Multivariate Asset Pricing

Zhen Pan

June 13, 2012

Master program Stochastics and Financial Mathematics

Supervisors:
Dr. Karma Dajani
Prof. Michel Vellekoop
MSc. Geeske Vlaming

Institute:
Universiteit Utrecht
All Options

Contents

Acknowledgements

This is my thesis for the Master program in Stochatics and Financial Mathematics. The research herein described was conducted during an internship at All Options (Amsterdam). The Master studies gave me a lot of theoretic backgrounds in mathematics as well as finance. However, I really had not understood how useful they are till I started the internship.

Working as intern at All Options has been a great experience. The project was highly related to what I studied during the master program, so I could apply most of the theory I studied. To my personal experience it is more challenging to understand and solve the practical problems than those theoretic ones. However, the theory plays a large role too.

Another great experience I had was reading piles of papers and understanding how to translate the mathematics into Matlab codes. I really appreciate that All Options gave me this opportunity. Through this internship I understood how to do research and I also improved a lot my simulation skills as well as learned a lot of basics regarding trading.

So I thank my University supervisor, professor Karma Dajani, who always encouraged and supported me as well as she believed in me.

I also would like to thank professor Michel Vellekoop and Ms. Geeske Vlaming, Alexandre Alavi, Norbert Huurnink and Sing from All Option. They have always been available to help and give me advices on my project. I appreciated Edwin de Lange, Dimitri Wesdorp, Ariyanto Gunawan, Hans de Koning, and Joost Harm, who made a cheerful atmosphere during my internship and tried to make me comfortable. I thank Ruud Groot, Peter van den Berg and Pieter Moore taught me something regarding trading.

I thank my brother Shengti and my sister-in-law Guimin who, despite the large distance separating us, always encouraged and advised me. I also thank my parents-in-law, papa and mamma, and my brother-in-law, Stefano, for the love they gave me and the enduring support.

Lastly I would like to thank my husband Davide. Whenever I had problems, he was always there to encourage and help me with great patience and love.

Zhen Pan

June 2012

Chapter 1

Introduction

There is a lot of empirical evidence showing that the skew[1] of an index is much steeper than the corresponding skews of its individual stocks. This is a very strange thing since the index is just the weighted sum of the single stocks and, because of diversification, the index is less volatile than the single stocks. by far, the real reasons have not yet been figure out. A hypothesis proposed is the leverage effect, which is used to describe the negative correlation between the stock return and the implied volatility or realized volatility. This principle says that the volatility rises when stock price drops and falls when price goes up. However, Figlewski et al.(2000)[1] strongly rejected this hypothesis in an empirical study of OEX (the Standard and Poors 100 stock index), because they found a strong leverage effect only in a crashing market and no significant effect in a market with positive returns. Their results are not consistent with the traditional leverage theory and they conclude that the "leverage effect" is actually a "down market" effect. Bakshi et al.(2003)[2] argued that the difference in the skews of the stocks and index is partly related to differences in the risk neutral distributions. The skew of single stock under risk neutral measure is less negative than the market skew, while the risk-neutral index return distribution is possible negative skewed [2] even if the physical return distribution is symmetric, which means the index skew under risk neutral measure is steeper than the market index skew. In contrast to

[1] In finance, the skew is also called volatility skew. Explicit definitions can be found in section 2.1

[2] In probability theory, the skew is a measure of the asymmetry of the probability distribution. Negative skewed means the left tail of the distribution is longer than the right tail.

Bakshi et al., Bollen et al.(2004)[3] argued that the different steepness of the skew comes from a "net buying pressure", which is referred to the difference between the number of buyer and seller's wished contracts traded everyday. They found that the most traded options of index are puts whereas for the stocks call options are traded most. So the "net buying pressure" leads to the different slopes of index and stocks skews.

Recently, Branger et al.(2004)[4] has proposed that differences in skew (or smile) are caused by differences in risk-neutral distributions rather than physical distributions. They think that the risk-neutral and physical measures have different effects on diffusion and jump parts. In particular, The volatilities and correlations for the diffusion parts will not change under both measures, but in case of jumps, the risk-neutral and physical measures have different effects on the variances and correlations. Thus the jump components will generate the difference in the skew under the risk neutral measure. Furthermore they think that the steepness of the index skew for low strikes is influenced by the correlations among its stocks during a bear market (i.e. a market with downward movements).

We think that the skew of the index can be explained by the skew of each component as well as by the correlations among them. To test our hypothesis, we had to find a model fitting very well the skew of the index and each of its components. Then we tried to simulate the index using Monte-Carlo simulation with the well fitted parameters from its components and various correlations, which may be constant or stochastic, on stock price processes. Afterwards, we tried to compare the simulated skew with the real skew of the index to see whether they had the same steepness. The thesis is arranged in seven parts: Chapter 1: Explaining the problem of this project (why is the skew of the index much steeper than its single stocks?) and what the reasons might be. Chapter 2: We explain basic concepts on options, Black-Scholes model and correlation. Chapter 3: We compare three types of models: discrete time model (GARCH model), continuous time model (Heston model) and models with a jump (Bates model). According to the advantages and drawbacks of these models, we chose the Bates model. Then we fitted the market data with the chosen model and found that it fitted very well the single stocks as well as the index. After that we tried to fit the skew surface, but we found that big errors happened on short maturities and we explained the possible reasons. Chapter 4: We firstly simulated the AEX[3] with well fitted 25 stocks and then applied constant correlation, which assumes that all cross correlations are the same and they are not strike price[4] dependent. However, we found that this assumption did not work well because the index option price was under priced for low strikes and over priced for high strikes. Then we built our own correlation models based on the implied correlation (i.e. the correlation corresponding to the "at the money" strike[5]). During the stock price simulation we gave higher correlation when the stock price $S_t < K_{at}$ (K_{at} : the strike at the money level), and vice versa. Our models improved the results, but we did not get the same steepness as the market skew and we found difficulties to guarantee the positive definiteness of the stochastic correlation models which is time dependent and all cross correlations are different. In the end we tried one type of Multivariate GARCH model (DCC model), which could easily guarantee the positive definiteness of the correlation matrix. However, even if our estimated correlation from DCC model had very the same trend as the market one (the real correlation), the steepness of the AEX skew was not improved. Chapter 5: Discussion. Chapter 6: Future study. Chapter 7: Appendix (matlab code).

[3] AEX is the dutch index, which is formed by 25 stocks.

[4] Strike price is the agreed price between the buyer and the seller for an option, more details can be found in section 2.1

[5] An option can be called "in the money", "at the money" and "out of the money" according to the relationship between stock and strike prices. For a call option, "in the money" means the stock price is larger than the strike price. "At the money" means the stock price equals the strike price whereas "Out of the money" is when the stock price is lower than the strike price. For a put option, the reverse case holds.

Chapter 2

Background on options, correlations and Monte-Carlo simulation

2.1 Put, Call options and Skew

In finance there are put and call options. A *call option* gives the holder the right to buy an asset at or before a certain date and of a certain price. While a *put option* gives the holder the right to sell an asset at or before a certain date and of a certain price. The date is called the *expiration date* or the *maturity date*. The agreed price is called *exercise price* or *strike price*. Usually the letters T and K are used to represent the maturity date and strike price respectively. At the maturity time, the call option has the payoff $max(S_T - K, 0)$, while the payoff for the put option is $max(K - S_T, 0)$. An option can be also classified as *American option* or *European option*. An *American option* can be exercised at any moment up to the expiration date, whereas a *European option* can be exercised only at the maturity date. The indices have European options, while the stocks of the index have American options. In this section, we will use the following notations:

S_0 : Price at time $t = 0$ (Spot price of the underlying)
$N(\cdot)$: The comulative distribution function of standard normal distribution
K: The strike price
r: The risk-free interest rate

T: The maturity time
σ: The volatility

The well known Black-Scholes formulas for pricing European options with non-dividend-paying stocks are as follows:

$$Call_{price} = S_0 N(X_1) - Ke^{-rt} N(X_2) \tag{2.1}$$
$$Put_{price} = Ke^{-rt} N(-X_2) - S_0 N(-X_1) \tag{2.2}$$

where

$$X_1 = \frac{log(S_0/K) + (r + \sigma^2/2)T}{\sigma\sqrt{T}}$$
$$X_2 = \frac{log(S_0/K) + (r - \sigma^2/2)T}{\sigma\sqrt{T}}$$

From the Black-Scholes pricing formula we know, the only parameter we can not observe is the volatility, but once we know the option price from the market, we can get it by inverting the Black-Scholes formula, and we call it ***implied volatility***. This is because we have $Call_{price}(orPut_{price}) = f(S_0, K, r, \sigma, T)$, where $f(S_0, K, r, \sigma, T)$ represent the right side of equation (2.1) or (2.2). Given the starting price S_0, strike K, interest rate r, maturity time T, and market Call or Put price, we can get σ by finding the root of

$$Call_{price}(orPut_{price}) - f(S_0, K, r, \sigma, T) = 0$$

So, for every strike we can get the corresponding implied volatility. If we plot the implied volatilities against the strikes we get a graph which is downward sloping, we call it the ***volatility skew***. If the graph turns up at both ends, the term ***volatility smile*** is used.

2.2 Risk neutral measure

Under the real probability measure, the expected return of the asset is not the risk-free rate. In order to avoid arbitrage, we normally pass from the real probability measure to an equivalent martingale measure[6] (i.e. the risk neutral measure), such that the discounted process of the equity under this measure is a martingale and expected return of the asset is the risk free rate.

Definition 2.2.1. *A family of $\sigma-$algebras $\{\mathcal{F}_t, t \geq 0\}$ is called a filtration of a probability space $(\Omega, \mathcal{F}, P)$ if $\mathcal{F}_t \subset \mathcal{F}$ for all t and $\mathcal{F}_s \subset \mathcal{F}_t$ for $s \leq t$. A stochastic process $\{X_t : t \geq 0\}$, is said to be adapted to $\{\mathcal{F}_t : t > 0\}$ if for each t, the random variable X_t is $\mathcal{F}_t$-measurable.*

Definition 2.2.2 (Conditional expectation). *Let $(\Omega, \mathcal{F}, P)$ be a triple and $X \in \mathcal{L}^1(\Omega, \mathcal{F}, P)(\mathbb{E}(|X|) < \infty)$. $\mathcal{G}$ is a sub-σ-algebra of $\mathcal{F}$, then a random variable Y is called a* ***conditional expectation*** *of X given $\mathcal{G}$, symbolically $\mathbb{E}(X|\mathcal{G}) := Y$, if :*
(i) Y is $\mathcal{G}$ measurable.
(ii) For any $G \in \mathcal{G}$, we have $\mathbb{E}(X 1_G) = \mathbb{E}(Y 1_G)$.
For $A \in \mathcal{G}$, $P(A|\mathcal{G}) = \mathbb{E}(1_A|\mathcal{G})$ is called a conditional probability of A given the σ-algebra $\mathcal{G}$.

Definition 2.2.3. *Let X_t be a stochastic process adapted to a filtration $\mathcal{F}_t$ and $E|X_t| < \infty$ for all $t \in T$. Then X_t is called a martingale with respect to $\mathcal{F}_t$ if for any $s \leq t$ in T,*

$$E[X_t|\mathcal{F}_s] = X_s, \quad a.s.\ (almost\ \ surely).$$

Definition 2.2.4 (Brownian motion). A process $W = (W_t : t \geq 0)$ is Brownian motion if:

1. W_t is continuous and $W_0 = 0$.
2. W_t is $\mathcal{N}(0, t)$ distributed.
3. For $0 \leq s \leq t$, $W_t - W_s$ is independent of $\{W_u : u \leq s\}$ and it is distributed as $\mathcal{N}(0, t - s)$.

Theorem 2.2.1: (Girsanov Theorem) *Let $(\Omega, \mathcal{F}, \{\mathcal{F}_t\}_{t\in[0,T]}, P)$ be a filtered probability space, where $0 < T < \infty$ and we assume for convenience*

[6] Suppose P and $\mathbb{Q}$ are two probability measure defined on the same measurable space $(\Omega, \mathcal{F})$. We say $\mathbb{Q}$ is equivalent to P ($\mathbb{Q} \approx P$), if for any A in $\mathcal{F}$, $\mathbb{Q}(A) = 0$ if and only if $P(A) = 0$. We also say $\mathbb{Q}$ is absolutely continuous with P.

that $\mathcal{F} = \mathcal{F}_T$. Let W_t be an $(\mathcal{F}_t, P)$-Brownian motion. Let $\theta(t)$ be an adapted process satisfying $E[exp(\frac{1}{2}\int_0^T \theta^2(s)\,\mathrm{d}s)] < \infty$ a.s. and define

$$Z_T = exp(\int_0^T \theta(s)\,\mathrm{d}W_s - \frac{1}{2}\int_0^T \theta^2(s)\,\mathrm{d}s)$$

Suppose that $E(Z_T) = 1$, and define a measure $\mathbb{Q}$ on $(\Omega, \mathcal{F})$ by $\frac{\mathrm{d}\mathbb{Q}}{\mathrm{d}P} = Z_T$, then under measure $\mathbb{Q}$, the process $\tilde{W}_t$ defined by

$$\tilde{W}_t = W_t - \int_0^t \theta(s)\,\mathrm{d}s$$

is an $\mathcal{F}_t$ Brownian motion.

2.3 Black-Scholes model

In Black-Scholes's model, the asset price process (S_t) is assumed to follow geometric Brownian motion, i.e.

$$S_t = S_0 e^{\mu t + \sigma W_t}$$

Where μ is the drift, and σ is the volatility which is assumed to be constant. W_t is the Brownian motion part, which is distributed as $\mathcal{N}(0, t)$, and for any $s < t$, $W_t - W_s$ is normally distributed with mean 0 and variance $t - s$. Under real probability, the Stochastic Differential Equation(SDE) of S_t is:

$$dS_t = S_t \mu dt + S_t \sigma dW_t \tag{2.3}$$

Let us define the discounted asset price as $\tilde{S}_t$, then $\tilde{S}_t = e^{-rt}S_t$. So the SDE of $\tilde{S}_t$ is:

$$\begin{aligned} d\tilde{S}_t &= e^{-rt}dS_t + (-r)e^{-rt}S_t dt \\ &= e^{-rt}(S_t\mu dt + S_t\sigma dW_t) - re^{-rt}S_t dt \\ &= e^{-rt}S_t[(\mu - r)dt + \sigma dW_t] \end{aligned}$$

By $\tilde{S}_t = e^{-rt}S_t$, we have:

$$d\tilde{S}_t = \tilde{S}_t[(\mu - r)dt + \sigma dW_t] \tag{2.4}$$

Setting $\tilde{W}_t = W_t - \frac{(r-\mu)t}{\sigma}$, then by Girsanov theorem $\tilde{W}_t$ is a new Brownian motion under $\mathbb{Q}$. So we can rewrite equation (2.4) as: $d\tilde{S}_t = \tilde{S}_t \sigma d\tilde{W}_t$. Since $E^{\mathbb{Q}}(S_t - S_s|\mathcal{F}_s) = E^{\mathbb{Q}}(\int_s^t \tilde{S}_u \sigma \, \mathrm{d}\tilde{W}_u|\mathcal{F}_s)$ and $E^{\mathbb{Q}}(\int_s^t \tilde{S}_u \sigma \, \mathrm{d}\tilde{W}_u|\mathcal{F}_s) = 0$ almost surely. so $\tilde{S}_t$ under measure $\mathbb{Q}$ is a martingale process.
Then dS_t can be rewritten as:

$$\begin{aligned}
dS_t &= d(e^{rt}\tilde{S}_t) \\
&= e^{rt} d\tilde{S}_t + \tilde{S}_t d(e^{rt}) \\
&= e^{rt}\tilde{S}_t \sigma d\tilde{W}_t + re^{rt}\tilde{S}_t dt \\
&= S_t \sigma d\tilde{W}_t + rS_t dt
\end{aligned}$$

If we use $\mathbb{Q}$ to denote the risk neutral measure and V_t represents the option value at time t, then the expected payoff of the the option at any time t is :

$$V_t = E^{\mathbb{Q}}[e^{-r(T-t)}V_T|\mathcal{F}_t]$$

Theorem 2.3.1 (*Itô's* **formula**) *If the stochastic process S_t satisfies the SDE(2.3) and $\phi : R \to R$ is twice continuously differentiable, then*

$$d\phi(S_t) = \phi'(S_t)dS_t + \frac{1}{2}\phi''(S_t)d[S]_t \tag{2.5}$$

Let $\phi(S_t) = log(S_t)$ and apply theorem 2.3.1 we get:

$$\begin{aligned}
dlog(S_t) &= \frac{1}{S_t}dS_t - \frac{1}{2S_t^2}d[S]_t \\
&= \frac{1}{S_t}(S_t\sigma d\tilde{W}_t + rS_t dt) - \frac{1}{2S_t^2}(S_t^2\sigma^2 d[\tilde{W}]_t) \\
&= (r - \frac{\sigma^2}{2})dt + \sigma^2 d\tilde{W}_t \qquad (2.6)
\end{aligned}$$

In the discrete case, equation (2.6) becomes

$$log(S_{t+1}) - log(S_t) = (r - \frac{\sigma^2}{2})\Delta t + \sigma\Delta\tilde{W}_t \tag{2.7}$$

Since $\Delta\tilde{W}_t$ is distributed as $\mathcal{N}(0,t)$, and $\sqrt{\Delta t}Z_t$ is also $\mathcal{N}(0,t)$ distributed(where Z_t has standard normal distribution), we can write $\Delta\tilde{W}_t = \sqrt{\Delta t}Z_t$. Then equation (2.7) can be rewrite as:

$$S_{t+1} = S_t exp((r - \frac{\sigma^2}{2})\Delta t + \sigma\sqrt{\Delta t}Z_t) \tag{2.8}$$

Remark 2.3.1: $[S]_t$ is the quadratic variation, which is defined as $[S]_t = \lim_{\delta_k \to 0} \sum_{n=1}^{k} (S_{t_n}^k - S_{t_{n-1}}^k)^2$, where the limits are taken over the partitions: $0 = t_0^k < t_1^k < t_2^k \cdots < t_k^k = t$, with $\delta_k = max_{\{1 \leq n \leq k\}}(t_n^k - t_{n-1}^k)$. For Brownian motion we have $d[\tilde{W}]_t \xrightarrow{P} dt$(converge in probability), so $d[S]_t \xrightarrow{P} S_t^2 \sigma^2 dt$.

2.4 "Weights" of the index

The weights of the stocks in the index are always changing but the number of shares of each stock in the index stays the same for several months. So, instead of using real weights, one normally uses the number of shares as "weights". The spot price of the index is calculated by $S_0^{index} = (\sum_i^n S_i N_i)/divisor$, where n refers to the number of stocks inside the index, while N is the number of shares of the individual stock in the index. The divisor is a real number which is normally known or we can calculate it directly from the known stock prices of the index and its components. Taking the AEX (includes 25 stocks) for example $S_0^{AEX} = (\sum_{i=1}^{25} S_i N_i)/divisor$, and the $divisor = (\sum_{i=1}^{25} S_i N_i)/S_0^{AEX}$. These two expressions can be applied to most indices with a certain number of stocks. Sometimes it is difficult to work with high dimensional cases and to make things simple, we try to select some stocks from the index. However, doing this, we would like the synthetic index and the true index to have the same stock price at time 0. For example, let us choose stock RDSA (Royal dutch shell), ING (ING bank), PHI (Philips) to form a synthetic AEX, and we use $N_{ING}, N_{RDSA}, N_{PHI}$ to represent the orginal number of shares that the corresponding stocks have in the real AEX. To keep the S_0^{AEX} unchanged, we firstly calculate

$$divisor = (S_{ING} * N_{ING} + S_{RDSA} * N_{RDSA} + S_{PHI} * N_{PHI})/S_0^{AEX}$$

then the new number of shares of the corresponding stocks of the synthetic AEX are as follows:

$$N_{ING}^{new} = N_{ING}/divisor$$

$$N_{RDSA}^{new} = N_{RDSA}/divisor$$
$$N_{PHI}^{new} = N_{PHI}/divisor$$

2.5 Theory on correlation

The key in understanding the price process of the index is to study the correlations of its components. No matter what kind of correlation models we use or build, we have to guarantee the positive definiteness of the correlation matrix, because we have to use Cholesky decomposition to decompose the correlated Gaussian random variable to uncorrelated Gaussian random variables in the simulation, and the condition for this application is that the matrix should be positive definite. From [5], we know that it is not easy to guarantee the positive definiteness of the correlation matrix, especially in high dimensional problems. For example, if we have n stocks , then the correlation matrix is $n \times n$ dimension. If we would like to estimate the correlations, then we have $\frac{n(n-1)}{2}$ unknown parameters.

