) = 0.9769137$. This is different from $Z(0;2) = 0.9770264$; it seems that there is an arbitrage opportunity with this pricing.

2.3.5 Cap pricing

The caps have been have been priced using the formula described in the "Model and hypothesis" section. The pricing algorithm for one cap goes as follows:

1. For each simulation:
   (a) Calculate all the LIBOR rates required for the cap
   (b) Price each caplet using the LIBOR rates (Definition 27)
   (c) Calculate the cap price as the sum of the caplet prices (Definition 28)
   (d) Store the cap price
2. Once all simulations are completed, obtain the cap price by averaging the cap price obtained for each simulation (expectation)
3. Compute the implied volatility for the cap using Black76

I have assumed quarterly payments for the caps (i.e. caplet maturity is 3M). The following prices have been obtained after running 30,000 simulations. I have priced caps for a wide range of strikes and maturities so that we can observe the volatility skew and the term structure.$\text{Cap}(t;T;K)$ is the price as of today ($time = 0$) of a cap with strike K that start at time t and matures at times T. This can be decomposed into n 3M caplets ($n = \frac{T-t}{4}$).

Example. In the market, a 1×2 cap is a 1 year cap starting in 1 year ($\text{Cap}(1;2;K)$ in our notation) can be decomposed into 4 caplets: [1,1.25], [1.25,1.5], [1.5,1.75] and [1.75,2]

Remark. The first caplet is generally ignored for spot-starting cap because there is no uncertainty about the level of first LIBOR rate (it is already fixed).

Example. In the market, a one year cap starting from spot ($\text{Cap}(0;1;K)$ in our notation) can be decomposed into 3 caplets: [0.25,0.5], [0.5,0.75] and [0.75,1]

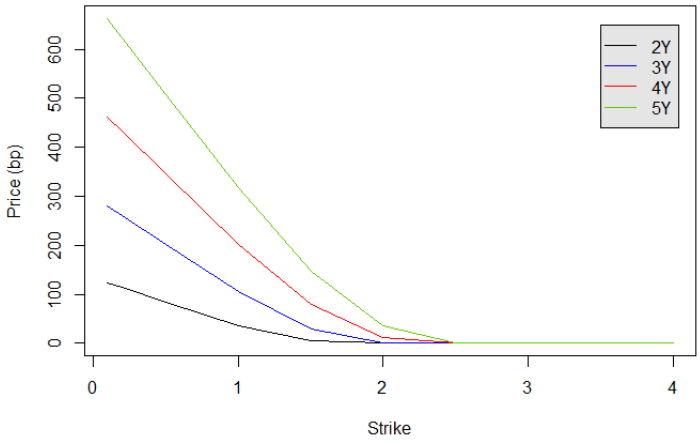

Figure 2.5: Cap volatility skew

Volatility skew

The below table displays the price (in basis points) and the implied volatility (%) of a 1×3 cap (2 year cap starting in one-year) for various strikes (0.1%,0.2%,0.5%,...,2%). As for the caplets (which are calls on LIBOR), cap prices increase when strike decreases because the smaller the strike is, the larger cash-flows the cap delivers.

	Price (bp)				
Cap($t = 1; T; K$)	K (%)				
T	0.1	0.2	0.5	1	2
3	280.51	260.92	202.15	105.83	1.77

Implied Volatility (%)				
K (%)				
0.1	0.2	0.5	1	2
105.60	76.04	41.14	20.12	8.16

The plot (Figure 2.5) shows the price (in basis points) of 4 caps (1×2,1×3,1×4 and 1×5) for various strikes. When the caps are in the money, the relationship between the price and the strike is almost linear. We can also observe that the price of the cap increases with maturity; this is inline with the definition of a cap price (e.g. A 1×5 cap can be decomposed into 16 caplets whereas a 1×2 cap can only be decomposed into 4 caplets; it makes sense intuitively that the price of a 1×5 cap is higher than the price of a 1×2 cap).