Theorem 2.5.1 : (Cholesky decomposition) *If C is positive definite, then C can be factored uniquely in the form $C = A^T A$, where A is the upper triangular matrix with positive diagonal elements.*

Since correlation matrices are symmetric matrices, in this thesis we will only talk about positive definiteness of a symmetric matrix. From the basic Algebra, we know that:

a symmetric matrix C is positive definite is equivalent to the following arguments:

- *If and only if for any vector $y \neq 0$, $y^T C y > 0$.*
- *If and only if It's determinant is positive.*
- *If and only if all its eigenvalues are positive.*
- *If and only if there exist a nonsingular matrix Q, such that $C = Q^T Q$, where Q is the upper triangular matrix.*

An example on applying Cholesky decomposition

Applying (2.8) to three dimensions, we have :

$$S_{t+1}^{(1)} = S_t^{(1)} exp((r - \frac{\sigma_1^2}{2})\Delta t + \sigma_1\sqrt{\Delta t}Z_t^{(1)})$$

$$S_{t+1}^{(2)} = S_t^{(2)} exp((r - \frac{\sigma_2^2}{2})\Delta t + \sigma_2\sqrt{\Delta t}Z_t^{(2)})$$

$$S_{t+1}^{(3)} = S_t^{(3)} exp((r - \frac{\sigma_3^2}{2})\Delta t + \sigma_3\sqrt{\Delta t}Z_t^{(3)})$$

Here $Z_t^{(1)}$, $Z_t^{(2)}$ and $Z_t^{(3)}$ are correlated Gaussian random variables which have standard normal distribution. Define 3×1 vectors $Z_t = (Z_t^{(1)}, Z_t^{(2)}, Z_t^{(3)})'$ and $\epsilon_t = (\epsilon_t^{(1)}, \epsilon_t^{(2)}, \epsilon_t^{(3)})'$, where $\epsilon_t^{(i)}, (i = 1, 2, 3)$ are i.i.d standard normal distributed. Suppose there exists an 3×3 non-singular matrix A such that $Z = A\epsilon$. Then from the Theorem 2.1 of [5], we know that $var(Z) = var(A\epsilon) = Acov(\epsilon)A'$. Let us use Σ and C to denote the covariance and correlation matrix of Z respectively, and let $D_{ii} = \sqrt{var(Z_i)}$, then we have $\Sigma = DCD$. Since all $Z_t^{(i)}$, $\epsilon_t^{(i)}$ are standard normally distributed, so

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = cov(\epsilon)$$

Then we get $\Sigma = C = AA'$. If we use $\rho_t^{(i,j)}$ denote the correlation between random variable $Z_t^{(i)}$ and $Z_t^{(j)}$, then we can write $Z_t^{(1)}$, $Z_t^{(2)}$, $Z_t^{(3)}$ as:

$$\begin{aligned} Z_t^{(1)} &= a_{11}\epsilon_t^{(1)} \\ Z_t^{(2)} &= a_{21}\epsilon_t^{(1)} + a_{22}\epsilon_t^{(2)} \\ Z_t^{(3)} &= a_{31}\epsilon_t^{(1)} + a_{32}\epsilon_t^{(2)} + a_{33}\epsilon_t^{(3)} \end{aligned}$$

Where $\epsilon_t^{(1)}$, $\epsilon_t^{(2)}$ and $\epsilon_t^{(3)}$ are uncorrelated random variables from standard normal distribution, a_{ij} satisfies $\sum a_{ij}^2 = 1$, for $j \le i$ and $\sum_{k=1}^{j} a_{ik}a_{jk} = \rho_t^{(ij)}$, (for $j \le i$).

$$A = \begin{pmatrix} a_{11} & & \\ a_{21} & a_{22} & \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$$

A is the transpose of the upper triangular matrix from the Cholesky decomposition. Therefore, given an positive definite correlation matrix, we can always find such A by Cholesky decompsiton and then we can decompose the correlated random variables $Z_t^{(i)}$ to linear combinations of uncorrelated random variables $\epsilon_t^{(i)}$. If the correlation does not change with time, then A remains unchanged, otherwise A should be written as A_t and the entries of A should be also updated with time.

2.6 Monte-Carlo simulation

Normally, one uses the tree or Monte-Carlo methods to simulate option price, but the latter method tends to be numerically more precise than the tree method. Monte Carlo simulation method is used under risk neutral measure. So, before applying it, we should get the stochastic process under $\mathbb{Q}$ measure, then we discritize the time interval $[t_0, T]$ as $t_0 = \tau_0 < \tau_1 < \tau_2 < \cdots < \tau_N = T$, and simulate the sample paths of the discrete time to approximate the paths of continuous time. The main steps are as follows.

- Step 1: simulate N sample paths and get payoff's at maturity time T by $C_i = max(S_T^i - K, 0)$ if it is a European call option.
- Step 2: take the average of the summed payoff's by $\bar{C} = \frac{\sum_{i=1}^{N} C_i}{N}$
- Step 3: discount $\bar{C}$ to time zero by $C_0 = e^{-rt}\bar{C}$

Suppose the payoff's follow a certain distribution and we use m denote the mean of this distribution. Let us use S_{dev} denote the standard deviation of the payoffs from the simulated sample paths, then by $\frac{\sqrt{N}(\bar{C}-m)}{S_{dev}} \sim \mathcal{N}(0,1)$, we know that a 95% confidence interval for the $\bar{C}$ is:

$$m - \frac{1.96 S_{dev}}{\sqrt{N}} < \bar{C} < m + \frac{1.96 S_{dev}}{\sqrt{N}}$$

So the standard error for the $\bar{C}$ is $\frac{S_{dev}}{\sqrt{N}}$. Since the option price is just the discounted payoff, the standard error for the C_0 is also $\frac{S_{dev}}{\sqrt{N}}$. So, the more trials we do, the less error we will get. However, the drawback of the Monte-Carlo simulation is that it is a very time consuming method. For example, in Chapter 4, the correlaiton model 1, 2, 3 normally take 1 to 2 hours and the stochastical correlation model takes day and night if we do 500,000 simulations.

Chapter 3

Fitting skew of stocks and AEX

3.1 Model selection

In Black-Scholes model, the volatility is assumed to be constant, but it is often observed that the volatility tends to change with time, so there are a lot of papers proposing models with stochastic volatility. However, they can be mainly divided into three types: the Garch-family models, bivariate diffusion models (such as Heston model) and models with jumps. Normally one prefers a pure diffusion process because it has advantages in constructing closed form solutions for European options prices, and it also has continuous sample path and satisfies the Markov property. However, in practice, especially in crash moments, a stock price process is not continuous. Furthermore, their Markovian structure normally contradicts the emperical findings, because we often see that the log returns depend on its past. So, in the past few years there was a growing interest in pricing with the Garch-family model because it is non-Markovian and can explain the autocorrelations of the log-return as well as the volatility clustering (i.e. a big fluctuation followed by a big fluctuation and a small fluctuation followed by a small fluctuation). The drawbacks of this kind of model are that it is discrete, analytic solution for the option pricing normally do not exist, thus it should be done by numeric algorithm [6], and it has infinite risk-neutral measures, so the difficulty in option-pricing with GARCH is the identification of a risk-neutral probability measure. It is believed that the reason of the non-uniqueness of the risk-neutral measure is that the market is not complete [7][8][9]. Normally, in a complete market every agent is able to exchange assets, directly or indirectly, with every other agent without transaction costs. For the incomplete markets, the shortage of securities will likely restrict individuals from transferring the desired level of wealth among states. By far, the common used risk-neutral measure methods are the Local Risk Neutral Valuation Principle (LRNVR), the conditional Esscher transform and the Extended Girsanov Principle. Although for Gaussian distributed innovations(noise) all the different risk neutral measures reach the same result, the skewness and the leptokurtosis[7]of returns cannot be well explained by GARCH models with normally distributed innovations(noise)[10], and there is few literature about the option pricing of GARCH models with non-normal distributed innovations(residuals). Then, there are a lot of references saying that a jump model works very well and fits better than the other two models [11][12][13]. The classic jump models are the Merton jump model and the Bates model. The former one adds an independent jump to the Black-Sholes model, while in the later one adds an independent jump to the Heston model. Since Heston model is able to generate skew and is the most used recently, we choose Bates model as our trial. In Chapter (4), we will see that this model works well and fits almost perfectly the skew for each stock.

3.1.1 GARCH model

GARCH model is usually used to capture the volatility clustering of the returns and it is assumed that the volatility at time t depends on its past residuals and volatilities. Here we use r_t denote the return at time t and $r_t = log(\frac{S_t}{S_{t-1}})$. Let $\mathcal{F}_{t-1} = \sigma\{r_s : s < t\}$. The residual (or noise) is defined by $\epsilon_t = r_t - E(r_t|\mathcal{F}_{t-1})$. We call σ_t^2 the variance conditioned on $\mathcal{F}_{t-1}$, which is $E(\epsilon_t^2|\mathcal{F}_{t-1})$. People normally assume that the noise is normally distributed, so $\epsilon_t|\mathcal{F}_{t-1} \sim \mathcal{N}(0, \sigma_t^2)$. The standardized residual $Z_t = \frac{\epsilon_t}{\sigma_t}$ and $Z_t \sim \mathcal{N}(0, 1)$.

[7]**Kurtosis** measures the peakedness of a distribution. If we use μ_x, and σ_x denote the mean and standard deviation of a random variable X, the kurtosis is defined by $K(x) = E[\frac{(X-\mu_x)^4}{\sigma_x^4}]$. The kurtosis for a normal distributed random variable is 3. **Exess kurtosis=kurtosis-3**, so the excess kurtosis for a normal distributed random variable is zero. If a return distribution has positive excess kurtosis compared to the normal distribution, then we say this distribution is leptokurtotic or it exhibits **leptokurtosis**.

The general GARCH(p, q) process is defined by

$$\sigma_t^2 = \alpha_0 + \sum_{i=1}^{p} \alpha_i \epsilon_{t-i}^2 + \sum_{i=1}^{q} \beta_i \sigma_{t-i}^2$$

The α_0, $\alpha_i' s$, $\beta_i' s$ are non-negative. In oder to have stationary solution[8] for the above process, we need $\sum_{i=1}^{p} \beta_i + \sum_{i=1}^{q} \alpha_i < 1$.
In finance it is observed that the return is negative correlated with the volatility and we call this as leverage effect [9], so here we only introduce GARCH models which reflect the leverage, specifically the GARCH models with lag one[10]. we let $h_t = \sigma_t^2$.

EGARCH(1, 1):	$logh_t = \alpha_0 + \alpha_1 logh_{t-1} + \beta_1[\lvert Z_{t-1}\rvert - E\lvert Z_{t-1}\rvert] + \beta_2 Z_{t-1}$
GJR-GARCH(1, 1):	$h_t = \alpha_0 + \alpha_1 h_{t-1} + \beta_1 \epsilon_{t-1}^2 + \beta_2 \epsilon_{t-1}^2 1_{\epsilon_{t-1} \leq 0}$
NGARCH(1, 1):	$h_t = \alpha_0 + \alpha_1 h_{t-1} + \beta_1 h_{t-1}(Z_{t-1} - \theta)^2$
TGARCH(1, 1):	$\sqrt{h_t} = \alpha_0 + \alpha_1 \sqrt{h_{t-1}} + \beta_1 \lvert\epsilon_{t-1}\rvert + \beta_2 max(-\epsilon_{t-1}, 0)$

It is easy to see the leverage effect from GJR-GARCH(1, 1) and TGARCH(1, 1), because the volatilities increase when bad news happens ($\epsilon_t < 0$). In NGARCH(1, 1) we need $\theta > 0$ to guarantee that leverage exists, so when the standardized noise Z_{t-1} is less than zero, the volatility increases. In order to have leverage effect in EGARCH(1,1), we need $\beta_2 < 0$. Then we can easily see that volatility increases when bad news happens ($Z_t < 0$) and decreases when good news happens ($Z_t > 0$). We should notice that $Z_t \sim \mathcal{N}(0, 1)$, so $E|Z_{t-1}|$ is a constant and equal to $\sqrt{\frac{\pi}{2}}$.
All the above GARCH processes share the same log return processes under real probability measure, which is:

$$ln(\frac{S_t}{S_{t-1}}) = r - \frac{1}{2}h_t + \lambda\sqrt{h_t} + \epsilon_t$$

[8]Stationary solution means that distribution of the variable $(\epsilon_t, \epsilon_{t-1} \cdots, \epsilon_L)$ does not depend on starting time t but only the time length L

[9]Leverage effect means negative returns generate more volatility than positive returns.

[10]Lag one means the variance at time t depends on the history at time $t - 1$

Where r is the risk free rate and λ is the risk premium[11].
As mentioned earlier, in GARCH-family model there is no unique risk neutral measure form. Currently the most known risk neutral measure methods are: LRNVR, the conditional Esscher transform, and the Extended Girsanov principle. Under the assumption of normally distributed noise, all the three different risk neutral measure methods reach the same result. In other words, they all have mean $r - \frac{h_t}{2}$ and variance h_t conditioned on $\mathcal{F}_{t-1}$. So, Under risk-neutral measure, all the GARCH processes take the same form:

$$S_t = S_{t-1}exp[(r - \tfrac{1}{2}h_t) + \zeta_t], \text{ and } \zeta_t|\mathcal{F}_{t-1} \sim \mathcal{N}(0, h_t),\ \zeta_t = \epsilon_t + \lambda\sqrt{h_t}.$$

The corresponding proof can be found in [14].

3.1.2 Heston model

Under the real probability $\mathbb{P}$, the Heston Model [15] is as follows:

$$\begin{aligned} \mathrm{d}S_t &= \mu S_t \mathrm{d}t + \sqrt{V_t} S_t \mathrm{d}B_s(t) \\ \mathrm{d}V_t &= \kappa(\theta - V_t)\mathrm{d}t + \sigma_v \sqrt{V_t}\mathrm{d}B_v(t) \end{aligned}$$

Where μ is the expected return under real probability, S_t is stock price at time t, and V_t is the volatility at time t. $B_s(t)$ and $B_v(t)$ are the Brownian motion processes of stock price and volatility respectively. κ is the mean reverting parameter and θ is the long run variance. ρ is the correlation between $B_s(t)$ and $B_v(t)$. $d < B_s(t), B_v(t) >= \rho dt$.

3.1.3 Bates model

Bates model only adds an independent jump on Heston model, so under real probability, the Bate model is as follows:

$$dS_t = (\mu - \lambda\mu_J)S_t dt + \sqrt{V_t} S_t dB_s(t) + J_t S_t dN_t \tag{3.1}$$

$$dV_t = \kappa(\theta - V_t)dt + \sigma_v\sqrt{V_t}dB_v(t) \tag{3.2}$$

[11]Risk premium is the amount that the expected return of an asset exceeds the risk free rate. For example, if $E(r_s)$ is the expected return of a stock, then the risk premium is $E(r_s) - r$.

μ, κ, θ, σ_v, ρ, B_s, B_v are the same as in Heston model. N_t is Poisson count process with intensity λ. The probability to have a jump of size one on a small time interval dt is λdt, and the logarithm of the jump size J_t is distributed as Gaussian

$$log(1 + J_t) \sim N(log(1 + \mu_J) - \frac{(\sigma_J^2)}{2}, \sigma_J^2)$$

3.1.4 Deriving characteristic function of Bates model

We can guess a solution for the Bates Model which is analogue to the Black-Scholes formula:

$$C(S_t, V_t, K, T) = S_t \mathcal{N}_1^{Bates} - Ke^{-r(T-t))} \mathcal{N}_2^{Bates} \tag{3.3}$$

Where $\mathcal{N}_1^{Bates}$ and $\mathcal{N}_2^{Bates}$ are the probabilities that S_T ends in the money, It is the probability of $S_T > K$. Normally people solve $\mathcal{N}_1^{Bates}$, $\mathcal{N}_2^{Bates}$ through their characteristic function as it is difficult to solve them directly.

Definition 3.1 (Characteristic Function). *Let X be a random variable with distribution μ, then for every $t \in \mathbb{R}$, the characteristic function of X is defined by: $\varphi_X(t) = \mathbb{E}(e^{itX}) = \int_{-\infty}^{\infty} e^{itX} \mu(\mathrm{d}x)$. If Y is a random variable, then the conditional characteristic function of Y conditioned on X is $\varphi_{Y|X}(t) = \mathbb{E}[e^{itY}|X = x]$.*

Conditioning on $logS_t, V_t(t \leq T)$, the probabilities $\mathcal{N}_j^{Bates}$ can be written as:

$$\mathcal{N}_j^{Bates}(u, V, T, K) = P[logS_T \geq log(K)|logS_t = u, V_t = V] \tag{3.4}$$

By the definition of conditional expectation (definition 2.2.2), we know that $\mathcal{N}_j^{Bates}$ are functions of $logS_t, V_t$, then characteristic function of $logS_T$ corresponding to $\mathcal{N}_j^{Bates}$ are also function of $logS_t, V_t$ and we can write them as $\varphi_j^{Bates}(u, V, t, y)$. For handy notation we write $\varphi_j^{Bates}(u, V, t, y)$ as $\varphi_j^{Bates}(y)$, then $\varphi_j^{Bates}(y)$ is:

$$\varphi_j^{Bates}(y) = \mathbb{E}[e^{iylogS_T}|logS_t = u, V_t = V] \tag{3.5}$$

From equation (3.1) we know that $logS_t$ has the Brownian motion part $(\mu - \lambda\mu_J)t + \sqrt{V_t}B_s(t)$ and the jump part J_tN_t. Since these two parts are independent, then the characteristic function can be expressed as

$$\varphi_j^{Bates}(y) = \varphi_j^B(y)\varphi_j^{Jump}(y) \tag{3.6}$$

where $\varphi_j^B(y)$ is the characteristic function for the Brownian motion part. Let $\varphi_j^{Heston}(y)$ denote the characteristic functions for Heston model, then we will see $\varphi_j^B(y)$ is almost the same as $\varphi_j^{Heston}(y)$ except for the $log(S_t)$ in $\varphi_j^B(y)$ has an additional constant term $-\lambda\mu_J$. Therefore, firstly we try to derive $\varphi_j^{Heston}(y)$ and $\varphi_j^B(y)$ can be easily derived.
We guess an option pricing formula for Heston model:

$$C(S_t, V_t, K, T) = S_t\mathcal{N}_1 - Ke^{-r(T-t))}\mathcal{N}_2 \tag{3.7}$$

Where $\mathcal{N}_1$ and $\mathcal{N}_2$ are the probabilities of Heston model that S_T ends in the money.
By the fact that the value of any asset $P(S, V, t)$ must satisfy the following PDE (Partial Differential Equation):

$$\begin{aligned} \frac{1}{2}VS^2\frac{\partial^2 P}{\partial S^2} + \rho\sigma_v VS\frac{\partial^2 P}{\partial S\partial V} + \frac{1}{2}\sigma_v^2 V\frac{\partial^2 P}{\partial V^2} \\ +rS\frac{\partial P}{\partial S} + \{\kappa(\theta - V) - \lambda(S, V, t)\}\frac{\partial P}{\partial V} - rP + \frac{\partial P}{\partial t} = 0 \end{aligned} \tag{3.8}$$

$\lambda(S, V, t)$ is the risk premium. According to [16], the risk premium is proportional to V, so here we set $\lambda(S, V, t) = \beta V$. For ease of notation, we use S and V instead of S_t and V_t. Then the equation (3.8) can be rewritten as :

$$\begin{aligned} \frac{1}{2}VS^2\frac{\partial^2 P}{\partial S^2} + \rho\sigma_v VS\frac{\partial^2 P}{\partial S\partial V} + \frac{1}{2}\sigma_v^2 V\frac{\partial^2 P}{\partial V^2} \\ +rS\frac{\partial P}{\partial S} + \{\kappa\theta - (\kappa + \beta)V\}\frac{\partial P}{\partial V} - rP + \frac{\partial P}{\partial t} = 0 \end{aligned} \tag{3.9}$$

Define $f_1 = S_t\mathcal{N}_1$ and $f_2 = -Ke^{-r(T-t))}\mathcal{N}_2$, then $C(S_t, V_t, K, T) = f_1 + f_2$. Since $C(S_t, V_t, K, T)$ must satisfy (3.9) and f_1, f_2 do not have shared terms, it means f_1, f_2 both should satisfy the PDE in (3.9). Define $u = logS_t$, then $\frac{\partial \mathcal{N}_1}{\partial S} = \frac{\partial \mathcal{N}_1}{\partial u}\frac{\partial u}{\partial S} = \frac{\partial \mathcal{N}_1}{\partial u}\frac{1}{S}$. The corresponding derivatives from f_1 are :

$$
\begin{align}
\frac{\partial f_1}{\partial S} &= \mathcal{N}_1 + S\frac{\partial \mathcal{N}_1}{\partial S} = \mathcal{N}_1 + S\frac{\partial \mathcal{N}_1}{\partial u}\frac{1}{S} = \mathcal{N}_1 + \frac{\partial \mathcal{N}_1}{\partial u} \tag{3.10}\\
\frac{\partial^2 f_1}{\partial S^2} &= \frac{\partial \mathcal{N}_1}{\partial S} + \frac{\partial^2 \mathcal{N}_1}{\partial^2 u}\frac{1}{S} = \frac{\partial \mathcal{N}_1}{\partial u}\frac{1}{S} + \frac{\partial^2 \mathcal{N}_1}{\partial^2 u}\frac{1}{S} \tag{3.11}\\
\frac{\partial^2 f_1}{\partial S \partial V} &= \frac{\partial \mathcal{N}_1}{\partial V} + \frac{\partial^2 \mathcal{N}_1}{\partial u \partial V} \tag{3.12}\\
\frac{\partial^2 f_1}{\partial V^2} &= S\frac{\partial^2 \mathcal{N}_1}{\partial V^2} \tag{3.13}\\
\frac{\partial f_1}{\partial V} &= S\frac{\partial \mathcal{N}_1}{\partial V} \tag{3.14}\\
\frac{\partial f_1}{\partial t} &= S\frac{\partial \mathcal{N}_1}{\partial t} \tag{3.15}
\end{align}
$$

Using the results of equation (3.10)-(3.15) to replace the corresponding terms in equation (3.9), we get:

$$
\begin{aligned}
\frac{1}{2}VS^2\{\frac{1}{S}(\frac{\partial \mathcal{N}_1}{\partial u} + \frac{\partial^2 \mathcal{N}_1}{\partial u^2})\} + \rho\sigma_v VS(\frac{\partial \mathcal{N}_1}{\partial V} + \frac{\partial^2 \mathcal{N}_1}{\partial u \partial V}) + \frac{1}{2}\sigma_v^2 VS\frac{\partial^2 \mathcal{N}_1}{\partial V^2}\\
+rS(\mathcal{N}_1 + \frac{\partial \mathcal{N}_1}{\partial u}) + \{\kappa\theta - (\kappa+\beta)V\}S\frac{\partial \mathcal{N}_1}{\partial V} - rS\mathcal{N}_1 + S\frac{\partial \mathcal{N}_1}{\partial t} = 0
\end{aligned}
$$

Since stock price $S \neq 0$, we can omit it from the above equation. Then we get the PDE for $\mathcal{N}_1$ is:

$$
\begin{aligned}
\frac{1}{2}V\frac{\partial^2 \mathcal{N}_1}{\partial^2 u} + \rho\sigma_v V\frac{\partial^2 \mathcal{N}_1}{\partial u \partial V} + \frac{1}{2}\sigma_v^2 V\frac{\partial^2 \mathcal{N}_1}{\partial V^2}\\
+(r + \frac{1}{2}V)\frac{\partial \mathcal{N}_1}{\partial u} + \{\kappa\theta - (\kappa+\beta-\rho\sigma_v)V\}\frac{\partial \mathcal{N}_1}{\partial V} + \frac{\partial \mathcal{N}_1}{\partial t} = 0
\end{aligned} \tag{3.16}
$$

By the same way, we can get the corresponding derivatives of f_2. For convinience, let us set $K_t = -Ke^{-r(T-t)}$, so we have

$$\frac{\partial f_2}{\partial S} = K_t \frac{\partial \mathcal{N}_2}{\partial S} = K_t \frac{\partial \mathcal{N}_2}{\partial u}\frac{1}{S} \tag{3.17}$$

$$\frac{\partial^2 f_2}{\partial S^2} = K_t (\frac{\partial^2 \mathcal{N}_2}{\partial u^2}\frac{1}{S^2} - \frac{\partial \mathcal{N}_2}{\partial u}\frac{1}{S^2}) \tag{3.18}$$

$$\frac{\partial^2 f_2}{\partial S \partial V} = K_t \frac{\partial^2 \mathcal{N}_2}{\partial u \partial V}\frac{1}{S} \tag{3.19}$$

$$\frac{\partial^2 f_2}{\partial V^2} = K_t \frac{\partial^2 \mathcal{N}_2}{\partial V^2} \tag{3.20}$$

$$\frac{\partial f_2}{\partial V} = K_t \frac{\partial \mathcal{N}_2}{\partial V} \tag{3.21}$$

$$\frac{\partial f_2}{\partial t} = \frac{\partial K_t}{\partial t} + \frac{\partial \mathcal{N}_2}{\partial t} = rK_t\mathcal{N}_2 + K_t \frac{\partial \mathcal{N}_2}{\partial t} \tag{3.22}$$

Again, by applying results of (3.17)-(3.22) to (3.9), we get the following:

$$\frac{1}{2}VS^2\{K_t\frac{1}{S^2}(\frac{\partial^2 \mathcal{N}_2}{\partial u^2} - \frac{\partial \mathcal{N}_2}{\partial u})\} + \rho\sigma_v VS(\frac{1}{S}K_t\frac{\partial^2 \mathcal{N}_2}{\partial u \partial V}) + \frac{1}{2}\sigma_v^2 V K_t \frac{\partial^2 \mathcal{N}_2}{\partial V^2}$$
$$+rS(K_t\frac{\partial \mathcal{N}_2}{\partial u}\frac{1}{S}) + \{\kappa\theta - (\kappa+\beta)V\}K_t\frac{\partial \mathcal{N}_2}{\partial V} - rK_t\mathcal{N}_2 + (rK_t\mathcal{N}_2 + K_t\frac{\partial \mathcal{N}_2}{\partial t}) = 0$$

Since $K_t \neq 0$, let us omit it from the above equation and combine similar terms together. Then we will get the PDE for $\mathcal{N}_2$, which is:

$$\begin{aligned} \frac{1}{2}V\frac{\partial^2 \mathcal{N}_2}{\partial^2 u} + \rho\sigma_v V \frac{\partial^2 \mathcal{N}_2}{\partial u \partial V} + \frac{1}{2}\sigma_v^2 V \frac{\partial^2 \mathcal{N}_2}{\partial V^2} + (r - \frac{1}{2}V)\frac{\partial \mathcal{N}_2}{\partial u} \\ +\{\kappa\theta - (\kappa+\beta)V\}\frac{\partial \mathcal{N}_2}{\partial V} + \frac{\partial \mathcal{N}_2}{\partial t} = 0 \end{aligned} \tag{3.23}$$

We can write (3.16) and (3.23) together as follows:

$$\begin{aligned} \frac{1}{2}V\frac{\partial^2 \mathcal{N}_j}{\partial^2 u} + \rho\sigma_v V \frac{\partial^2 \mathcal{N}_j}{\partial u \partial V} + \frac{1}{2}\sigma_v^2 V \frac{\partial^2 \mathcal{N}_j}{\partial V^2} \\ +(r + e_j V)\frac{\partial \mathcal{N}_j}{\partial u} + \{\kappa\theta - \omega_j V\}\frac{\partial \mathcal{N}_j}{\partial V} + \frac{\partial \mathcal{N}_j}{\partial t} = 0 \end{aligned} \tag{3.24}$$

where $e_1 = \frac{1}{2}$, $e_2 = -\frac{1}{2}$, $\omega_1 = \kappa + \beta - \rho\sigma_v$, $\omega_2 = \kappa + \beta$, for $j = 1, 2$.
We know that the characteristic functions of Heston model takes the form:

$$\varphi_j(u, V, t, y) = exp[C_j(T-t, y) + D_j(T-t, y)V + iyu]$$

$\varphi_j(u, V, t, y)$ will also satisfy the PDE of $\mathcal{N}_j$, by applying (3.24) to $\varphi_j(u, V, t, y)$, we get:

$$\begin{aligned} \frac{1}{2}V\frac{\partial^2\varphi_j}{\partial u^2} + \rho\sigma_v V\frac{\partial^2\varphi_j}{\partial u\partial V} + \frac{1}{2}\sigma_v^2 V\frac{\partial^2\varphi_j}{\partial V^2} \\ +(r + e_j V)\frac{\partial\varphi_j}{\partial u} + \{\kappa\theta - \omega_j V\}\frac{\partial\varphi_j}{\partial V} + \frac{\partial\varphi_j}{\partial t} = 0 \end{aligned} \tag{3.25}$$

The corresponding derivatives of equation (3.25) are:

$$\begin{aligned} \frac{\varphi_j}{\partial u} &= iy\varphi_j \\ \frac{\partial^2\varphi_j}{\partial u^2} &= \varphi_j(iy)^2 = -y^2\varphi_j \\ \frac{\partial^2\varphi_j}{\partial u\partial V} &= iy\varphi_j D_j \\ \frac{\partial^2\varphi_j}{\partial V^2} &= \varphi_j D_j^2 \\ \frac{\partial\varphi_j}{\partial V} &= \varphi_j D_j \\ \frac{\partial\varphi_j}{\partial t} &= \varphi_j(\frac{\partial C_j}{\partial t} + \frac{\partial D_j}{\partial t}V) \\ &= \varphi_j(\frac{\partial C_j}{\partial \tau}\frac{\partial \tau}{\partial t} + \frac{\partial D_j}{\partial \tau}\frac{\partial \tau}{\partial t}) \\ &= -\varphi_j(\frac{\partial C_j}{\partial \tau} + \frac{\partial D_j}{\partial \tau}V) \end{aligned}$$

Here we let $\tau = T - t$, because it is more handy for calculation and in most literatures the explicit solution for the characteristic function is a function with respect to τ. Let us substitute the above partial derivatives to (3.25) to get the following result:

$$\begin{aligned} \varphi_j V(\frac{1}{2}y^2 + \rho\sigma_v V iy D_j + \frac{1}{2}\sigma_v^2 D_j^2 + iye_j - w_j D_J - \frac{\partial D_j}{\partial \tau}) \\ +\varphi_j(iyr + \kappa\theta D_j - \frac{\partial C_j}{\partial \tau}) = 0 \end{aligned} \tag{3.26}$$

Since $\varphi_j \neq 0$ and $V \neq 0$, we can rewrite (3.26) as :

$$\frac{\partial D_j}{\partial \tau} = \frac{1}{2}y^2 + \rho\sigma_v V iyD_j + \frac{1}{2}\sigma_v^2 D_j^2 + iye_j - w_j D_j \tag{3.27}$$

$$\frac{\partial C_j}{\partial \tau} = iyr + \kappa\theta D_j \tag{3.28}$$

where the $C_j(\tau, y)$ and $D_j(\tau, y)$ can be solved through an ordinary differential equation of Riccati-type. **The Riccati-equation is of the form:**

$$\phi^{'}(x) = a_0(x) + a_1(x)\phi(x) + a_2(x)\phi^2(x) \tag{3.29}$$

where $a_0(x) \neq 0$, and $a_2(x) \neq 0$
Let $z(x) = a_2(x)\phi(x)$, then

$$z^{'}(x) = a_2^{'}(x)\phi(x) + a_2(x)\phi^{'}(x) \tag{3.30}$$

In equation(3.30), substituting $\phi(x)$ with $\frac{z(x)}{a_2(x)}$, and $\phi^{'}(x)$ with equation(3.29), we get:

$$\begin{aligned}
z^{'}(x) &= a_2^{'}(x)\frac{z(x)}{a_2(x)} + a_2(x)[a_0(x) + a_1(x)\phi(x) + a_2(x)\phi^2(x)] \\
&= a_2^{'}(x)\frac{z(x)}{a_2(x)} + a_2(x)[a_0(x) + a_1(x)\frac{z(x)}{a_2(x)} + a_2(x)(\frac{z(x)}{a_2(x)})^2] \\
&= z(x)^2 + (a_1(x) + \frac{a_2^{'}(x)}{a_2(x)})z(x) + a_2(x)a_0(x)
\end{aligned}$$

Set $F(x) = a_1(x) + \frac{a_2^{'}(x)}{a_2(x)}$ and $G(x) = a_0(x)a_2(x)$, so

$$z^{'}(x) = z^2(x) + F(x)z(x) + G(x) \tag{3.31}$$

By setting $z(x) = -\frac{g^{'}(x)}{g(x)}$, then we obtain

$$\begin{aligned}
z^{'}(x) &= -[g^{''}(x)\frac{1}{g(x)} - g^{'}(x)\frac{g^{'}(x)}{g^2(x)}] \\
&= -\frac{g^{''}(x)}{g(x)} + (\frac{g^{'}(x)}{g(x)})^2 \\
\Rightarrow -\frac{g^{''}(x)}{g(x)} + (\frac{g^{'}(x)}{g(x)})^2 &= (-\frac{g^{'}(x)}{g(x)})^2 + F(x)(\frac{g^{'}(x)}{g(x)}) + G(x)
\end{aligned}$$

So,

$$g''(x) - F(x)g'(x) + G(x)g(x) = 0 \tag{3.32}$$

Supposing that $g(x) = e^{\alpha x}$ is a solution of (3.32), then we get:

$$\begin{aligned} \alpha^2 e^{\alpha x} - F(x)\alpha e^{\alpha x} + G(x)e^{\alpha x} = 0, \\ \Rightarrow [\alpha^2 - \alpha F(x) + G(x)]e^{\alpha x} = 0 \end{aligned} \tag{3.33}$$

So, finding the possible solutions of (3.32) is actually done by finding the roots of (3.33), which means if equation (3.33) has two roots α, β. Then $e^{\alpha x}$, $e^{\beta x}$ are solutions of (3.32). Furthermore we can easily check that the linear combination of $e^{\alpha x}$, $e^{\beta x}$ is also a solution of (3.32), so in the end, we can set $g(x) = Re^{\alpha x} + Se^{\beta x}$.

Let us go back to the definitions of $z(x) = -\frac{g'(x)}{g(x)}$ and $\phi(x) = \frac{z(x)}{a_2(x)}$, we have the following

$$\phi(x) = -\frac{1}{a_2(x)}\frac{g'(x)}{g(x)} = -\frac{1}{a_2(x)}\frac{R\alpha e^{\alpha x} + S\beta e^{\beta x}}{Re^{\alpha x} + Se^{\beta x}} \tag{3.34}$$

So now we can write equation (3.27) as :

$$\frac{\partial D_j}{\partial \tau} = (-\frac{1}{2}y^2 + iye_j) + (\rho\sigma_v Viy - w_j)D_j + \frac{1}{2}\sigma_v^2 D_j^2 \tag{3.35}$$

where $a_0(\tau) = -\frac{1}{2}y^2 + iye_j$, $a_1(\tau) = \rho\sigma_v Viy - w_j$ and $a_2 = \frac{1}{2}\sigma_v^2$

So we can have

$$\begin{aligned} F &= a_1(\tau) + \frac{a_2'(\tau)}{a_2(\tau)} = \rho\sigma_v iy - \omega_j \\ G &= a_0(\tau)a_2(\tau) = \frac{1}{2}\sigma_v^2(-\frac{1}{2}y^2 + iye_j) \end{aligned}$$

The roots of the equation $x^2 - xF(\tau) + G(\tau) = 0$ are : $\alpha = \frac{F+\sqrt{F^2-4G}}{2}$, $\beta = \frac{F-\sqrt{F^2-4G}}{2}$

Let

$$\begin{aligned} \xi &= \sqrt{F^2 - 4G} = \alpha - \beta \\ &= \sqrt{(\rho\sigma_v iy - \omega_j)^2 - \sigma_v^2(-y^2 + 2iye_j)} \end{aligned}$$

then

$$\alpha = \frac{\rho\sigma_v iy - \omega_j + \xi}{2}$$
$$\beta = \frac{\rho\sigma_v iy - \omega_j - \xi}{2}$$

Since equation (3.34) is the solution form of the Riccati-type ODE, apply directly to our case, we get that

$$D_j(\tau, y) = -\frac{1}{a_2(\tau)} \frac{R\alpha e^{\alpha\tau} + S\beta e^{\beta\tau}}{Re^{\alpha\tau} + Se^{\beta\tau}} \tag{3.36}$$

By $D_j(\tau, 0) = 0 \Longrightarrow \alpha R + S\beta = 0$, so $\alpha R = -S\beta$
Then $D_j(\tau, y)$ can be further simplified as :

$$\begin{aligned}
D_j(\tau, y) &= -\frac{1}{a_2(\tau)}[\frac{-S\beta e^{\alpha\tau} + S\beta e^{\beta\tau}}{-S\frac{\beta}{\alpha}e^{\alpha\tau} + Se^{\beta\tau}}] \\
&= -\frac{1}{a_2(\tau)}[\frac{S\beta e^{\beta\tau}(1 - e^{(\alpha-\beta)\tau})}{Se^{\beta\tau}(1 - \frac{\beta}{\alpha}e^{(\alpha-\beta)\tau})}] \\
&= \frac{-\beta}{a_2(\tau)}[\frac{1 - e^{\xi\tau}}{1 - ge^{\xi\tau}}] \\
&= \frac{\omega_j - \rho\sigma_v iy + \xi}{\sigma_v^2}[\frac{1 - e^{\xi\tau}}{1 - ge^{\xi\tau}}]
\end{aligned}$$

Here $g = \frac{\beta}{\alpha} = \frac{-\beta}{-\alpha} = \frac{\omega_j - \rho\sigma yi + \xi}{\omega_j - \rho\sigma yi - \xi}$
$C_j(\tau, y)$ can be solved by integrating $iyr + \kappa\theta D_j$.

Lemma 3.1: *If $f \in \mathcal{R}[a, b]$ and the function f is continuous at point $x \in [a, b]$, and function F is defined as $F(x) = \int_a^x f(t)\,\mathrm{d}t$ is differentiable at the point x, and the following equality holds:*

$$F^{'}(x) = f(x)$$

Applying Lemma 3.1 to $C_j(\tau, y)$:

$$\begin{aligned}
C_j(\tau, y) &= \int_0^\tau (ryi + \frac{\kappa\theta}{\sigma_v^2}(\omega_j - \rho\sigma yi + \xi)(\frac{1 - e^{\xi x}}{1 - ge^{\xi x}}))\,\mathrm{d}x \\
&= ryi\tau + \int_0^\tau \frac{\kappa\theta}{\sigma_v^2}(\omega_j - \rho\sigma yi + \xi)(\frac{(1+\frac{1}{g})e^{\xi x}}{\frac{1}{g} - e^{\xi x}})\,\mathrm{d}x
\end{aligned}$$

Since $\quad g = \frac{\omega_j - \rho\sigma yi + \xi}{\omega_j - \rho\sigma yi - \xi} \Longrightarrow 1 - \frac{1}{g} = \frac{2\xi}{\omega_j - \rho\sigma yi + \xi}$

So $\quad C_j(\tau, y) = ryi\tau + \frac{\kappa\theta}{\sigma_v^2}(\omega_j - \rho\sigma yi + \xi)\tau + \frac{\kappa\theta}{\sigma^2}\int_0^\tau \frac{2\xi e^{\xi x}}{\frac{1}{g} - e^{\xi x}}\,\mathrm{d}x$

Let $e^{\xi x} = h, \quad \xi e^{\xi x}\mathrm{d}x = \mathrm{d}h, \quad$ then $\xi h\mathrm{d}x = \mathrm{d}h$

$$\begin{aligned}
\int_0^\tau \frac{2\xi e^{\xi x}}{\frac{1}{g} - e^{\xi x}}\,\mathrm{d}x &= \int_1^{e^{\xi\tau}} \frac{2\xi h}{\frac{1}{g} - h}\frac{1}{\xi h}\,\mathrm{d}h \\
&= 2\int_1^{e^{\xi\tau}} \frac{1}{\frac{1}{g} - h}\,\mathrm{d}h \\
&= -2log(\frac{1}{g} - h)|_1^{e^{\xi\tau}} \\
&= -2[log(\frac{1}{g} - e^{\xi\tau}) - log(\frac{1}{g} - 1)] \\
&= -2log(\frac{1 - ge^{\xi\tau}}{1 - g})
\end{aligned}$$

So, $C_j(\tau, y) = ryi\tau + \frac{\kappa\theta}{\sigma^2}\{(\omega_j - \rho\sigma yi + \xi)\tau - 2log(\frac{1 - ge^{\xi\tau}}{1 - g})\}$

$$\varphi_j(u, V, t, y) = exp[C_j(T - t, y) + D_j(T - t, y)V + iyu]$$

This is equivalent to $\varphi_j(u, V, T - \tau, y) = exp[C_j(\tau, y) + D_j(\tau, y)V + iyu]$

So, we have:

$$
\begin{aligned}
C_j(\tau, y) &= ryi\tau + \frac{\kappa\theta}{\sigma^2}\{(\omega_j - \rho\sigma yi + \xi)\tau - 2log(\frac{1 - ge^{\xi\tau}}{1 - g})\} \\
D_j(\tau, y) &= \frac{\omega_j - \rho\sigma yi + \xi}{\sigma^2}(\frac{1 - e^{\xi\tau}}{1 - ge^{\xi\tau}}) \\
g &= \frac{\omega_j - \rho\sigma yi + \xi}{\omega_j - \rho\sigma yi - \xi} \\
\xi &= \sqrt{(\rho\sigma yi - \omega_j)^2 - \sigma^2(2e_j yi - y^2)}
\end{aligned}
$$

Theorem 3.1: *Let $f(t)$ be the characteristic function of a distribution function $F(x)$. Then for every x,*

$$
F(x) = \frac{1}{2} - \frac{1}{\pi}\int_0^\infty [\frac{Im(f(t)e^{-itx})}{t}]\,\mathrm{d}t
$$

Let $f(t)e^{-itx} = A + Bi$, then $\frac{f(t)e^{-itx}}{i} = -iA + B$, so $\frac{Im[f(t)e^{-itx}]}{t} = Re[\frac{f(t)e^{-itx}}{it}]$

From Theorem 3.1, we have:

$$
F(X \geq x) = 1 - F(X \leq x) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty Re[\frac{f(t)e^{-itx}}{it}]\,\mathrm{d}t \tag{3.37}
$$

Apply equation (3.37) directly to the characteristic function of Heston model, we get

$$
\begin{aligned}
\mathcal{N}_j(u, V, T, K) &= P[logS_T \geq log(K) | logS_t = u, V_t = V] \\
&= \frac{1}{2} + \frac{1}{\pi}\int_0^\infty Re(\frac{e^{-iylog(K)}\varphi_j(y)}{iy})\mathrm{d}y
\end{aligned}
$$

For simplicity I use $\varphi_j(y)$ to represent $\varphi_j(u, V, T - \tau, y)$. we can further calculate that $\varphi_1(y) = \frac{\varphi_2(y-i)}{\varphi_2(-i)}$. Here we do not write the steps of the calculation of $\varphi_1(y)$, because this is just simply to calculate C_2, D_2, g_2, ξ_2 of $\varphi_2(y - i)$ and $\varphi_2(-i)$, meanwhile calculating C_1, D_1, g_1, ξ_1 of $\varphi_1(y)$, and in the end get $\varphi_1(y) = \frac{\varphi_2(y-i)}{\varphi_2(-i)}$. $\mathcal{N}_2$ is the probability under risk-neutral measure, So $\varphi_2(y)$ is the corresponding characteristic function under risk neutral measure.

In Heston Model the risk premium is proportional to V and it can be written as $\lambda(S,V,t)=\beta V$, as we can see that the β doesn't appear in Heston's stock price process and volatility process, but it does exist in the pricing formula through $\mathcal{N}_j$. Since it is difficult to estimate the risk premium, we need to find a way such that the β can be effectively "eliminated". This can be solved by defining $\kappa^\star=\kappa+\theta$ and $\theta^\star=\frac{\kappa\theta}{\kappa+\theta}$. Then volatility process of Heston model becomes $\mathrm{d}V_t=\kappa^\star(\theta^\star-V_t)\mathrm{d}t+\sigma_v\sqrt{V_t}\mathrm{d}B_v(t)$, and this is the volatility process under risk neutral measure. So, under risk-neutral measure, the Heston model with dividend paid becomes:

$$\begin{aligned} dS_t &= (r-q)S_t dt+\sqrt{V_t}S_t dB_s(t) \\ dV_t &= \kappa^\star(\theta^\star-V_t)dt+\sigma_v\sqrt{V_t}dB_v(t) \end{aligned}$$

Risk neutral form of Bates Model
The Bates model is obtained from the Heston model by adding an independent jump, the price process under risk-neutral probability satisfies:

$$\begin{aligned} dS_t &= (r-q-\lambda\mu_J)S_t dt+\sqrt{V_t}S_t dB_s(t)+J_t S_t dN_t \\ dV_t &= \kappa^\star(\theta^\star-V_t)dt+\sigma_v\sqrt{V_t}dB_v(t) \end{aligned} \tag{3.38}$$

Lemma 3.2 (Ito's lemma for Poisson process) *Let x be a random variable and*

$$dx=a(.)dt+b(.)dW+g(.)dN$$

where W is a Brownian motion and N a Poisson process. Consider the function $F(t,x)$ and the differential of this function is

$$dF(x,t)=\{F_t+F_x a(.)+\frac{1}{2}F_{xx}b^2(.)\}dt+F_x b(.)dW+\{F(t,x+g(.))-F(t,x)\}dN$$

Applying lemma 3.2 (taking $F(x,t)=logS_t$), we get the risk neutral form for the $logS_t$ is:

$$\begin{aligned} d(logS_t) &= (\mu-q-\lambda\mu_J-\frac{V_t}{2})dt+\sqrt{V_t}dB_s(t)+log(1+J_t)dN_t \\ dV_t &= \kappa^\star(\theta^\star-V_t)dt+\sigma_v\sqrt{V_t}dB_v(t) \end{aligned} \tag{3.39}$$

In[Bate 1996][17], the characteristic function of $\mathcal{N}_2^{Bates}$ is :

$$\begin{aligned} \phi_2^{bates}(y) = exp[&C_2^{bates}(\tau,y)+D_2^{bates}(\tau,y)V+iyu \\ &+\lambda\tau((1+\mu_J)^{iy}e^{\frac{1}{2}\sigma_J^2((iy)^2-iy)}-1)] \end{aligned} \tag{3.40}$$

The $\frac{\partial C_j^{Bates}}{\partial \tau}$ equal to $iy(r-q-\lambda\mu_J)+\kappa\theta D_j$ instead of $\frac{\partial C_j}{\partial \tau}=iyr+\kappa\theta D_j$ in Heston model with no dividend. In the end the $C_j^{Bates}(\tau,y)$ has additional terms $-qiy\tau-\lambda\mu_J iy\tau$ added. So, the $C_2^{Bates}$of equation (3.40) can be written as:

$$C_2^{bates}(\tau,y)=iy(r-q-\lambda\mu_J)\tau+\frac{\kappa\theta}{\sigma^2}\{(\omega_j-\rho\sigma yi+\xi)\tau-2log(\frac{1-ge^{\xi\tau}}{1-g})\}$$

$D_2^{bates}(\tau,y)$, g, ξ are the same as those in Heston model. Looking back equation (3.4), (3.5), if we condition on $logS_0$, V_0, then

$$\begin{aligned}\phi_2^{bates}(y)=exp[C_2^{bates}(\tau,y)+D_2^{bates}(\tau,y)V+iyu\\+\lambda\tau((1+\mu_J)^{iy}e^{\frac{1}{2}\sigma_J^2((iy)^2-iy)}-1)]\end{aligned}\tag{3.41}$$

is changed to

$$\begin{aligned}\phi_2^{bates}(y)=exp[C_2^{bates}(\tau,y)+D_2^{bates}(\tau,y)V_0+iy(logS_0)\\+\lambda\tau((1+\mu_J)^{iy}e^{\frac{1}{2}\sigma_J^2((iy)^2-iy)}-1)]\end{aligned}\tag{3.42}$$

So, the explicit characteristic function of $\phi_2^{Bates}(\tau,y)$ can be represented as follows:

$$\begin{aligned}\phi_2^{Bates}(y)&=e^{A+B+C+D}\\A&=iys_0+iy(r-q)\tau\\B&=\frac{\theta^\star\kappa^\star}{\sigma_v^2}((\kappa^\star-\rho\sigma_v iy-d)\tau-2log(\frac{1-g_1e^{-d\tau}}{1-g_1}))\\C&=\frac{\frac{V_0}{\sigma_v^2}(\kappa^\star-\rho\sigma_v iy-d)(1-e^{-d\tau})}{1-g_1e^{-d\tau}}\\D&=-\lambda\mu_J iy\tau+\lambda\tau((1+\mu_J)^{iy}e^{\frac{1}{2}\sigma_J^2((iy)^2-iy)}-1)\\d&=\sqrt{(\rho\sigma_v iy-\kappa^\star)^2+\sigma_v^2(iy+y^2)}\\g_1&=\frac{\kappa^\star-\rho\sigma_v iy-d}{\kappa^\star-\rho\sigma_v iy+d}\end{aligned}$$

Here $g_1=\frac{1}{g}$, where the g is the one we get in the Heston model. For the reasons and more details, please see [18].