Volatility surface

The below table displays the price (in basis points) and the implied volatility (%) of 4 caps (1×2,1×3,1×4 and 1×5) for various strikes. As we have seen earlier, the price of a cap increases when the maturity of the cap increases and the price of cap decreases when the strike increases. On the other hand, the implied volatility of the cap decreases when the maturity of the cap increases (implied volatility is an average of the volatility across the life of the cap). The implied volatility decreases when the strike increases.

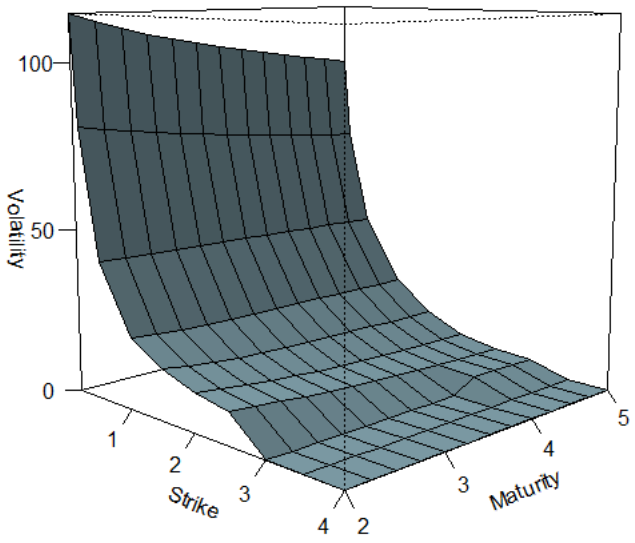

Figure 2.6: Cap volatility surface

	Price (bp)				
Cap($t=1,T,K$)	K (%)				
T	0.1	0.2	0.5	1	2
2	123.28	113.41	83.79	36.03	0.09
3	280.51	260.92	202.15	105.83	1.77
4	462.81	433.70	346.40	202.53	11.32
5	661.78	623.40	508.25	317.96	35.08

Implied Volatility (%)				
K (%)				
0.1	0.2	0.5	1	2
114.31	81.22	41.72	21.40	11.82
105.60	76.04	41.14	20.12	8.16
99.70	72.36	40.54	20.02	6.56
95.09	69.32	39.77	20.10	5.92

2.3.6 Swaption pricing

The swaptions have been priced using the formula described in the "Model and hypothesis" section. The pricing algorithm for one cap goes as follows:

1. For each simulation:
   (a) Calculate all the LIBOR rates required for the cap
   (b) Calculate the swaption price (Definition 31)
   (c) Store the swaption price
2. Once all simulations are completed, obtain the cap price by averaging the swaption price obtained for each simulation (expectation)
3. Compute the implied volatility for the swaption using Black76

Remark. The step that computes the implied volatility for swaptions has not been implemented

I have assumed quarterly payments for the swaption. The following prices have been obtained after running 30,000 simulations. I have priced swaptions for a wide range of strikes and swap tenors so that we can observe the volatility skew and the term structure.PayerSwption$(t;T;K)$ is the price as of today ($time = 0$) of a swaption with strike K that expires at time t (i.e the option expires at time t). If the swaption is exercised, the swaption payer will enter a payer swap that starts at time t and matures at times T (i.e. the swap tenor is $T-t$). This swap has n period of 3M each ($n = \frac{T-t}{4}$).