3.2 Fitting with Bates Model

Possible optimization methods are fminsearch, fminsearchbnd, lsqnonlin and fmincon. The first one does not solve problems with parameters which are bounded, while the second one does but it is quite time consuming. The lsqnonlin method needs the number of functions equal to the number of parameters. While fmincon minimizes the function with variables that are bounded and subject to linear equalities $Ax = b$, or inequalities $Ax \leq b$. A is a $n \times n$ matrix and b is a $n \times 1$ vector (n refers to the number of parameters). In the model we chose, there are 8 parameters to be optimized, which are: ν_0, $\theta^\star$, ρ, $\kappa^\star$, σ_v, λ, μ_J, vJ . Those parameters should be subject to the following constrains: $\nu_0 > 0$, $\theta^\star > 0$, $-1 \geq \rho \leq 1$, $\kappa^\star > 0$, $\sigma_v > 0$, $\lambda > 0$, $\mu_J > -20$, $0 < vJ < 1$. So, fmincon works the best.
Generally speaking, the μ_J can be any negative or positive number, but negative μ_J can motivate skew, so we set a negative value for the lower bound.

3.2.1 Calibration of Bates model

In this section, We will talk about calibrating the Bates model with market data. Our aim is to find parameters of the Bates model under which the model prices are very close to the market prices for several strikes. The way to find them is to minimize the sum of the squared errors between model and market prices. Once the parameters are got, we could calculate the option prices from Bates model and then by inversing the Black-Scholes formula, we will get the skew from the model. After that we compare it with the market skew and check whether they are very close, if not, we change the starting guess for the parameters which are needed to be estimated till the model skew fits the market skew well.

Fitting the market skew for single stocks

The index has European options, but the stocks inside the index have American options, so there is the early exercise problem. To avoid this problem, we try to calibrate the Bates model for all single stocks by minimizing the premium[12], which is:

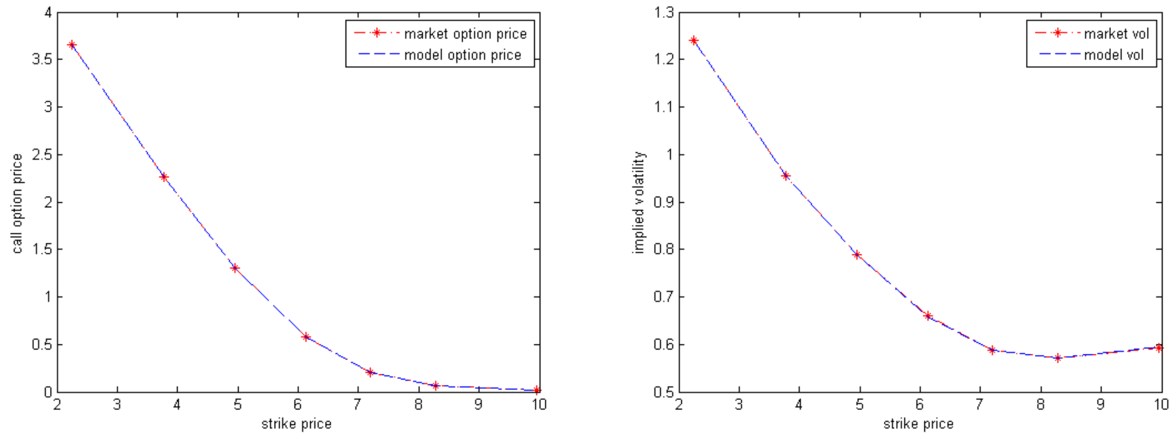
[12]Option price is formed by two parts: intrinsic value and premium. Intrinsic value is the payoff that the option is exercise immediately, so it is equal to $S_t - K$. While the

$$\sum_{K<S_0}(Put_{thoretic} - Put_{market})^2 + \sum_{K>S_0}(Call_{thoretic} - Call_{market})^2$$

we just took the fitting results of stock ING with startng date 10 October 2011 and maturity date 16 December 2011 as examples to show how well the model fits the market, because the other stocks and AEX are fitted as well as ING. we used trading days (there are 252 trading days a year), which is the remaining days of the actual days cutting off the holidays and weekends. To judge the good fitting of the model, we always look at the errors between the market and model values. There are two commonly used methods to measure the errors. One is the absolute error: $error = |Value_{market} - Value_{model}|$. The other is relative error: $error = \frac{|Value_{market} - Value_{model}|}{Value_{market}}$. Both measures are not precise. For example, if the market call option price is 0.005 euro, and the call option price from the model is 0.004 euro. The absolute difference is only 0.1 cents, but the relative error reaches to 20%, which means the relative error is quite high, but 0.1 cent is very small. On the other hand, if the market call price is 100 euro and the model call price is 101, then the relative error is low to 1%, while the absolute error is 1 euro, which means the absolute error is high, while the relative error is not high. However, the lower the absolute error, the smaller the relative error. So, here we judge the goodness of fit by looking at whether the absolute errors are less than 0.1 cents. If this error is lower than 0.1 cents, we say the fitting is good. We only take ING for example because all the other stocks and AEX fit as well as ING.

---

premium is the time value. Since the early exercise is for the options which are in the money case (Call option: $S_t > K$, Put option: $S_t < K$). For out of money options (Call option: $S_t < K$, Put option: $S_t > K$), the intrinsic values are zero, only premiums are paid. So, out of the money options are cheaper than the in the money options. In fact, the most traded options are out of the money options.



(a) ING Call price (b) ING skew

Figure 3.1: Fitted ING with the starting date 10 October 2011 and the maturity date 16 December 2011, it was calibrated by minimizing $\sum_{K<S_0}(Put_{thoretic} - Put_{market})^2 + \sum_{K>S_0}(Call_{thoretic} - Call_{market})^2$.

Strikes	9.948115	8.289616	7.204066	6.124477	4.948776	3.765623	2.247195
market call	0.016536	0.070116	0.207516	0.579118	1.301498	2.258174	3.658789
Bates call	0.016870	0.069436	0.208214	0.578581	1.301863	2.258034	3.658787
Price errors	0.000334	0.000679	0.000699	0.000536	0.000365	0.000140	0.000002
market VOL	0.593267	0.571844	0.586676	0.659186	0.787727	0.954172	1.239270
Bates VOL	0.595135	0.570435	0.587512	0.658667	0.788170	0.953866	1.239252
VOL errors	0.001868	0.001409	0.000837	0.000520	0.000443	0.000305	0.000017

Table 3.1: This table is to see how good the fit and how big the errors are for stock ING, as we can not observe the difference between the market and model values from the figure (3.1). Here **VOL** is short hand for **Volatility**. **Bates call** is the call option prices calculated by the Bates model. **Bates VOL** is the implied volatilities. The errors are calculate in absolute value (e.t $Priceerror = |maketcall - Batescall|$). As we can see from the table that the price errors are far less than 0.1 cents, which means the fitting is very good.

3.2.2 Fitting skew surface

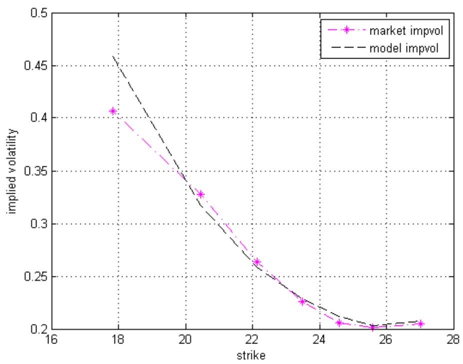
We selected several different maturity dates which are 11/18/2011, 12/16/2011, 3/16/2012, 6/15/2012, 9/21/2012, 12/21/2012 , and the starting maturity date was 10/10/2011. As mentioned before, in order to avoid early exercise problem, we tried to minimize the premium. For the dividend paid stock, firstly We adjusted the S_0 by $S_0^{new} = S_0^{old} - FDIV * e^{-r\tau}$ (FDIV is the cumulative dividend value at maturity time T and S_0^{old} is the original starting stock price). This is because the dividend cause the price to drop a certain amount (here the amount is FDIV) at maturity time, So under risk neutral measure the starting price should be $e^{-rT}(S_T - FDIV)$, which is equal to $S_0^{old} - FDIV * e^{-rT}$. Here we take UNA (Unilever) as an example (other stocks have similar results).

Firstly we show the single skew changes of UNA over different maturities under the same starting parameters. As we can see from figure (3.2), among six maturities, the fit on long maturities are better than the short ones. This is the same as we see from the 3-D plot (figure 3.3) in which the pictures (h, j) show us that there are big errors on short maturity for the option prices and skews.

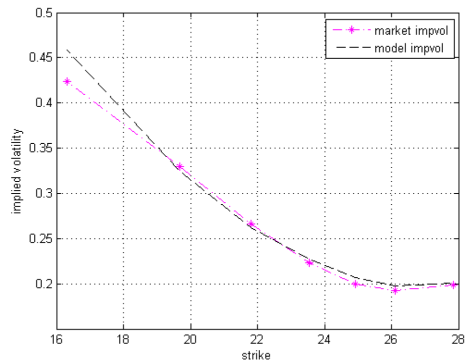
We assume that the reasons might come from two aspects: one is that parameters are very time dependent but this can not fully explain the big errors happened on short maturities, because it might be that parameters are always time dependent no matter the maturity is short or long. So, we think the other reason might be that the jump has different effect on short and long maturities. So we further checked these two assumptions and these will be shown in subsection (3.2.3) and (3.2.4).

Figure 3.2: UNA skews over different maturities

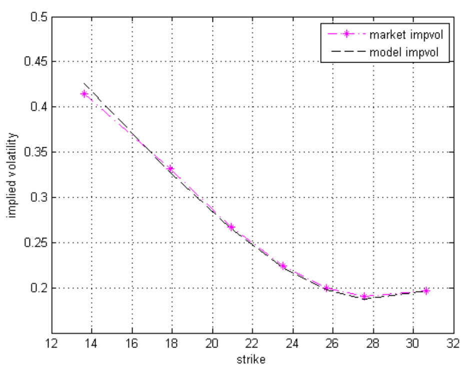
(a) 11/18/2011



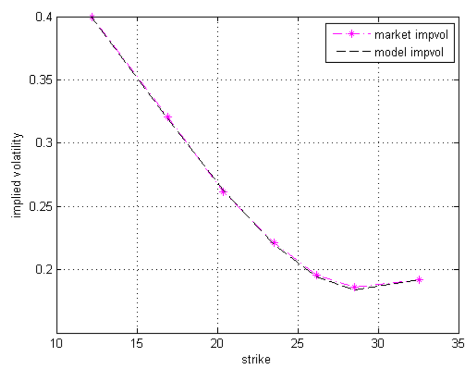
(b) 12/16/2011



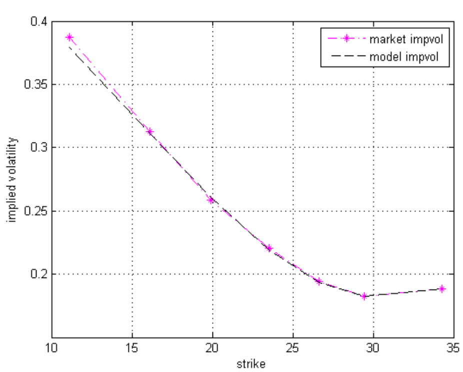
(c) 3/16/2012



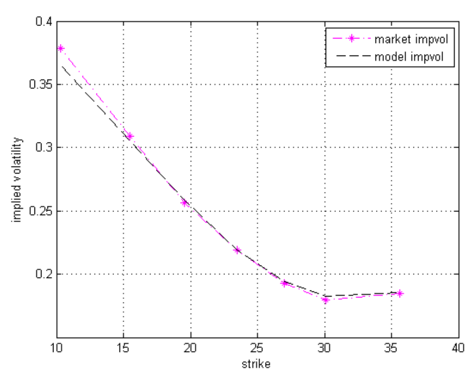
(d) 6/15/2012

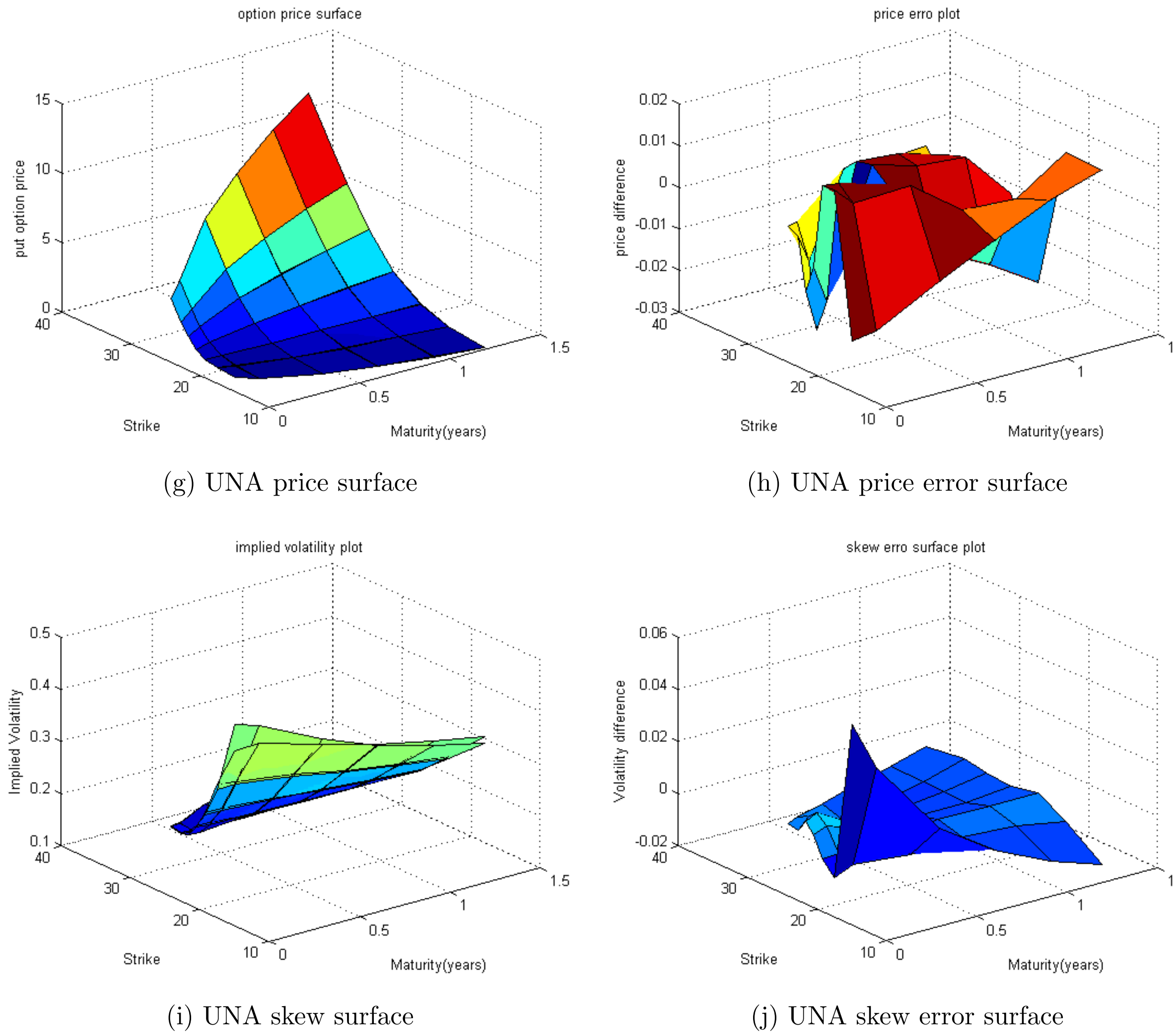


(e) 9/21/2012



(f) 12/21/2012





(g) UNA price surface

(h) UNA price error surface

(i) UNA skew surface

(j) UNA skew error surface

Figure 3.3: Stock UNA (Unilever NV), which is dividend paid over all maturities. The surfaces are obtained through minimizing $\sum_{i}^{T_n}\{\sum_{K<S_0}(Put_{thoretic} - Put_{market})^2 + \sum_{K>S_0}(Call_{thoretic} - Call_{market})^2\}$, here n refers to the number of maturities. T_n means the n^{th} maturity, K refers to the strike prices.

3.2.3 Checking varieties of parameters over maturities

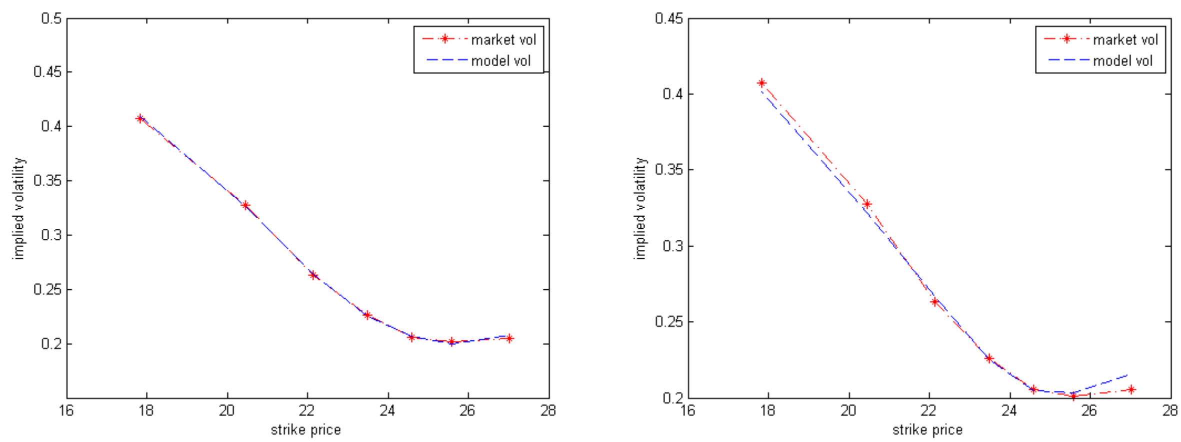
Taking UNA as an example, we tried to fit the stock well for every maturity and to see the stability of the parameters over different maturities.

Table 3.2: parameters of Stock UNA over different maturities

parameters	ν_0	$\theta^\star$	ρ	$\kappa^\star$	σ_v	λ	μ_J	vJ
11/18/2011	0.033897	0.054247	-0.369917	13.070810	1.067130	0.323576	-0.233765	0.003522
12/16/2011	0.044944	0.049430	-0.499288	10.387740	1.273956	0.224574	-0.200349	0.041803
3/16/2012	0.112442	0.019536	-0.523215	16.928530	0.881195	0.326202	-0.200805	0.066620
6/15/2012	0.811957	0.033317	-0.632419	42.423910	9.709552	0.368614	-0.048229	0.017690
9/21/2012	1.066548	0.033463	-0.644475	43.992830	12.318520	0.865870	-0.054851	0.007714
12/21/2012	0.938891	0.015292	-0.904913	42.284350	2.580545	0.124923	-0.277287	0.156852

As we can see from the table (3.2), the parameters are indeed time dependent, so it is reasonable that we could not fit skew surface well by sharing the same parameters for all maturities.

3.2.4 Checking jump effects on short and long maturities

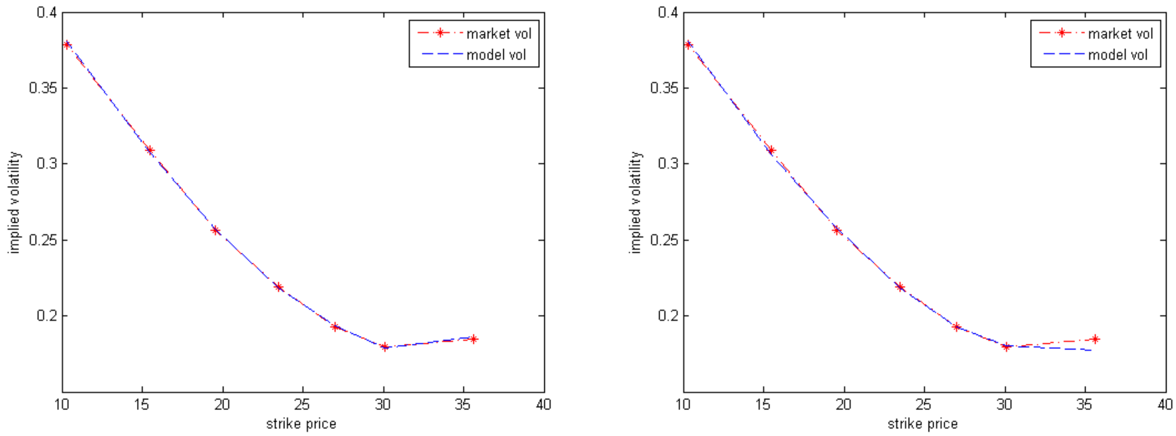


(a) UNA with jump (b) UNA without jump

Figure 3.4: Comparing the fits of stock UNA with jump and without jump on short maturity (Maturity: 11/18/2011). As we can see from the graph, the one with a jump fit much better than the one without a jump.

strikes	27.00697	25.58390	24.59000	23.48284	22.14174	20.44747	17.83381
vj errors	0.00222	0.00134	0.00087	0.00058	0.00059	0.00088	0.00277
vnj errors	0.01077	0.00194	0.00127	0.00065	0.00338	0.00627	0.00575

Table 3.3: This table shows the volatility errors in figure (3.5). **vj errors** means the volatility errors with a jump case, and **vnj errors** is the volatility errors without a jump case. We can see the "vj errors" are much smaller than the "vnj errors". Furthermore the error (without a jump) for the highest strike (27.006965) is more than one percent, which is high.



(a) UNA with jump

(b) UNA without jump

Figure 3.5: The fit for with a jump case and without a jump case on long maturity:(12/21/2012). As we can see from graph (b), the one without a jump already fits reasonably well although there are some tiny errors for the very high strikes.

strikes	35.61984	30.11833	27.00994	23.49625	19.54446	15.46154	10.29978
vj errors	0.00086	0.00071	0.00062	0.00052	0.00060	0.00086	0.00225
vnj errors	0.00778	0.00109	0.00006	0.00069	0.00151	0.00247	0.00221

Table 3.4: This table shows the volatility errors in figure (3.6). The meaning of **vj**, **vnj** are the same as those in table (3.3). Although for the long maturity, the one with a jump case still improves the fitting, we would say that the one without a jump already performs well, even for the highest maturity, the error is less than one percent. Because the deep in the money option and deep out of the money option are normally difficult to fit well.

Chapter 4

Correlation

4.1 Correlation assumptions

Our main purpose is to fit the skew of an index, and our assumption is that the skew of an index is influenced by two facts. One is the skew of each stock, the other is the correlations among stocks. We already showed that Bates model fit well the market skew in section 3.2.1. So for this part, we mainly show how correlations influence the skew.

For the index AEX, we have 25 stocks, so the correlation matrix is 25×25. Here we take the 3 dimensional case as an example. The correlation assumptions we made are the following three cases.

- Assumption 1: Constant correlation for each pair of stocks and all cross correlations are the same, but time dependent or independent.

$$\begin{pmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{pmatrix} \qquad or \qquad \begin{pmatrix} 1 & a & a \\ a & 1 & a \\ a & a & 1 \end{pmatrix}_t$$

- Assumption 2: Constant correlation for each pair stocks, the cross correlations are different and time independent.

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix}$$

- Assumption 3: Constant correlation for each pair stocks, the cross correlations are different and time dependent.

$$\begin{pmatrix} 1 & a & b \\ a & 1 & c \\ b & c & 1 \end{pmatrix}_t$$

4.2 Simulate AEX with constant correlation

We firstly try the correlation type 1, because it is easy to apply in any dimension, while correlation type 2 and type 3 are very difficult to guarantee the positive definiteness, specifically in the high dimensional cases. We use Monte Carlo simulations to simulate the AEX with the well fitted parameters of 25 stocks. All the stocks have starting date 10 October 2011 and maturity date 16 December 2011. we use equation(3.40) to simulate the AEX Under risk neutral measure, which is

$$\begin{aligned} d(logS_t) &= (\mu - q - \lambda\mu_J - \frac{V_t}{2})dt + \sqrt{V_t}dB_s(t) + log(1+J_t)dN_t \\ dV_t &= \kappa^\star(\theta^\star - V_t)dt + \sigma_v\sqrt{V_t}dB_v(t) \end{aligned} \tag{4.1}$$

Discritize the time as $0 = t_0 < t_1 \cdots < t_i < t_{i+1} \cdots < t_n = T$ and let $t_{i+1} - t_i = \Delta t$. (4.1) can be rewritten as:

$$log(S_{t+1}) = log(S_t) + (r - q - \lambda\mu_J - \frac{V_0}{2})\Delta t + \sqrt{V_t}\Delta B_s(t) + log(1+J_t)\Delta N_t \tag{4.2}$$

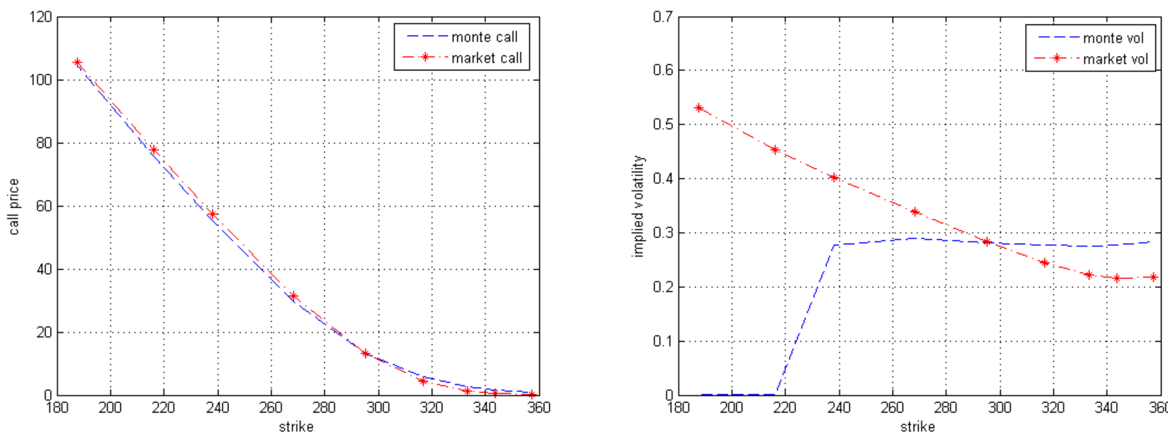
$$V_{t+1} = V_t + \kappa^\star(\theta^\star - V_t)\Delta t + \sigma_v\sqrt{V_t}\Delta B_v(t) \tag{4.3}$$

r, q, J_t, N_t, $\kappa^\star$, $\theta^\star$ are explained at pages 19 and 29. Since $B_s(t)$, $B_v(t)$ are correlated with ρ, and $\Delta B_s(t) = Z_t^{(1)}\sqrt{\Delta t}$, then $\Delta B_v(t) = (\rho Z_t^{(1)} + \sqrt{1-\rho^2}Z_t^{(2)})\sqrt{\Delta t}$, where $Z_t^{(1)}$ and $Z_t^{(2)}$ are i.i.d standard normal distributed. Substitute $dB_s(t)$ with $Z_t^{(1)}\sqrt{\Delta t}$, and $\Delta B_v(t) = (\rho Z_t^{(1)} + \sqrt{1-\rho^2}Z_t^{(2)})\sqrt{\Delta t}$ in equation (4.2), (4.3), we get

$$S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + log(1+J_t)\Delta N_t) \tag{4.4}$$

$$V_{t+1} = V_t + \kappa^\star(\theta^\star - V_t)\Delta t + \sigma_v\sqrt{V_t}(\rho Z_t^{(1)} + \sqrt{1-\rho^2}Z_t^{(2)})\sqrt{\Delta t} \tag{4.5}$$

We add correlation only for the stock price process and not for the volatility. Here we only consider the correlation to be between 0 and 1, and we exclude the extreme cases (0, 1), because the possibility is almost zero that all stocks are uncorrelated or all stocks always move exactly in the same direction with the same amount.
We tried different constant correlation values and we found that 0.8 works better than others, however, we did not get the same skew for both call and put options (see figure 4.1(b) and 4.2(b)).



(a) (b)

Figure 4.1: Simulating AEX **Call** option price and skew with $S_{t+1} = S_t * exp((r-q-\lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + log(1+J_t)\Delta N_t)$ under (500,000) Monte-Carlo simulations and constant correlation equal to 0.8.

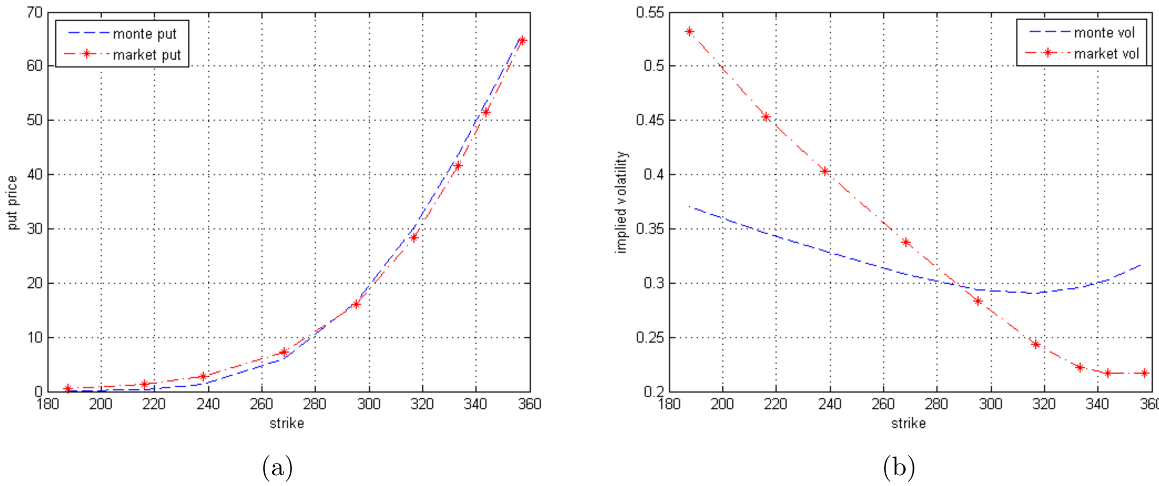


Figure 4.2: Simulating AEX **Put** option price and skew $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + log(1 + J_t)\Delta N_t)$ under (500,000) Monte-Carlo simulations with constant correlation equal to 0.8.

Since the skew for the call (figure 4.1(b)) behaves in a strange way and we would like to find what the reason is. As the jump sizes are small in our case, we tried to use J_t approximate $log(1 + J_t)$ and to see whether we still get the strange skew for the call option. From figure 4.3(b) and 4.4(b), we can see that call and put option have the same skew. So this approximation works well. Therefor, for all the further results, we use $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + J_t\Delta N_t)$ to simulate.

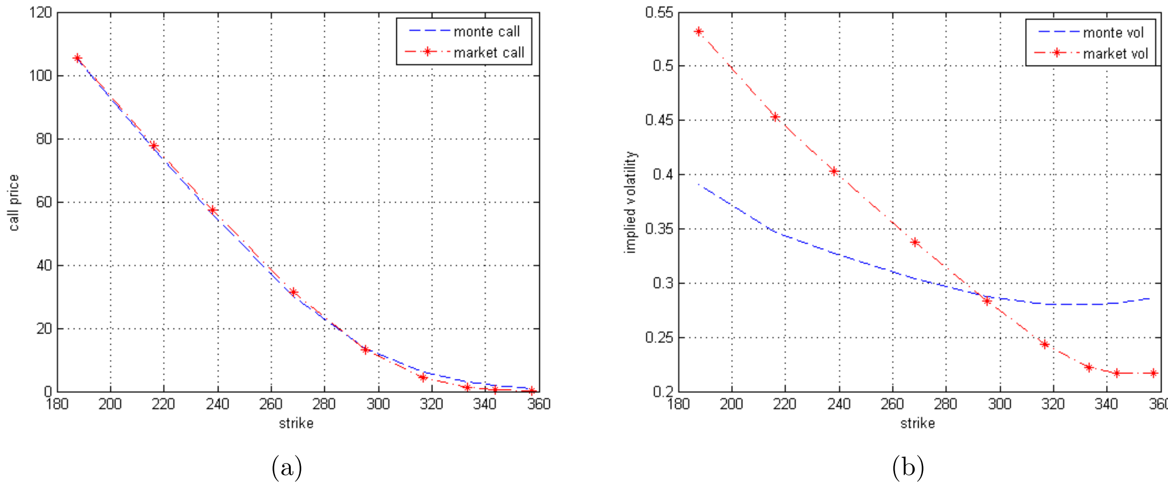


Figure 4.3: Simulating AEX **Call** option price and skew with $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + J_t\Delta N_t)$ under (500,000) Monte-Carlo simulations with constant correlation equal to 0.8.

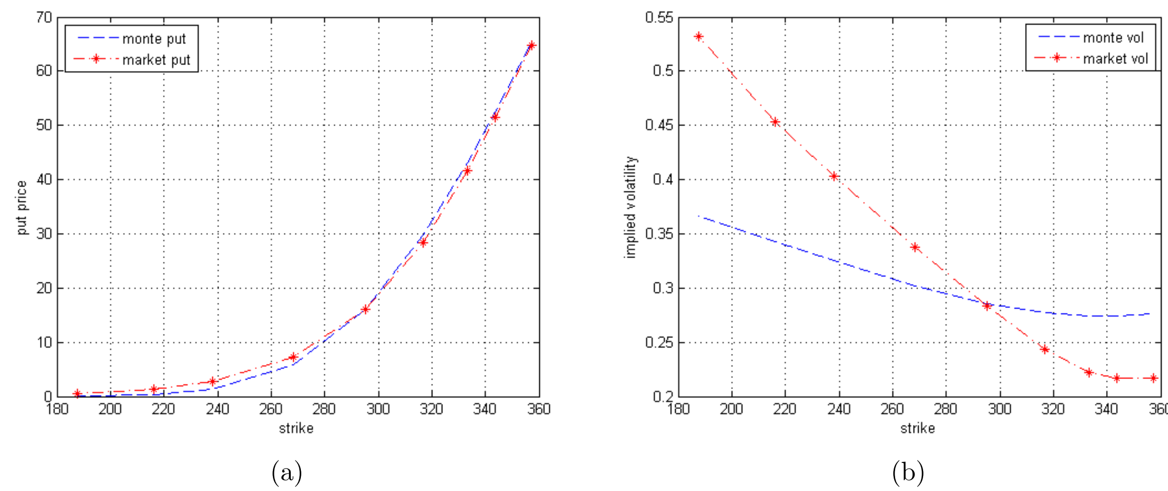


Figure 4.4: Simulating AEX **Put** option price and skew $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t}Z_t^{(1)}\sqrt{\Delta t} + J_t\Delta N_t)$ under (500,000) Monte-Carlo simulations with constant correlation equal to 0.8.

Conclusion: We can see from figure 4.3(a) and 4.4(a) that with these correlation assumptions, options with low strikes are under priced and options with high strikes are over priced. This is the same for the volatilities because the option price (both put and call) is an increasing function of the volatility if we keep all the other parameters unchanged, which is exactly what we see in figure 4.2(b) and 4.3(b). So, we can say that the constant correlation (the same for all strikes) does not work well, which means other types of correlation assumptions need to be made and the reasons for the steep skew need to be further researched. To check the possible facts that might cause a steep skew, the simplest and direct way is to compare the fitted parameters of AEX with its components. From the following table 4.1, we can see that the ρ of AEX is more negative than most of its components. This means the skew of the AEX is steeper than single stocks because the ρ reflects the negative correlation between underlying and volatility. The other thing that we can observe from the table is that the jump variance of the AEX is much higher than all its components, which means the jump also influences the steepness of the skew.

Remark: The parameters in table 4.1 are obtained through minimizing the premium, expressed in the following formula:

$$\sum_{K<S_0} (Put_{thoretic} - Put_{market})^2 + \sum_{K>S_0} (Call_{thoretic} - Call_{market})^2$$

All the stocks have starting date 10/Oct/2011 and maturity date 16/Dec/2011. Here we used trading days.

Table 4.1: Fitted parameters for stocks and AEX

parameters	ν_0	$\theta^\star$	ρ	$\kappa^\star$	σ_v	λ	μ_J	vJ
AEX	0.037143	0.169068	-0.774040	6.470988	1.536435	0.024248	-0.431970	0.657592
AF	0.420305	0.368542	-0.531282	5.077523	2.124942	0.215516	-0.216902	0.241377
AGN	0.394129	0.139633	-0.352032	5.013500	2.312570	0.283121	-0.427317	0.175129
AH	0.037884	0.077825	-0.424644	5.099371	1.142415	0.221142	-0.132072	0.065698
AKZA	0.111460	0.134471	-0.347604	8.650300	1.231725	0.673199	-0.137698	0.100476
APAM	0.403435	0.605266	-0.770629	6.522556	4.184424	8.453693	0.050553	0.015612
ASML	0.286284	0.024880	-0.422020	9.619432	1.509211	0.226022	-0.388642	0.080313
BOKA	0.126700	0.416918	-0.608509	0.087689	0.763535	0.323741	-0.206325	0.095764
CORA	0.069551	0.137373	-0.614287	5.974400	1.089494	0.256931	-0.168552	0.086568
DSM	0.106581	0.564561	-0.502543	0.032744	0.728442	0.401767	-0.251942	0.100612
FUG	0.183229	0.049034	-0.783477	5.091707	0.497412	0.565319	-0.214190	0.127666
HEIA	0.061699	0.074026	-0.635160	0.152139	0.468802	0.432953	-0.139957	0.060467
INGA	0.446804	0.583462	-0.576525	3.569300	2.973247	0.200673	-0.524258	0.596016
KPN	0.029849	0.125155	-0.274577	2.873190	0.717423	0.466189	-0.123414	0.056040
MT	0.155810	0.280366	-0.436487	5.190822	0.510341	1.372168	-0.205774	0.093452
PHIA	0.096358	0.281223	-0.549169	3.530070	1.092620	0.598306	-0.204061	0.106495
PNL	0.178516	0.392408	-0.210249	5.036041	2.090068	0.624568	-0.000520	0.118103
RDSA	0.040662	0.070376	-0.555301	8.081369	1.411878	0.107992	-0.279009	0.077570
REN	0.045209	0.058961	-0.375252	6.235844	0.590632	0.599503	-0.123382	0.052866
RND	0.157512	0.214291	-0.372447	7.522887	1.430216	0.527557	-0.245379	0.088632
SBMO	0.144642	0.207937	-0.468087	6.957124	1.352038	0.346433	-0.231887	0.177173
TNTE	0.185233	0.220348	-0.296115	7.782394	1.326597	0.611137	-0.151593	0.097327
TOM2	0.312679	0.386317	-0.336164	7.630839	2.951671	1.165344	0.004415	0.141301
ULA	0.074111	0.169877	-0.646710	1.516456	0.869413	0.208127	-0.207342	0.099191
UNI	0.032735	0.044746	-0.453156	3.293142	0.557817	0.501286	-0.159238	0.035318
WKL	0.076969	0.096262	-0.477337	2.837042	0.613707	0.395333	-0.176343	0.094739

4.3 Correlation models building

Correlation models 1, 2, 3:

$$\rho_t = \rho_{implied} + \beta(\frac{S_0 - S_t}{S_0})1_{\{S_t<S_0\}} - \alpha(\frac{S_t - S_0}{S_0})1_{\{S_t>S_0\}} \quad \textbf{(Model 1)}$$

ρ_t is the correlation for Brownian motion component and we assume that this ρ_t is stock price dependent. $\beta \in (0,1)$, $\alpha \in (0,1)$, α is higher than or as high as β (this is to allow leverage effect). $rho_{implied}$ is the correlation corresponds to the at money level, and it is the same as ρ in equation (4.6).

At time $t = 0$, we assume $correlation = \rho_{implied}$. However, we would like the model to reflect that negative return leads to higher correlation than positive return (property (1)) and the more that $S_t < S_0$, the higher the correlation (property (2)). The correlation becomes very low when S_t is much larger than S_0 (property (3)), the more the $S_t > S_0$, the lower the correlation (property (4)).

For an index, this holds: $S_{INDEX} = \sum_i^N (w_i S_i)$, and we have that the volatility should be equal to $\sigma_I^2 = \sum_{i=1}^N \omega_i^2 \sigma_i^2 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^N \omega_i\omega_j\sigma_i\sigma_j\rho_{ij}$
Since we assume that ρ_{ij} is the same for each pair of stocks, so we have :

$$\sigma_I^2 = \sum_{i=1}^N \omega_i^2 \sigma_i^2 + 2\sum_{i=1}^{N-1}\sum_{j=i+1}^N \omega_i\omega_j\sigma_i\sigma_j\rho$$

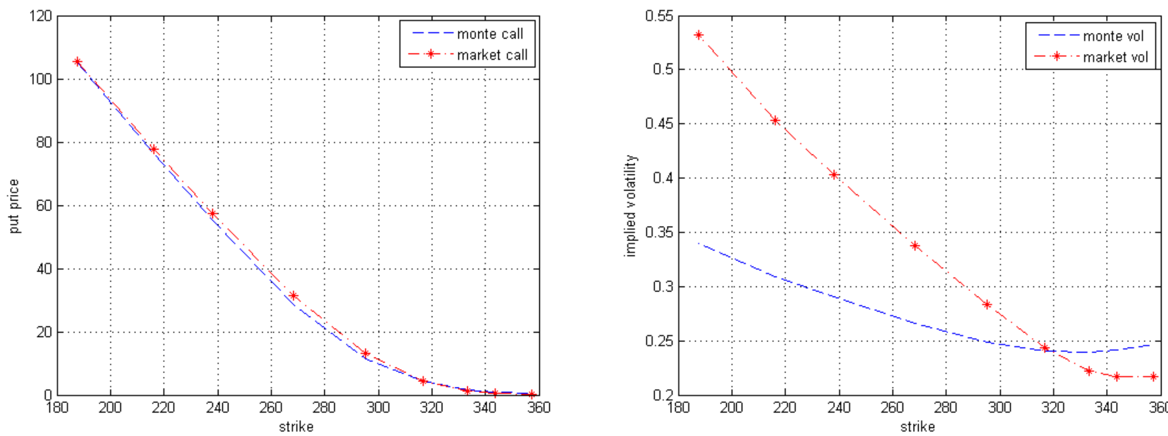
so, we can write ρ as:

$$\rho = \frac{\sigma_I^2 - \sum_{i=1}^N \omega_i^2\sigma_i^2}{(\sum_{i=1}^N \omega_i\sigma_i)^2 - \sum_{i=1}^N \omega_i^2\sigma_i^2} \tag{4.6}$$

We calculate the implied correlation with the above formula and σ_i is substitute with the $\sigma_i^{implied}$. We set $\beta = 0.4$, $\alpha = 0.6$, Starting with $correlation = \rho_{implied}$, and simulate AEX with fitted 25 stocks. At time step t_i we get $S_{t_i}^{AEX}$ and we compare it with S_0^{AEX}.
$if\ S_{t_i} < S_0,\ \rho_{t_i} = \rho_{implied} + \beta(\frac{S_0 - S_{t_i}}{S_0})$
$if\ S_{t_i} > S_0,\ \rho_{t_i} = \rho_{implied} - \alpha(\frac{S_0 - S_{t_i}}{S_0})$ if $\rho_{t_i} > 1$, we let $\rho_{t_i} = 1$. If $\rho_{t_i} < 0$, we let $\rho_{t_i} = 0$, because ρ must be in [0 1] interval to guarantee the positive definiteness of the correlation matrix, reasons about this can be found in [5].