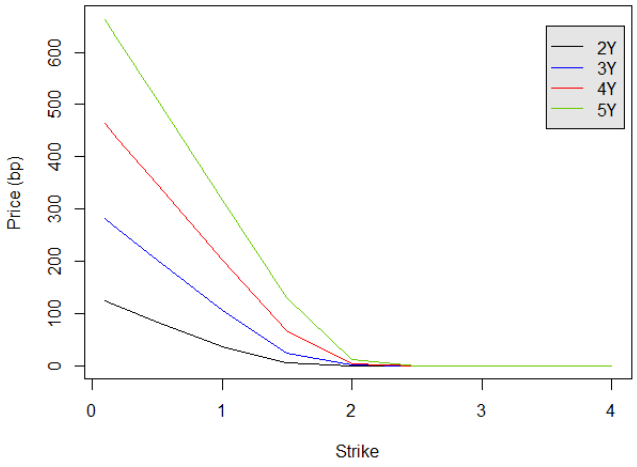

Figure 2.7: Swaption volatility skew

Volatility skew

The below table displays the price (in basis points) of a 1×3 swaption (1 year maturity option on a 2 year forward starting swap) for various strikes (0.1%,0.2%,0.5%,...,2%). Swaption prices increase when strike decreases (Figure 2.7).

	Price (bp)				
PayerSwaption$(t = 1, T, K)$	K (%)				
T	0.1	0.2	0.5	1	2
3	280.89	261.29	202.51	104.82	0.33

Volatility surface

The below table displays the price (in basis points) of 4 swaptions (1×2,1×3,1×4 and 1×5) for various strikes. As we have seen earlier, the price of a swaption increases when the tenor of the swap increases and the price of swaption decreases when the strike increases (Figure 2.8)

	Price (bp)				
PayerSwaption$(t = 1, T, K)$	K (%)				
T	0.1	0.2	0.5	1	2
2	123.51	113.63	84.00	36.58	0
3	280.89	261.29	202.51	104.82	0.33
4	463.26	434.16	346.84	201.31	2.72
5	662.26	623.87	508.70	316.76	11.60

2.4 Convergence and stability

I have run 30,000 simulations using pseudo-random numbers (rnorm native function in R) and low-discrepancy numbers (Sobol). I could not use Niederreiter numbers because their maximum dimension in NAG is 318 (for one year projection - 100 time-steps - HJM requires a dimension of $500 = 100 \times 5$ where 5 is the number of factors in use).With this data, i obtain a convergence diagram that compares the convergence of these 2 types of numbers. The simulations that use low-discrepancy numbers clearly converges faster than the simulations that use pseudo-random numbers.

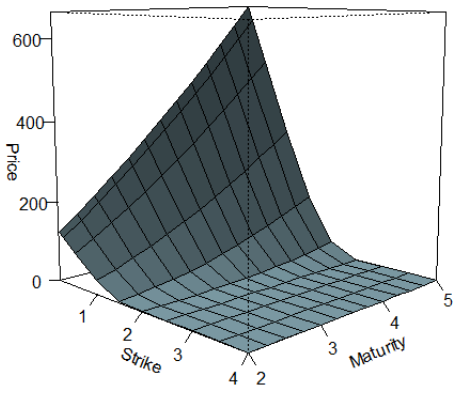

Figure 2.8: Swaption price for various swap tenors and strikes

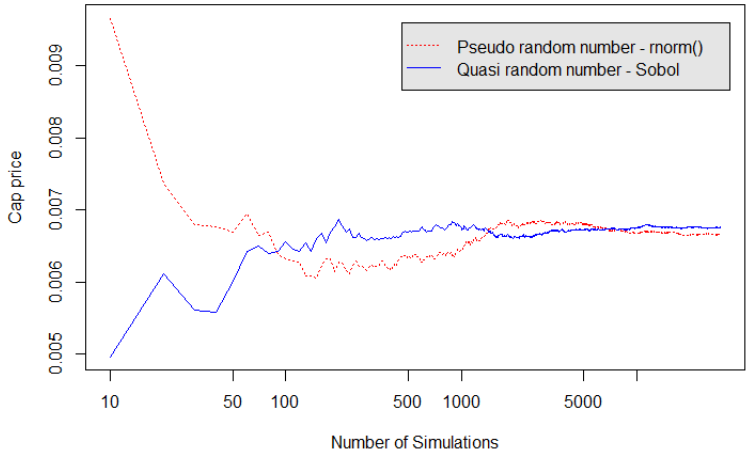

Figure 2.9: Pseudo-random vs. Sobol convergence diagram for a cap price

2.5 Numerical implementation

2.5.1 Highlights

The highlights of this implementation are:

- Integration with NAG Fortran library

  I have used the NAG Fortran library 64 bit[7] (FLW6I24DCL) because it can be integrated smoothly with R 64 bit. I have created a R function which first initialize a low-discrepancy number generator (NAG routine G05YLF) and then "draw" n numbers for each dimension (NAG routine G05YJF).

- Numerical integration for the drift

  Calculation of the HJM drift requires integration over the volatility fitted function. I have used R native numerical integration capabilities instead of using the trapezium rule integration which works but is computationally inefficient.

- Numerical integration to calculate LIBOR

  In HJM, one of the way to calculate the LIBOR rates is to integrate over the forward curve ($L(t;T) = \frac{1}{\tau}\int_0^\tau \overline{f}(t,s)\ ds$). I have fitted a cubic spline on the forward rate curve. The result is quite precise since we have a lot of data points (we have forwards for each month). Finally, i perform numerical integration over this cubic spline to calculate the LIBOR rates.

- Multithreading to improve performance

  One of the drawback of HJM is its poor performance. Populating the projection matrix for each simulation uses a lot of computing power (not to mention when we perform complex / heavy calculations with this matrix). R software is a single-threaded package; i have used R packages like doPararallel, plyr and foreach to make the calculation multithreaded so that it can use all the processors available on my computer.

- Dual-curve pricing

  I have used the UK OIS discount curve to discount cash-flows

- The core pricing function HeathJarrowMortonPricing() can price several instruments (bonds, caps and swaptions) at the same time. It makes it easy to perform analysis

- The core pricing function HeathJarrowMortonPricing() returns simulation results which can used to produce convergence diagrams

There are also improvement opportunities in this implementation:

- Complete implied volatility calculation for swaptions (it is only done for caps)

- Principal component analysis could be done on a covariance matrix which includes monthly instantaneous forwards between 0 and 5 years (instead of just semi-annual instantaneous forwards)

- Get a "longer" OIS spot curve (use OIS spread over LIBOR beyond 5 years for instance) in order to price caps and swaptions with maturities beyond 5 years

- Implement pricing algorithm in C++ to achieve better performance and get a cleaner object oriented design

2.5.2 Market data

Two types of market data have been required for this implementation:

- Historical instantaneous forward curve

- OIS spot curve

[7] This free trial license has been graciously given by the Numerical Algorithms Group

Historical instantaneous forward curve

Historical instantaneous forward rate have been downloaded from the Bank of England website[8] (BLC - Commercial bank liability curve). Data comes as two data sets: one for the short-end of the curve (0-5 years; data for each month) and one for the full curve (0-25 years; data for every six months). The one-month rate was taken from the short-end data set, the rest of the data came from the full curve data set. Minimum data clean-up (exclude public holidays) was required as the data is already cleansed by Bank of England.

OIS spot curve

Pricing is performed using 30-May-2014 as valuation date. The OIS spot curve has been downloaded from the Bank of England website (UK OIS spot curve). Discounting rates are given for tenors from 0 to 5 years (data for every six months). Liquidity in OIS contracts beyond 5 years is limited; therefore Bank of England only publishes OIS curve up to 5 years. Minimum data clean-up (exclude public holidays) was required as the data is already cleansed by Bank of England.