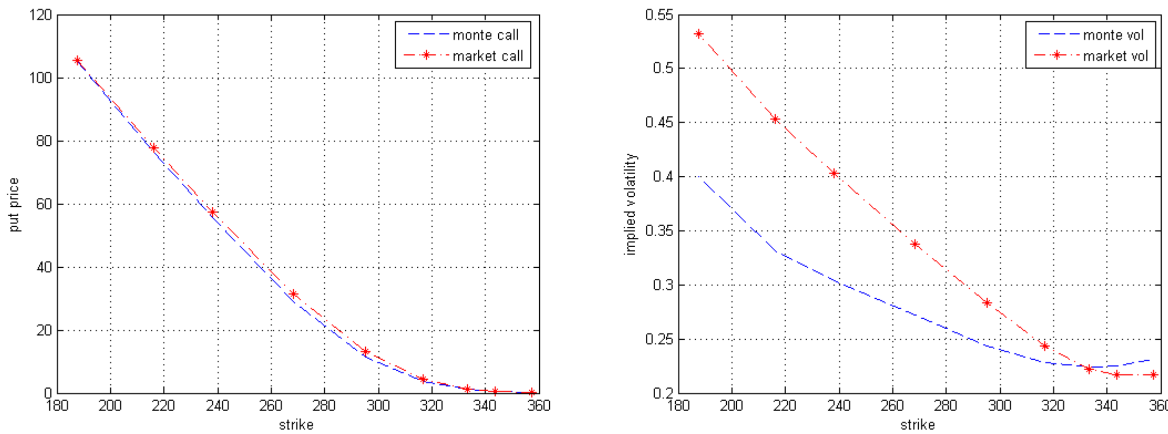
Under this model, the skew is improved a bit, but the steepness is not enough, please see figure(4.5(b)). This is probably because the $|S_0 - S_{t_i}|$ is very small compared to S_0, so the correlation does not change much, instead we change $\frac{S_0 - S_{t_i}}{S_0}$ as $\sqrt{\frac{S_0 - S_{t_i}}{S_0}}$ and the **model 2** becomes $\rho_t = \rho_{implied} + \beta(\sqrt{\frac{S_0 - S_t}{S_0}})1_{\{S_t<S_0\}} - \alpha(\sqrt{\frac{S_t - S_0}{S_0}})1_{\{S_t>S_0\}}$. This change makes the skew steeper than the model 1 (figure 4.6(b)), but still not enough. So, in the end, we tried different powers and we found that power (1/4) further improves a bit the results (figure 4.7(b)), and we call it **Model 3**, which is shown as follows.

$$\rho_t = \rho_{implied} + \beta(\frac{S_0 - S_t}{S_0})^{\frac{1}{4}}1_{\{S_t<S_0\}} - \alpha(\frac{S_t - S_0}{S_0})^{\frac{1}{4}}1_{\{S_t>S_0\}}$$



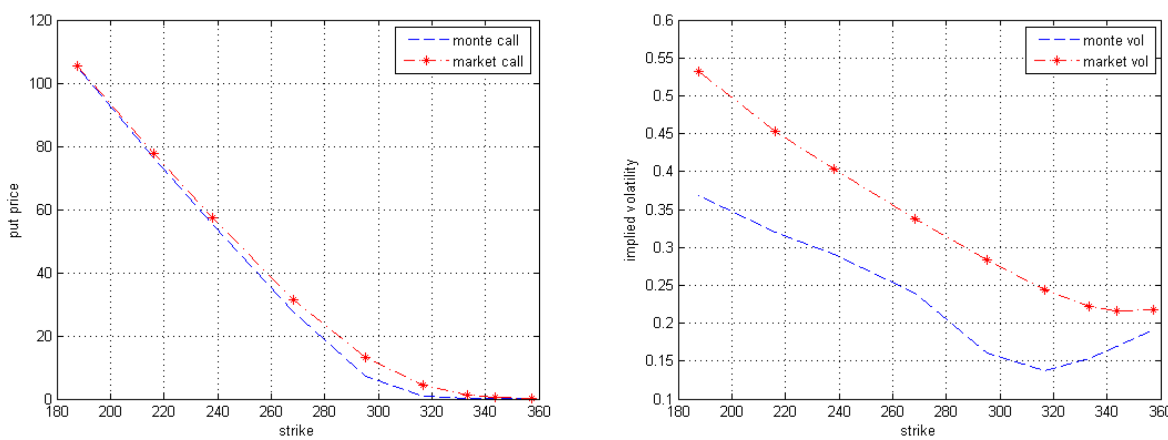
(a) (b)

Figure 4.5: Call option price and skew plot of model 1:
$\rho_t = \rho_{implied} + \beta(\frac{S_0 - S_t}{S_0})1_{\{S_t<S_0\}} - \alpha(\frac{S_t - S_0}{S_0})1_{\{S_t>S_0\}}$
under 500,000 simulations.



(a) (b)

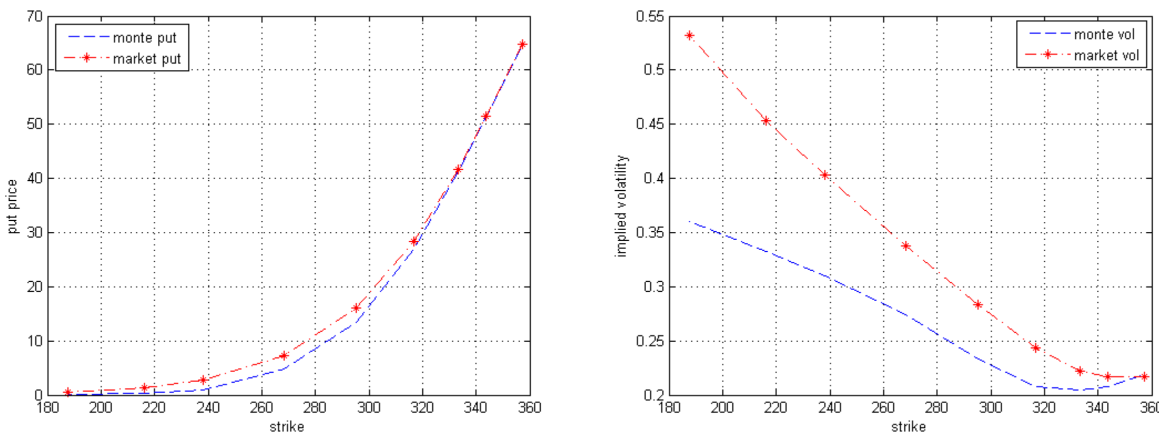
Figure 4.6: Call option price and skew plot of model 2:
$\rho_t = \rho_{implied} + \beta(\sqrt{\frac{S_0 - S_t}{S_0}})1_{\{S_t < S_0\}} - \alpha(\sqrt{\frac{S_t - S_0}{S_0}})1_{\{S_t > S_0\}}$
under 500,000 simulations.



(a) (b)

Figure 4.7: Call option price and skew plot of model 3:
$\rho_t = \rho_{implied} + \beta(\frac{S_0 - S_t}{S_0})^{\frac{1}{4}}1_{\{S_t < S_0\}} - \alpha(\frac{S_t - S_0}{S_0})^{\frac{1}{4}}1_{\{S_t > S_0\}}$
under 500,000 simulations.

It is a bit strange that there is a local minimum in figure 4.7(b), and we think it might be the drawbacks of the Bates model, which has different effects on call and put. But it can also be the discontinuity of the correlation model. Firstly we tried to see the put option result simulated with model 3. In figure 4.8(b) we can see that the skew curve appears nicer. So the problem may come from the defect of the Bates model. However, all the above curves are not so smooth, so we further tried to build a continuous exponential model (model 4).



(a) (b)

Figure 4.8: Put option price and skew plot of model 3:
$\rho_t = \rho_{implied} + \beta(\frac{S_0-S_t}{S_0})^{\frac{1}{4}}1_{\{S_t<S_0\}} - \alpha(\frac{S_t-S_0}{S_0})^{\frac{1}{4}}1_{\{S_t>S_0\}}$
under 500,000 simulations.

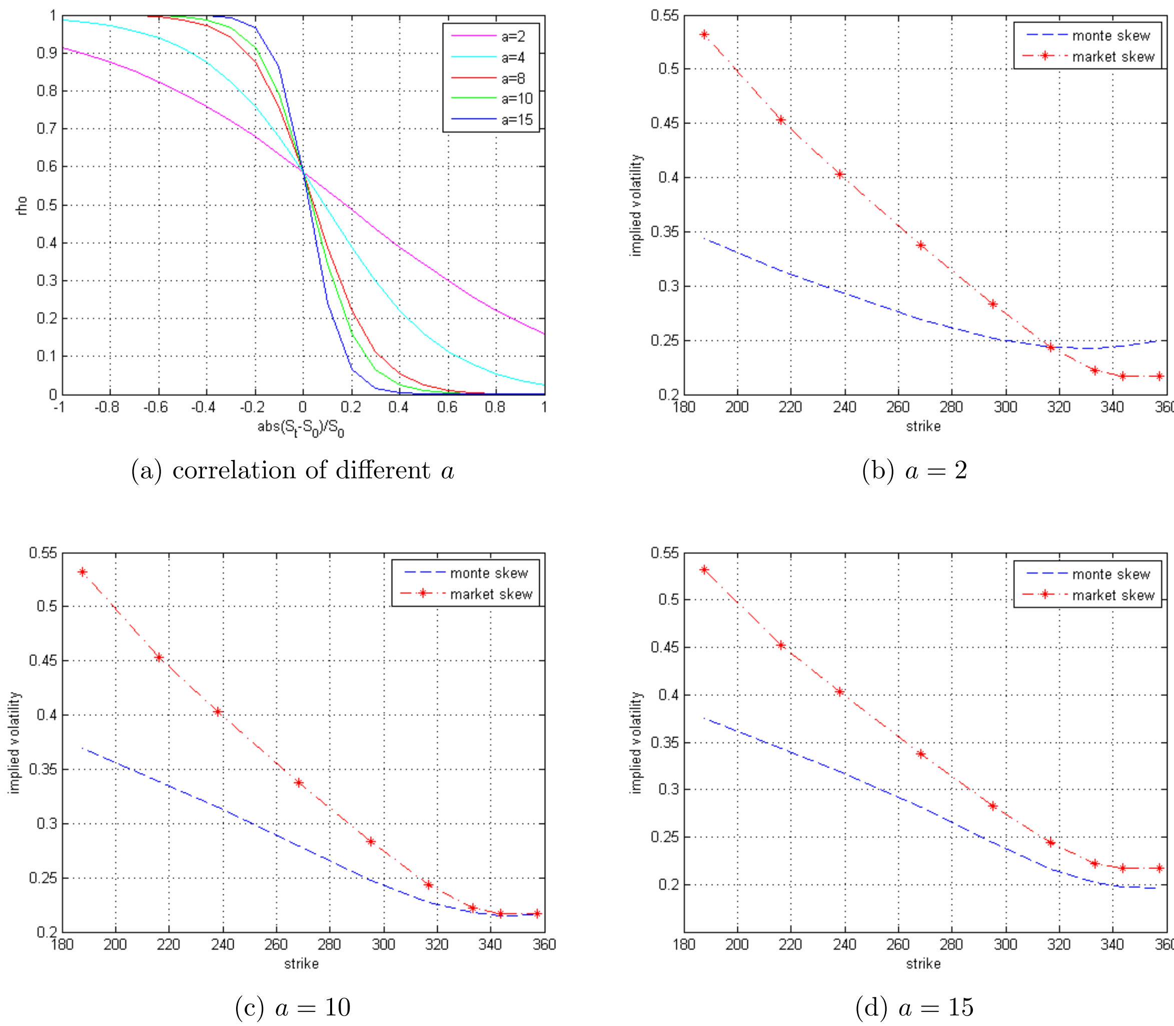
Correlation model 4

Set $x = (S_t - S_0)/S_0$

$\rho(x) = \frac{1}{1+exp(ax)(\frac{1}{\rho_{implied}}-1)}$

In this model, We use ρ to represent the correlation for the Brownian motion component and we did not add correlation to the jump part. The a in this model controls the steepness of the correlation. As we can see from figure 4.9 (a) the higher a, the steeper the slope of the correlation around $S_t = S_0$.

In fact, the simulated skew of the AEX gets steeper as a increases, this can be see seen from figure 4.9 (b) (c), (d).



(a) correlation of different a

(b) $a = 2$

(c) $a = 10$

(d) $a = 15$

Figure 4.9: Compare to previous skew results, this exponential model performs better, and specifically the one with $a = 15$, which has the steepest skew. This can be seen that the lowest point of the skew reaches to 0.2 or slightly below it, and the highest point of this skew is almost the same as the other skews.

Correlation model 5: stochastic correlation model

We assume the price and volatility processes for the stocks still follow Bates model, while the correlation model that we built for the Brownian motion components are:

$$d\rho_{ij,t} = \alpha(\rho_{implied} - \rho_{ij,t})dt + \sigma_\rho(-\eta_i dW_i - \eta_j dW_j) + (\beta - \rho_{ij,t})dN_t^\rho \quad (4.7)$$

$$dW_i = \rho_{ij}dW_j + \sqrt{1-\rho_{ij}^2}dW \quad (4.8)$$

In the above equations, the indice i refers to the i^{th} stock and $i \neq j$. W is the common Brownian motion which is independent from W_i (the Brownian motion related to stock i). η_i, η_j are positive numbers, thus ρ_{ij} is negative correlated with W_i, W_j. N_t^ρ is a Poisson counting process referring to the correlation process, and we use λ_ρ denote its jump rate. Taking 3 stocks as an example, we have

$$dW_2 = \rho_{21}dW_1 + \sqrt{1-\rho_{21}^2}dW$$
$$dW_3 = \rho_{31}dW_1 + \sqrt{1-\rho_{31}^2}dW$$

Let Z_2, Z_3, and Z be random variables from standard normal distribution. By the property 2 of the Brownian motion, we can writing the above equations as discrete cases, which are:

$$\sqrt{\Delta t}Z_2 = \rho_{21}Z_1\sqrt{\Delta t} + \sqrt{1-\rho_{21}^2}Z\sqrt{\Delta t}$$
$$\sqrt{\Delta t}Z_3 = \rho_{31}Z_1\sqrt{\Delta t} + \sqrt{1-\rho_{31}^2}Z\sqrt{\Delta t}$$

omitting the $\sqrt{\Delta t}$ from the above equations we get:

$$Z_2 = \rho_{21}Z_1 + \sqrt{1-\rho_{21}^2}Z$$
$$Z_3 = \rho_{31}Z_1 + \sqrt{1-\rho_{31}^2}Z$$

By $cov(aX, bY) = abcov(X, Y)$, and $var(Z_2) = 1$, $var(Z_3) = 1$

We get $\rho_{32} = \rho_{21}\rho_{31} + \sqrt{1-\rho_{21}^2}\sqrt{1-\rho_{31}^2}$ So, the correlation matrix is :

$$\begin{pmatrix} 1 & \rho_{21} & \rho_{31} \\ \rho_{21} & 1 & \rho_{21}\rho_{31} + \sqrt{1-\rho_{21}^2}\sqrt{1-\rho_{31}^2} \\ \rho_{31} & \rho_{21}\rho_{31} + \sqrt{1-\rho_{21}^2}\sqrt{1-\rho_{31}^2} & 1 \end{pmatrix}_t$$

And every time, we can updated ρ_{12}, ρ_{13} by the following equations:

$$d\rho_{12,t} = \alpha(\rho_{implied} - \rho_{12,t})dt + \sigma_\rho(-\eta_1 dW_1 - \eta_2 dW_2) + (\beta - \rho_{12,t})dN_t^\rho$$
$$d\rho_{13,t} = \alpha(\rho_{implied} - \rho_{13,t})dt + \sigma_\rho(-\eta_1 dW_1 - \eta_3 dW_3) + (\beta - \rho_{13,t})dN_t^\rho$$

Once we updated $\rho_{12,t}$ and $\rho_{13,t}$, we can get the corresponding $\rho_{32,t}$.

However, this correlation model can only be applied to low dimension. The stocks we chose were ING, RDSA, and PHI. So, the "weights" (number of shares) should be adjusted. This was already showed in Chapter 2.

We tried three different groups of parameters and they are shown in the following table. For this model, we only did 50000 Monte Carlo simulations because it takes days to simulate 500,000. in our case most jumps were negative (see table 4.1 μ_J), and according to the leverage effect, the correlation will increase. So, in the following table, the β was set very high for this reason. In this way, $\beta - \rho_{ij,t}$ will be positive when a jump happens, thus the correlation will increase.

parameters	skew1	skew2	skew3
α	0.1	0.1	0.1
β	0.9	0.9	0.9
$\rho_{implied}$	0.6	0.6	0.6
σ_ρ	0.3	0.3	0.3
η_1	0.1	0.1	0.5
η_2	0.3	0.3	0.6
λ_ρ	0.9	0.9	0.9
ρ_1	0.1	0.5	0.1
ρ_2	0.2	0.6	0.2

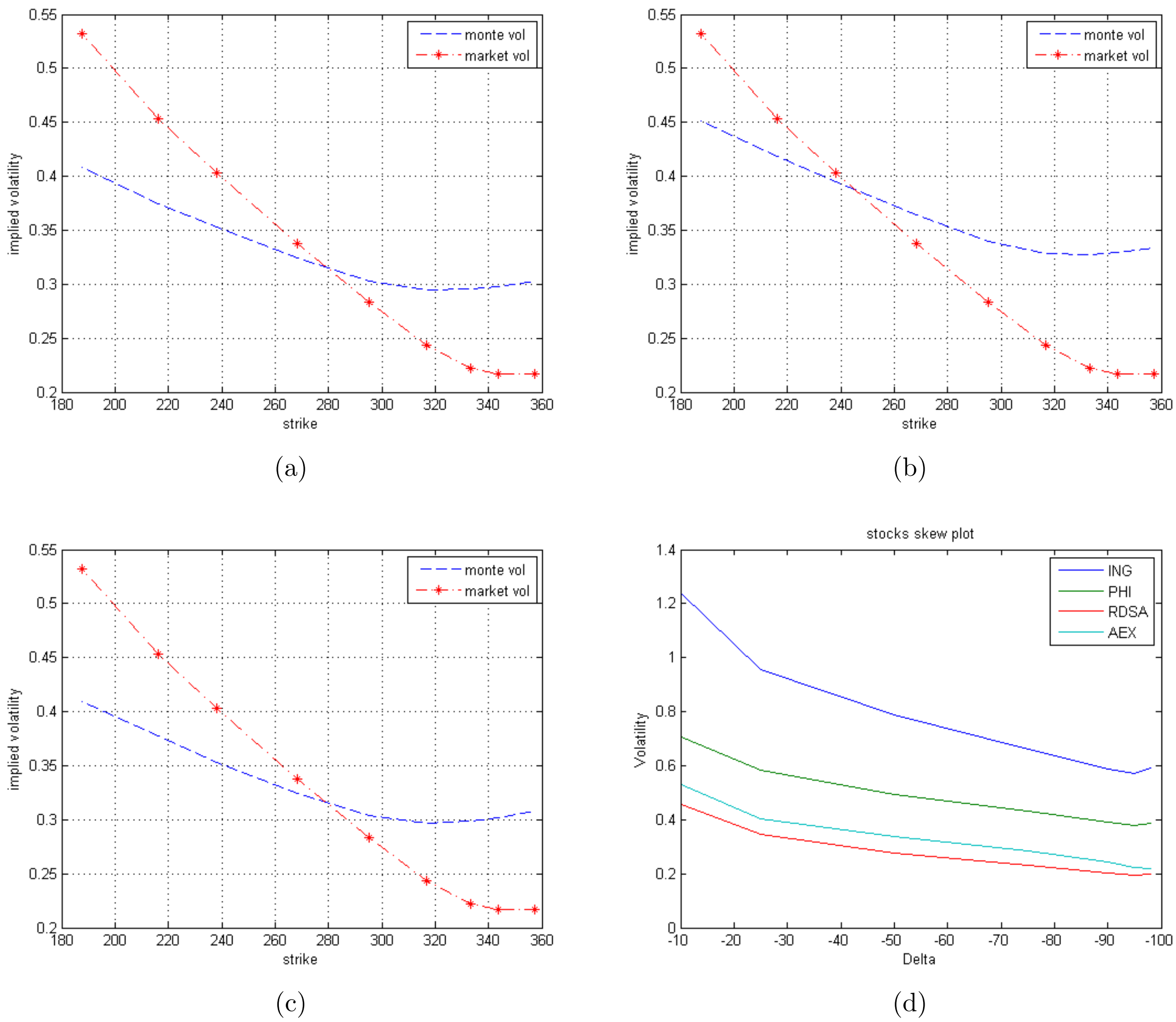


Figure 4.10: Skews plots of model 5: $d\rho_{ij,t} = \alpha(\rho_{implied} - \rho_{ij,t})dt + \sigma_\rho(-\eta_i dW_i - \eta_j dW_j) + (\beta - \rho_{ij,t})dN_t^\rho$. Picture (d) is the maket skew plots of the chosen three stocks and AEX.

From figure 4.10 (a), (b), (c), we can see that the skews are not steep but the highest point for the low strike is higher than the skews from previous models. This is because previous models are based on 25 stocks, while here we only chose three stocks and two of them had high volatilities and steep skews (please see figure 4.10 (d)), this might be the reason. So, randomly choosing three stocks are not so realiable. From previous model results, we know that we really need high correlations for low strikes and low correlations for high strikes to get steeper skews. Although we allowed leverage for single stocks in model 5, it does not mean it will generate suitable correlations for low and high strikes in the process of the synthetic AEX. So, this model failed.

Add correlation to the jump

From the Bates model, we have the jump parameter J_t, and J_t is distributed as Gaussian:

$$log(1 + J_t) \sim \mathcal{N}(log(1 + \mu_J) - \frac{(\sigma_J^2)}{2}, \sigma_J^2)$$

Suppose we have two stocks, then we can write :

$$log(1 + J_t^{(1)}) = x_1$$

$$log(1 + J_t^{(2)}) = x_2$$

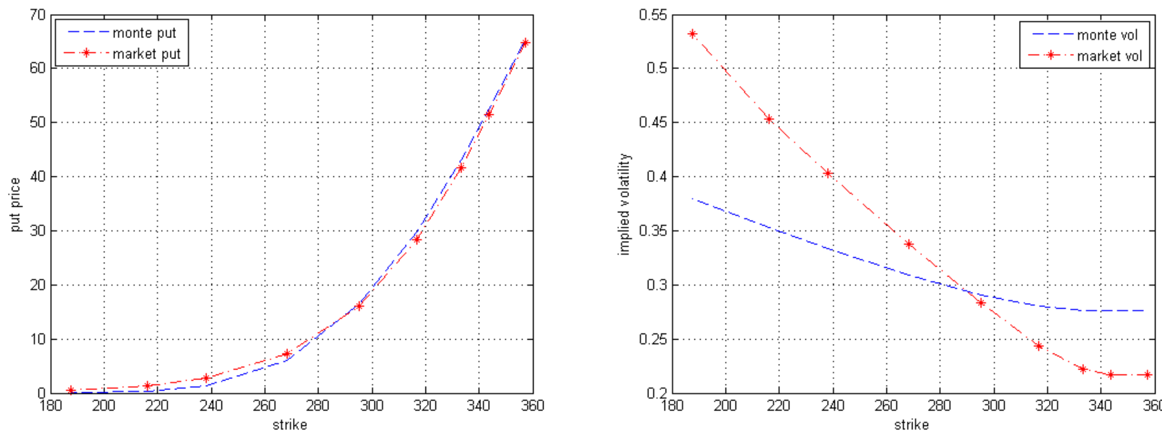
If $J_t^{(1)}, J_t^{(2)}$ are small (near 0), we could approximate $log(1 + J_t^{(1)})$ by $J_t^{(1)}$, and $log(1 + J_t^{(2)})$ by $J_t^{(2)}$. So, the correlation of $J_t^{(1)}, J_t^{(2)}$ is the same as the correlation of $log(1+J_t^{(1)}), log(1+J_t^{(2)})$. Then, the way of adding correlation to the jump is similar to the one as adding correlation to the diffusion part. What we do in the simulation is firstly we assume a value for the constant correlation (here we use $\rho_J$to present this value) and again with all cross correlations between different stock pairs are equal. Then we can generate a correlation matrix with diagonal elements equaling to 1 and other entries equaling to ρ_J. After that we use Chelosky decomposition to decompose this matrix to a lower triangular matrix (L), then generating a n dimension vector $\epsilon = (\epsilon_1, \epsilon_2, \epsilon_3, \cdots, \epsilon_n)'$ from standard normal distribution and letting $Z = L * \epsilon$, so Z was a vector with n elements (n is the number of stocks), and each pair of elements had the correlation ρ_J. In the end, we let $log(1+J_t^{(i)}) = x_i = ln(1 + \mu_J) - vJ/2 + vJ. * z_i$, $((z_i)'s$ were the elements of vector Z) and we got $J_t^{(i)} = exp(x_i) - 1$. In this way, we added the correlation to the jump process.

However, this approximation way may lead to big error if μJ and vJ are very big. In our case the $\mu J's$ and $vJ's$ are small , so that $log(1 + J_t)$ is close to J_t for almost all stocks, which can be seen from the following table:

stocks	log(1+J)	J
AFA	-0.0858	-0.0822
AGN	-0.5742	-0.4368
AH	-0.2139	-0.1926
AKZA	-0.3007	-0.2597
APAM	0.0517	0.0530
ASML	-0.5044	-0.3961
BOKA	-0.2008	-0.1820
CORA	-0.1202	-0.1133
DSM	-0.1790	-0.1639
FUG	-0.0811	-0.0779
HEIA	-0.2453	-0.2175
INGA	-1.8405	-0.8413

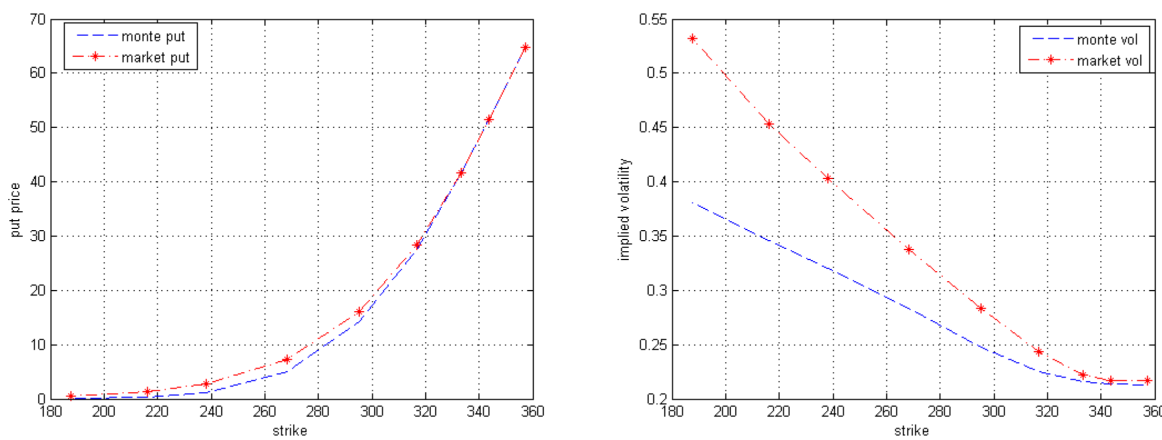
stocks	log(1+J)	J
KPN	-0.1060	-0.1005
MT	-0.0409	-0.0400
PHIA	-0.2891	-0.2511
PNL	-0.0735	-0.0709
RND	-0.4023	-0.3312
RDSA	-0.2766	-0.2416
REN	-0.1409	-0.1315
SBMO	-0.4715	-0.3759
TNTE	-0.4054	-0.3333
TOM2	-0.1139	-0.1077
ULA	-0.0284	-0.0280
UNI	-0.1892	-0.1724
WKL	-0.1735	-0.1593

We assumed that the higher correlation for the jump, the steeper skew, because high correlation for the jump lead to stocks jump almost the same time. Firstly we started with very simple case that we assumed the correlations for all strikes are the same for both jump part and Brownian motion part. From figure 4.11(b) we can see that this simple assumption seems not to improve the skew compared to 4.4 b (original skew). Then we kept constant correlation (time independent) for the jump part and combined the correlation model 2 which was applied to the Brownian motion part and from figure 4.12(b) we see that the skew was improved.



(a) (b)

Figure 4.11: $\rho_W = 0.8$, and $\rho_J = 0.9$. ρ_W refers to the correlation of the Brownian motion while ρ_J refers to the correlation of the jump component. The results are from 500,000 Monte-Carlo simulations.



(a) (b)

Figure 4.12: $\rho_J = 0.9$, ρ_W follows model 2 $\rho_t = \rho_{implied} + \beta(\sqrt{\frac{S_0 - S_t}{S_0}})1_{\{S_t < S_0\}} - \alpha(\sqrt{\frac{S_t - S_0}{S_0}})1_{\{S_t > S_0\}}$ The results are from 500,000 Monte-Carlo simulations.

4.4 Multivariate GARCH model

As we mentioned in Model 5 (the stochastic correlation model), the most difficult problem in modeling the correlation is to guarantee the positive definiteness of the correlation matrix. One way to solve such problem is to use Multivariate GARCH model [19]. This model is firstly applied to study the co-movement of different markets, which is to understand whether the volatility of one market leads to the volatilities of the other markets. Recently, this model has been applied to study correlations of asset returns and to investigate whether the correlations are time dependent.

Multivariate GARCH model is an extension of univariate GARCH model because it involves several assets instead of only one, thus we start to introduce some properties and applications of univariate GARCH model.

Let $y_t = log(\frac{S_t}{S_{t-1}})$ be the log return, $\mathcal{F}_t$ be the sigma field of y_t, and $\mathcal{F}_t = \sigma\{y_s : s \leq t\}$. Define $\mu_t = E(y_t|\mathcal{F}_{t-1})$ (the conditional mean) and $\sigma_t^2 = E((y_t - \mu_t)^2|\mathcal{F}_{t-1})$ (the conditional variance). Let $\epsilon_t = y_t - \mu_t$ be the innovations[2] and z_t be the standardized innovations which has standard normal distribution . Then the GARCH(p,q) model can be written as follows:

$$y_t = \mu_t + \epsilon_t \tag{4.9}$$

$$\epsilon_t = \sigma_t z_t \tag{4.10}$$

$$\sigma_t^2 = C + \sum_{i=1}^{p} \beta_i \sigma_{t-i}^2 + \sum_{i=1}^{q} \alpha_i \epsilon_{t-i}^2 \tag{4.11}$$

In principal, we should apply the kind of GARCH models which will reflect the leverage effect (e.t. GARCH models in section 3.1.1), but here we apply the standard GARCH (1,1) model, because the methods are the same for other GARCH models and most importantly the GARCH(1,1) has fewer parameters, this will avoid parameters to explored in Multivariate GARCH case. As the number of parameters become very big, estimation is very difficult and it will take huge amount of time to run the programing out.
The GARCH(1,1) model is:

$$\sigma_t^2 = C + \alpha\sigma_{t-1}^2 + \beta\epsilon_{t-1}^2$$

[2]Innovations is the time dependent residuals, the difficulties in using GARCH model is that we do not know what the distribution of the innovations is and usually people assume it as normal distribution.

Applying the univariate GARCH to multivariate GARCH, for N assets the y_t, μ_t, ϵ_t, z_t become $N \times 1$ vectors. Define the conditional variance matrix $H_t = var(y_t|\mathcal{F}_{t-1})$, and $\epsilon_t = \sqrt{H_t} z_t$ then we have:

$$\begin{aligned} H_t &= var(y_t|\mathcal{F}_{t-1}) \\ &= var(\mu_t + \epsilon_t|\mathcal{F}_{t-1}) \end{aligned}$$

Since μ_t is $\mathcal{F}_{t-1}$ measurable, so it is a constant given information up to time $t-1$, thus $var(y_t|\mathcal{F}_{t-1}) = H_t = var(\epsilon_t|\mathcal{F}_{t-1})$.

4.4.1 Application of DCC model

Model introduction

The introduction of all sorts of Multivariate GARCH model can be found in the reference. In this thesis, we applied DCC model[20], which is:

$$V_t = (1 - \theta_1 - \theta_2)\overline{V}_t + \theta_1 z_{t-1} z_{t-1}^{'} + \theta_2 V_{t-1} \tag{4.12}$$

$$V_t^\star = diag(\sqrt{V_{11,t}}, \cdots, \sqrt{V_{NN,t}}) \tag{4.13}$$

$$R_t = V_t^{\star-1} V_t V_t^{\star-1} \tag{4.14}$$

R_t is the correlation matrix. θ_1, θ_2 are unknown parameters and $\theta_1 + \theta_2 < 1$. V_t and $\overline{V_t}$ are the conditional and unconditional convariance matrix of z_t respectively. If we choose n samples and use i denote the i^{th} asset, then the elements of $\overline{V_t}$ can be calculated by

$$\begin{aligned} \overline{V}_{ii,t} &= \frac{1}{n}\sum_{t=1}^{n} z_{i,t}^2 \\ \overline{V}_{ij,t} &= \frac{1}{n}\sum_{t=1}^{n} z_{i,t} z_{j,t} \end{aligned}$$

Number of parameters to be estimated

If we use GARCH(1,1) model and have N assets, then number of parameters to be estimate in DCC model is $3N + 2$. In our case, we choose $N = 3$, so there are 9 parameters from the univariate garch model for each asset, which are $C = (C_1, C_2, C_3)^{'}$, $\alpha = (\alpha_1, \alpha_2, \alpha_3)^{'}$, $\beta = (\beta_1, \beta_2, \beta_3)^{'}$ and plus the parameters θ_1, θ_2 from the DCC mode. Let use θ denote the vector of the parameters $\theta = (C_1, C_2, C_3, \alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \theta_1, \theta_2)$.

Estimation methods

The most used estimation methods are maximium likelihood (ML) or qasi-maximum likelihood (QML). Technically they do not differ too much, but the formal method is used when we have correct assumption about the distribution of the residuals, while the later method allows the assumption of the residual distribution to be misspecified.

Maximum likelihood: Given $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ and assume they are from a certain family distribution $\{f(\cdot|\theta, \theta \in \Theta)\}$. Suppose θ_0 (unknown) is the true value of the parameter θ and we would like to find an estimator $\hat{\theta}$ which is as close to the true parameters as possible. This is done by maximizing the likelihood function or the log likelihood function. The likelihood function is usually expressed as :

$$\mathcal{L}(\theta|\epsilon_1, \epsilon_2, \cdots, \epsilon_n) = f(\epsilon_1, \epsilon_2, \cdots, \epsilon_n|\theta)$$

where $f(\epsilon_1, \epsilon_2, \cdots, \epsilon_n|\theta)$ is the joint density function of $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ conditioned on θ. If we write $f(\epsilon_1, \epsilon_2, \cdots, \epsilon_n|\theta)$ as $f_\theta(\epsilon_1, \epsilon_2, \cdots, \epsilon_n)$, by $f(x, y) = f(y)f(x|y)$, we have:

$$f_\theta(\epsilon_1, \epsilon_2, \cdots, \epsilon_n) = f_\theta(\epsilon_1)f_\theta(\epsilon_2|\epsilon_1)\cdots f_\theta(\epsilon_n|\epsilon_{n-1}\cdots\epsilon_1) \tag{4.15}$$

Quasi-maximum likelihood (pseudo likelihood)[21]: Pseudo likelihood conditions on the infinite past, which is:

$$f_\theta(\epsilon_1, \epsilon_2, \cdots, \epsilon_n, \cdots) = f_\theta(\epsilon_1|\epsilon_0, \epsilon_{-1}, \cdots)f_\theta(\epsilon_2|\epsilon_1, \epsilon_0, \cdots)\cdots f_\theta(\epsilon_n|\epsilon_{n-1}, \epsilon_{n-2}, \cdots)$$

If the very far past $\epsilon_0, \epsilon_{-1}, \cdots$ do not play an important role, we could simply drop them and work with the approximate likelihood function, which is the same as equation (4.15). Firstly let us look at the conditional probability $P(\epsilon_n|\epsilon_{n-1}, \cdots, \epsilon_1, \theta)$. By $z_n = \frac{\epsilon_n}{\sigma_n}$, we have:

$$\begin{aligned} P(\epsilon_n|\epsilon_{n-1}, \cdots, \epsilon_1, \theta) &= P(\epsilon_n \leq x|\epsilon_{n-1}, \cdots, \epsilon_1, \theta) \\ &= P(\sigma_n z_n \leq x|\epsilon_{n-1}, \cdots, \epsilon_1, \theta) \\ &= P(z_n \leq \frac{x}{\sigma_n}|z_{n-1}, \cdots, z_1, \theta) \end{aligned} \tag{4.16}$$

Then the corresponding density function of $P(\epsilon_n|\epsilon_{n-1}, \cdots, \epsilon_1, \theta)$ is:

$$\begin{aligned} f(\epsilon_n|\epsilon_{n-1}, \cdots, \epsilon_1, \theta) &= \frac{\partial P}{\partial x} \\ &= \frac{\partial P}{\partial z_n}\frac{\partial z_n}{\partial x} \\ &= \frac{1}{\sigma_n} f_z(\frac{\epsilon_n}{\sigma_n}) \end{aligned} \tag{4.17}$$

where f_z is the density function of random variable z. According equations (4.15) and (4.17), we have :

$$f(\epsilon_1, \epsilon_2, \cdots, \epsilon_n, |\theta) = \prod_{t=1}^{n} \frac{1}{\sigma_t} \frac{1}{\sqrt{2\pi\sigma_t^2}} e^{-\frac{\epsilon_t^2}{\sigma_t^2}} \tag{4.18}$$

so the log likelihood of $\epsilon_1, \epsilon_2, \cdots, \epsilon_n$ is:

$$log(\mathcal{L}(\theta|\epsilon_1, \epsilon_2, \cdots, \epsilon_n)) = -\frac{1}{2}\sum_{t=1}^{n} log(2\pi) + log\sigma_t^2 + \frac{\epsilon_t^2}{\sigma_t^2} \tag{4.19}$$

Here let us use k denote the number of assets, and $\epsilon_t = (\epsilon_{1,t}, \cdots, \epsilon_{k,t})'$. Since $\epsilon_t|\mathcal{F}_{t-1} \sim \mathcal{N}(0, H_t)$. $z_{1,t}, \cdots, z_{k,t}$ are from stand normal distribution, then the vector $(z_{1,t}, \cdots, z_{k,t})$ has multivariate normal distribution, and the density function is $f(z_{1,t}, \cdots, z_{k,t}) = \frac{1}{(2\pi)^{\frac{k}{2}}|H_t|^{\frac{1}{2}}} exp(-\frac{1}{2}\epsilon_t' H_t^{-1}\epsilon_t)$. Let us use $log(\mathcal{L}_M)$ denote the log likelihood of the multivariate GARCH model. By using the similar methods showed in (4.18), (4.19), we get:

$$log(\mathcal{L}_M) = -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + log(|H_t|) + \epsilon_t' H_t^{-1}\epsilon_t) \tag{4.20}$$

By $H_t = D_t R_t D_t$ and $z_t = D_t^{-1}\epsilon_t$, we can rewritten (4.20) as :

$$\begin{aligned} log(\mathcal{L}_M) &= -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + log(|H_t|) + \epsilon_t' H_t^{-1}\epsilon_t) \\ &= -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + log(|D_t R_t D_t|) + \epsilon_t' D_t^{-1} R_t^{-1} D_t^{-1}\epsilon_t) \\ &= -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + 2log(|D_t|) + log(|R_t|) + z_t' R_t^{-1} z_t) \\ &= -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + 2log(|D_t|) + \epsilon_t' D_t^{-1} D_t^{-1}\epsilon_t - z_t' z_t + log(|R_t|) + z_t' R_t^{-1} z_t) \end{aligned}$$

Let us use ϕ_1 and ϕ_2 denote the parameters in D and R respectively, then $log(\mathcal{L}_M)$ can be written as the sum of $L_v(\phi_1)$ and $L_c(\phi_2)$.

$$L_v(\phi_1) = -\frac{1}{2}\sum_{t=1}^{n}(klog(2\pi) + 2log(|D_t|) + \epsilon_t^{'}D_t^{-1}D_t^{-1}\epsilon_t) \tag{4.21}$$

$$L_c(\phi_2) = -\frac{1}{2}\sum_{t=1}^{n}(log(|R_t|) + z_t^{'}R_t^{-1}z_t - z_t^{'}z_t) \tag{4.22}$$

Equation (4.21) is the likelihood of a single asset, with a bit calculation we will get that L_v is the sum of individual GARCH likelihood, which means:

$$L_v(\phi_1) = -\frac{1}{2}\sum_{i=1}^{k}\sum_{t=1}^{n}(log(2\pi) + log\sigma_{i,t}^2 + \frac{\epsilon_{i,t}^2}{\sigma_{i,t}^2})$$

So, the estimation of the DCC model can be seperated into two steps. Firstly Fitting univariate GARCH (1,1) by maximizing L_v, then estimating the two parameters in correlation model by maximizing L_c.

simulating DDC model

Since the index is the weighted sum of individual stocks and it is very difficult to work with many stocks, I selected three stocks which have very similar trends as the AEX (The stocks I selected are ASML, PHI(Philips), and UL(Unibail-Rodamco)). In this way my synthetical AEX price process might be very close the real AEX and I could just work with 3 stocks instead of 25 stocks. As later on we would simulated the AEX with starting date $10/Oct/2011$ and maturity date $16/Dec/2011$, I selected the data in this period to compare the log return trends of stocks with the one of AEX.

From figure 4.13, we could see that my synthetic AEX return process is very close to the real AEX return process, so these three stocks are good to work with.

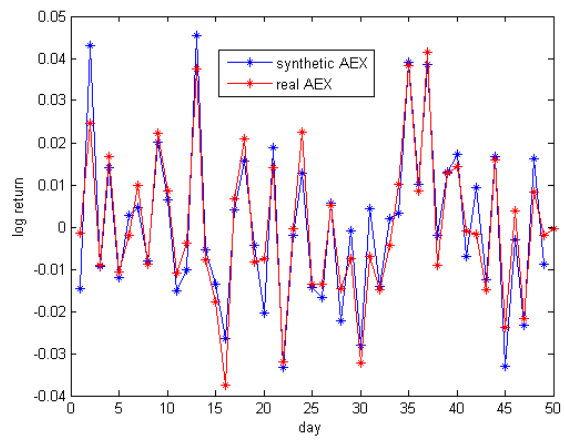


Figure 4.13: Selecting stocks which has similar log return trends and compare our synthetic AEX log return series with the one of real AEX.

Next, we will estimated the parameters in GARCH(1,1) and the DCC model, then we can get the correlations for in sample data (data used for fitting) and after that we will simulated the correlation for out of sample data (data for prediction)with the fitted parameters. Here the in sample data is in the period $02/Aug/2011 - 10/Oct/2011$. The out of sample data is from $11/Oct/2011$ to $19/Dec/2011$.