2.5.3 Algorithm

The algorithm implemented for this project (Interest Rate Derivatives using HJM framework) can be summarized as follows:

1. Load historical forward rates (LIBOR forward curve) and OIS discount rates (OIS spot curve) - *market_data_loading.R*
2. Perform principal component analysis - *principal_component_analysis.R*
   (a) calculate historical log differences instantaneous forward rates matrix
   (b) calculate covariance matrix
   (c) diagonalize the covariance matrix using Jacobi transformation (obtain eigenvalues and eigenvectors)
   (d) select 5 factors (principal components)
   (e) calculate the volatility vectors from the eigenvectors of the 5 factors
   (f) fit a cubic spline for each of these 5 volatility vectors
3. Monte Carlo simulation to price IR derivatives - *monte_carlo_simulation_functions.R*
   (a) generate pseudo-random numbers or low-discrepancy numbers
      i. start to run simulation
      ii. initialize and populate the HJM propagation matrix
      iii. perform instrument specific calculation (caps, swaptions) using the propagation matrix
      iv. store the results
   (b) average simulation results to get instrument price as an expectation
   (c) when required, convert instrument price to implied volatility using a root find algorithm on Black76

[8] http://www.bankofengland.co.uk/statistics/Pages/yieldcurve/archive.aspx

2.5.4 Instructions and package description

This program has been written using R (version 3.1.0, 64-bit) and Rstudio as IDE. In order to launch this program, please follow this procedure:

1. Copy the whole folder "interestrate-hjm" to your local computer (e.g. C:\temp\interestrate-hjm)
2. Open the file "interestrate-hjm\src\master_program.R" (e.g. C:\temp\interestrate-hjm\src\master_program.R)
3. Update the value of the variable **current_dir** to match your environment (e.g. current_dir = "C:/temp/interestrate-hjm/src")
4. Launch R
5. type: source ("C:/temp/interestrate-hjm/src/master_program.R") (replace the path to master_program.R to match your environment)
6. The program will start.

Folder	File	Description
.\interestrate-hjm		Top folder
.\interestrate-hjm\src		Folder containing source code (R)
	master_program.R	Main program that calls all the other programs
	class_definition.R	Defines several classes which are used throughout the project
	install_register_ packages.R	Installs and registers external packages which are required
	market_data_functions.R	Defines the function that loads the forward curve (BLC) and the OIS spot curve
	market_data_loading.R	Loads historical forward curve (short and long end) and load OIS spot curve
	jacobi_transformation _functions.R	Defines functions required for the Jacobi transformation
	principal_component _analysis.R	Performs principal component analysis
	black76.R	Defines Black76 function, cap/caplets and swaptions pricing functions that use Black76
	nag_library_wrapper.R	Defines a function that allows R to call the NAG Fortran 64 bit library
	monte_carlo _simulation_functions.R	Main pricing functions: Monte Carlo loop and bond pricing, LIBOR calculation, cap/caplet pricing and swaption pricing within one simulation matrix
	monte_carlo _simulation.R	Calls above mentioned functions; plots volatility surfaces and convergence diagrams
.\interestrate-hjm\data		Folder containing market data files
	ukblc05_mdaily_ fwdcurve_shortend.csv	BOE BLC forward curve (short end)
	ukblc05_mdaily_ fwdcurve_longend.csv	BOE BLC forward curve (long end)
	ukois09_mdaily_ spotcurve.csv	BOE OIS spot curve

3 Conclusion

Working on these two topics for the Final CQF Project has been a very exiting and enriching experience. It helps to get a "feel" of the amount of work and the type of work (research, analysis, etc.) required to implement industrial strength models in a production environment. It will certainly help me to structure my work on future assignments as quantitative analyst/developer.

References

[1] G. Darocczi, M. Puhle and E. Berlinger. Introduction to R for Quantitative Finance. Wiley Finance

[2] D. J. Duffy and A. Germani. *C# for Financial Market*. Wiley Finance

[3] J. Hull. *Options, Futures, and Other Derivatives*. Pearson Education

[4] D. Brigo and F. Mercurio. *Interest Rate Models – Theory and Practice*. Springer