Algorithm

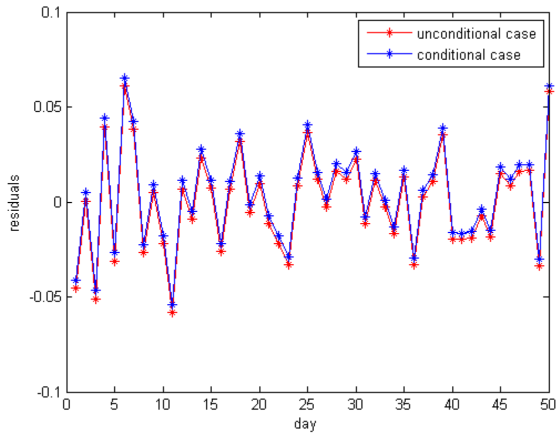
Step 1: Calculating the residuals ϵ_t. In principle we should calculate the residuals by $\epsilon_t = y_t - E_(y_t|\mathcal{F}_{t-1})$, but to make things simple, I use $\frac{1}{T}\sum_{t=1}^{T} y_t$ to substitute $E_(y_t|\mathcal{F}_{t-1})$. From figure 4.13(a) we can see that the residuals from the two methods almost have the same trend, so the simplification does not influence the result.

Step 2: Calculating the initial volatility σ_{t-1}. From the GARCH (1,1) model we know that σ_t depend on ϵ_{t-1} and σ_{t-1}. Since σ_t is $\mathcal{F}_{t-1}$ measurable, σ_{t-1} is $\mathcal{F}_{t-2}$ measurable, so I calculate σ_{t-1} by taking the standard deviation of all the data before time $t-1$ (the data from the first day till the $(t-2)^{th}$ day.

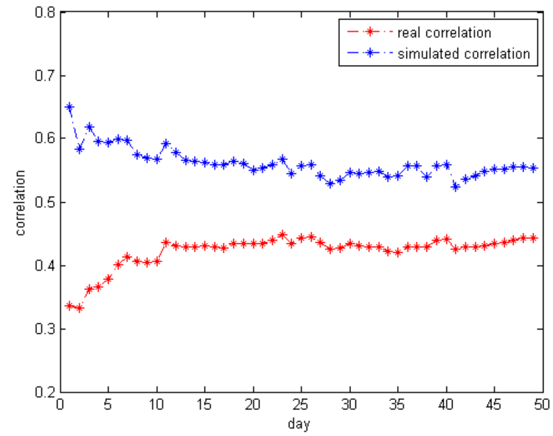
Step 3: Estimating the parameters in the univariate GARCH(1,1) model by minimizing $\sum_{t=1}^{n}(klog(2\pi) + 2log(|D_t|) + \epsilon_t^{'} D_t^{-1} D_t^{-1} \epsilon_t)$ and meanwhile get H_t. Here D is a $k \times k$ diagonal matrix (k is the number of assets and $k = 3$ in our case) and ϵ_t is a $k \times 1$ vector.

Step 4: Calculating the standardized residuals z_t by $\frac{\epsilon_{i,t}}{\sqrt{H_{ii,t}}}$, $i = 1, 2, 3$, then estimating parameters in the correlation model by minimizing $\sum_{t=1}^{n}(log(|R_t|) + z_t^{'} R_t^{-1} z_t - z_t^{'} z_t)$.

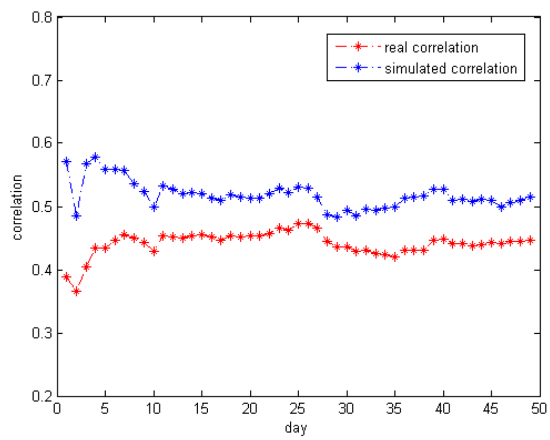
Step 5: Using the estimated parameters to simulate the future correlations and apply these correlations in simulating AEX.



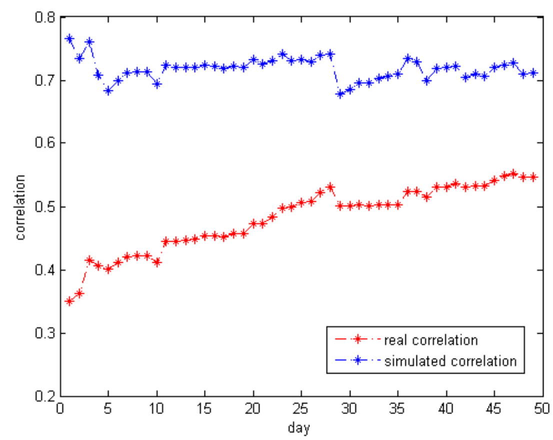
(a) residuals



(b) ASML-PHI



(c) ASML-UL



(d) PHI-UL

Figure 4.14: The graph (a)is the plot of residual ϵ_t under conditional mean $E_(y_t|\mathcal{F}_{t-1})$ and normal average $\frac{1}{T}\sum_{t=1}^{T} y_t$, as we can see that those two ways do not influence the residual process so much. Graph (b), (c), (d) are the plots of the in the sample correlation estimated by DCC model (the blue line) and the real correlation (the red line) between each pair stocks.

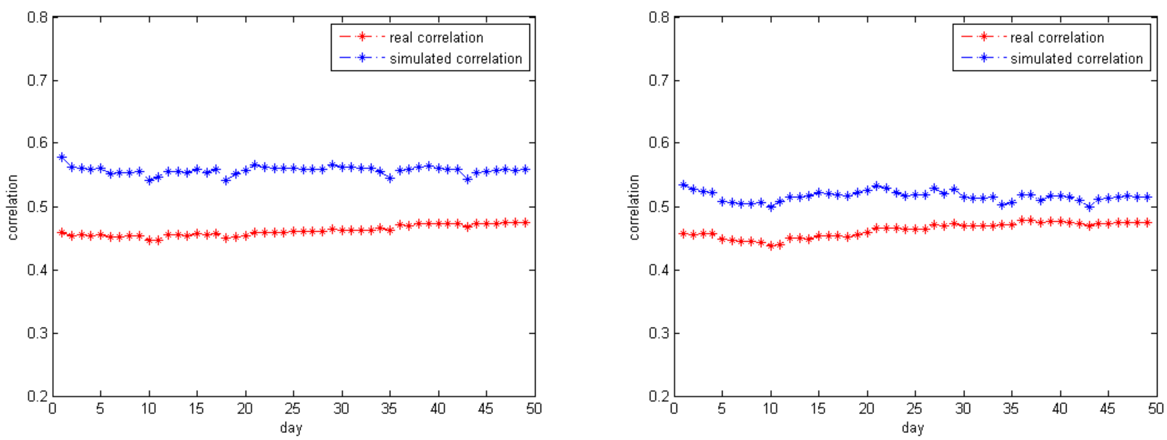
From figure 4.14 we can see that the good thing is that we can get good trends for all three stocks except picture (b) and (c) have some opposite trends for the first 5 days compared to the real correlation processes, but this will not be a big problem because this is the correlation got from the in sample data and when we try to simulate the correlation for the out of sample data, only the most recent data is very important. In fact, we can see that the trends for all three stocks in the last 25 days are very good. The problem is that my correlations are much higher than the real ones and this might be influenced by several reasons. From the Algorithm we explained before, the correlations depend on the volatility processes of the individual stocks, so finding the good volatility model is crucial. Our GARCH(1,1) model might be too simple and probably the volatility process does not follow this model. Furthermore we did not try the models with leverage considered and this also influences the volatility process. Another reason is that normal distributed noise are not working well and this is mentioned in many literatures. Lastly, it is difficult to guess the good starting values for a lot of parameters, and we always find the local minimum. Since the volatility processes of the individual stocks are mainly decided by the parameters, if we could not find good parameters, then we could not find good volatility processes and then this will lead to big error for the correlations because the correlation depend on the volatility. However, here we just would like to introduce a simple MGARCH model to see what the result would be. The improvements can be left for future studies.

Simulate correlations for the out of sample data

We use the *par* in table 4.2 to simulate the correlations for the time period $11/Oct/2011 - 19/Dec/2011$.

	par0	par
C_1	0.0003	0.0003001
C_2	0.0003	0.0003002
C_3	0.0001	0.0001000
α_1	0.002	0.881555
α_2	0.0001	0.001007
α_3	0.002	0.0020328
β_1	0.3	0.948229
β_2	0.03	0.039811
β_3	0.02	0.024913
θ_1	0.03	0.030000
θ_2	0.7	0.700000

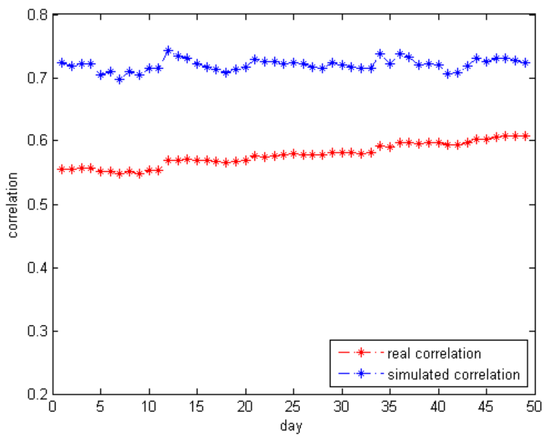
Table 4.2: par0 is the starting parameter, par is the optimized parameter. C_i, α_i, β_i are the coefficients of the GARCH(1,1) model: $\sigma_{i,t}^2 = C_i + \alpha_i \sigma_{i,t-1}^2 + \beta_i \epsilon_{i,t}^2$, $(i = 1, 2, 3)$.



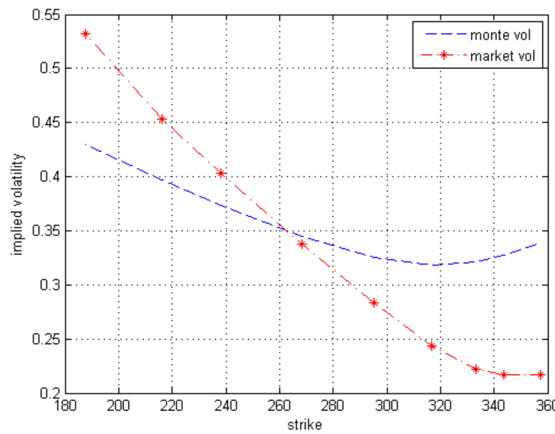
(a) ASML-PHI

(b) ASML-UL

Figure 4.15: Out of the sample correlation simulated by DCC model. Blue line is simulated by the model, and the red line is the realistic correlation between each pair of stocks.



(a) PHI-UL



(b) AEX skew

Figure 4.16: Out of the sample correlation simulated by DCC model. Blue line is simulated by the model, and the red line is the realistic correlation between each pair of stocks. Picture (b) is simulated skew (50,000 sample path) of our synthetic AEX with the simulated correlations applied.

As we can see from figure 4.15 (a), (b) and 4.16 (a), the trends of the simulated correlation process for each pair stocks are very similar to the real ones, this is very good. However, the skew did not become steeper when we use the simulated correlation to simulate the AEX (see figure 4.15(b)), the reason might be that the correlation under real probability and risk neutral probability are different.

Chapter 5

Discussion

The main aim of this project is to find a model that fits the skew of all stocks inside the index very well and to find or build correlation models to get steep skew for the index itself. The main findings are as follows:

1. Normally some models either fit well options with short maturity (e.g.[22]) or with long maturity (e.g. Heston model), but we found Bates model fits very well for both short and long maturity options. However, we found big errors for short maturity options when we tried to fit the skew surface, and we thought the reason might be that the parameters are very time dependent and the jump has significant effects on short maturity options. This is reasonable because stocks behave differently every day, furthermore, when big news happens, it will influence immediately the market for a short period and then the effect disappears.

2. When we tried to use $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t} Z_t^{(1)} \sqrt{\Delta t} + log(1 + J_t)\Delta N_t)$ to simulate the AEX with the well fitted stocks, we did not get the same skew for the call and put options, but we were successful when we tried to use $S_{t+1} = S_t * exp((r - q - \lambda\mu_J)\Delta t + \sqrt{V_t} Z_t^{(1)} \sqrt{\Delta t} + J_t\Delta N_t)$ to simulate the AEX. In our case, the jumps are small, so we could approximate the $log(1 + J_t)$ with J_t. We do not know whether simulating the index with the first formula will have the same problem or not for options with big jumps, but if it is does not work well for option with small jumps, then it means the model needs to be further improved.

3. For the correlation part, we found that assuming the same constant

correlation for all strikes failed to generate a steep skew for the index, because our skew is always crossed with the market skew around the at money strike (see figure 4.2 (b) and 4.11 (b)). Furthermore, with this assumption, the options with low strikes get under priced and the ones with high strikes get over priced. So, we should give high correlations for low strikes and low correlation for high strikes of the index. However, adding constant correlation (the same for all strikes) only for the Brownian motion processes (figure 4.4(b), or for both the Brownian and jump processes (figure 4.11(b)) did not make the skew steeper, the skew does not become steeper.

4. Without proper correlation models, for any other pricing model that one uses, the skew of the model crossed with the market skew. This is mentioned also in literature[4].

5. The idea of choosing the implied correlation as the reference point and giving higher correlation when S_t is less than the starting value, and lower correlation when S_t is larger than the starting value improved the steepness of the skew compared to the one with constant correlation applied. This can be seen from figure 4.6(b), 4.8(b), 4.9(d), and 4.12(b).

6. We tried to build a correlation model which is time dependent and with each element different, but it is very difficult to guarantee the positive definiteness and to work with higher dimensional cases. Although we could use an already developed matlab programme to change the non-positive definite matrix to a positive one, but it is very time consuming to run the program. It takes day and night to run 500.000 simulations. However, this model seems not to improve the steepness of the skew. The reason is that the stochastic correlation model allows the correlation processes to be negative correlated with the processes of single stocks, but this cannot gurantee that the correlation matrix will give high correlations for low strikes and low correlations for high strikes of the index, and this is what we need to get a steeper skew according the previous results.

7. In the end we tried the Dynamic Conditional Correlation (DCC) model, which is easy to guarantee the positive definiteness of the correlation matirx. However, even if we could get the very same correlation trend

as the market one, the skew is not improved at all. So it might be that the correlations under real probability and risk neutral are different.

8. Although some of our built models improved the steepness of the skew, the highest volatility level for the lowest strike is always around 0.4 and it is much below the market one (around 0.55). We have not figure out the reason and have not found a way to further improve the steepness of the skew for lower strikes. Generally speaking, all the models are worth to try, such as the GARCH models with leverage and the Wishart model mentioned in the following Chapter, but I would suggest to still focus on constant correlation models which give high correlation for the low strikes and low correlation for the high strikes of the index, becasue according the results of model 5, probably the two models mentioned in the Future study will not work.

Chapter 6

Future study

We did not try the GARCH models with leverage allowed, this can be left for a future trial to see whether the result is further improved.

Except multivariate GARCH model, there is a Wishart model[23] which is recently used to model the stochastic convariance matrix processes. In this model the volatility processes is not a scalar but a matrix process. For a single asset, the process is:

$$\begin{aligned} \frac{dS_t}{S_t} &= rdt + Tr(\sqrt{V_t}dB_t) \\ dV_t &= (\theta U^T U + MV_t + V_t M^T)dt + \sqrt{V_t}dW_t U + U^T dW_t^T \sqrt{V_t} \end{aligned} \tag{6.1}$$

r is the interest rate
$U \in \widetilde{S}_n^+(R)$: the sets of real symmetric positive definite square matrix.
$M \in \widetilde{S}_n^-(R)$: the sets of real symmetric negative definite square matrix.
$\theta > n-1$, $V_0 \in \widetilde{S}_n^+(R)$, W_t and B_t are matrix Brownian motions.

Let R be a matrix descriping the correlation between B_t and W_t, and Z_t be a matrix Brownian motion which is independent of W_t, then

$$dB_t = dW_t R + dZ_t \sqrt{I_n - R^T R}$$

where I_n is a $n \times n$ identity matrix.

Definition 1: *Consider $k \geq n$ independent random vectors $X_1, X_2, \cdots, X_k$*

of R^n distributed according to multivariate Gaussian distributions $N(0,\Sigma)$. Consider also k non-random vectors of R^n denoted by $\mu_1,\mu_2,\cdots,\mu_k$ The distribution of

$$W=\sum_{i=1}^{k}(X_i+\mu_i)(X_i+\mu_i)^T$$

is a non-central Wishart distribution with k degrees of freedom, denoted as $W_k(\mu,\Sigma)$ where $\mu=\sum_{i=1}^{k}\mu_i\mu_i^T$

Theorem 1 (Simulating central Wishart random variables): *Let Σ be a symmetric positive definite matirx, let θ be an integer strictly greater than n and let $C\in M_n(R)$ defined as follows*

$$i\in[1,n], C^{ii}=\chi_{\theta-i+1}$$
$$1\leq j<i\leq n, C^{ij}=N^{ij}$$

where the $\chi_{\theta-i+1}$ denote the square roots of independent χ^2 random variables with $\theta-i+1$ degree of freedom, the N^{ij} are independent ordinary Gaussian random variables (also independent of the χ^2) and where all other entries of C are zero. Then the random matrix $\sqrt{\Sigma}CC^T\sqrt{\Sigma}$ has the law of $W_\theta(0,\Sigma)$.

Proposition 1 (Simulating non-central Wishart random variables): *Let θ satisfying the conditions above and let R and Σ be symmetric positive matrix. Let $(m_i)_{1\leq i\leq n}$ be the i^{th} column of $\sqrt{R}$, let now the $(Z_i)_{1\leq i\leq n}$ be independent vectorial Gaussian variables $\sim N(0,I_n)$ and Y be a central Wishart random variable $\sim W_{\theta-n}(0,I_n)$ independent of the $(Z_i)_{1\leq i\leq n}$, we have:*

$$\sum_{i}^{n}(m_i+\sqrt{\Sigma}Z_i)(m_i+\sqrt{\Sigma}Z_i)^T+\sqrt{\Sigma}Y\sqrt{\Sigma}\sim W_\theta(R,\Sigma)$$

Theorem 2: *If the process $(V_t)_{t\geq 0}$ has dynamic (1) then conditionally on V_t, V_{t+s} has the distribution $W_\theta(\Delta(s)V_t\Delta(s)^T,\Sigma(s))$, where*

$$\Delta(s)=e^{sR}$$
$$\Sigma(s)=\int_0^s e^{xR}U^TUe^{xR^T}\mathrm{d}x$$

Proposition 2: *Let $X \sim W_\theta(R, \Sigma)$ and let $\mu = R + \theta\Sigma$ be its expectation matrix. Then we have*

$$E(X-\mu)^2 = \frac{1}{\theta}((R+\theta\Sigma)^2 - R^2 + (R+\theta\Sigma)Tr(R+\theta\Sigma) - RTr(R))$$

Define $\mu_t(s) = e^{sM} V_t e^{sM^T}$, where M is the one mentioned in equation (1) and recall theorem 2 and proposition 2, then we have:

$$\begin{aligned} E(V_{t+s}|V_t) &= \mu_t(s) + \theta\Sigma(s) \\ \theta Var(V_{t+s}|V_t) &= (\mu_t(s) + \theta\Sigma)^2 - \mu_t(s)^2 \end{aligned}$$
$$+(\mu_t(s) + \theta\Sigma(s))Tr(\mu_t(s) + \theta\Sigma(s)) - \mu_t(s)Tr(\mu_t(s)))$$

In simulation, people normally try to approximate the distribution of V_{t+s} conditioned on V_t by a non-central Wishart random variable $W_\theta(R, \Sigma)$, where R, and Σ are symmetrical positive definite and they are chosen to match the first two moments. The procedure is to use first moment matching to match the expectation, however, this does not guarantee that the second moment (Σ) will be positive definite, then the second moment matching is required. For the details about moment matching, please see[23].

However, the Wishart model for one assets is already quite complicated because of the volatility matrix process, so the application in more than one asset would be quite difficult. Due to the time limit, we did not try this model, and it might be interesting for future research.

Another difficulty we got in working with the index is the high dimensional problem, so we need a way to reduce the dimension and meanwhile do not lose too much information. The common used method is **Principal Component Analysis (PCA)**. Here we will give a simple introduction about this method which might be useful for future research.

PCA is a transformation, which transforms the dataset to a new coordinate system such that the greatest variance by any projection of the dataset comes to lie on the first coordinate. The second greatest projected variance lies on the second coordinate and so on. For simple case, we only take the linear transformation for an example. The purpose of PCA is to reduce the dimensionality and meanwhile do not lose so much information. Since the mean does not give so much important information, here the information are mainly referring to the variance. Let $X = (X_1, X_2, X_3, \cdots, X_m)$ be a m dimensional vector, centering X by $X_i - mean(X_i)$, for $i = (1, 2, \cdots, m)$, then we get X_i has mean zero. Let P as a m dimensional unit vector, so P has norm 1. Let us say A is the projection of X onto P, then we have $A = X^T P$, so $var(A) = P^T cov(X^T, X)P$. We would like to find the first principal component has the first largest variance. The second principal has the second largest variance and it is orthogonal to the first principal component and so on the k^{th} largest principal has the k^{th} largest variance and it is orthogonal to all the previous components. If we let $\lambda = (\lambda_1, \lambda_2, \cdots, \lambda_m)$ be the eigenvalues of the $\Sigma = cov(X^T, X)$, and P be the unit eigenvectors of the corresponding eigenvalues. Then we sort λ by descending order and find the first k largest $\lambda' s$ which represent more than 85 percent information. In the end we find corresponding eigenvectors for those $\lambda' s$ and we get a $k \times m$ matrix A, $(k \leq m)$. Writing A as $(Z_1, Z_2, Z_3, \cdots, Z_k)^T$, we get the following:

$$Z_1 = p_{11}X_1 + p_{12}X_2 \cdots + p_{1m}X_m$$

$$Z_2 = p_{21}X_1 + p_{22}X_2 \cdots + p_{2m}X_m$$

$$Z_3 = p_{31}X_1 + p_{32}X_2 \cdots + p_{3m}X_m$$

$$\vdots$$

$$Z_k = p_{k1}X_1 + p_{k2}X_2 \cdots + p_{km}X_m$$

In this way, we reduce the dimension from m to k.

Bibliography

[1] Stephen Figlewski and Xiaozu Wang. Is the "leverage effect" a leverage effect? *Review of Finance*, 8:1–40, 2000.

[2] Nikudj Kapadia Gurdip Bakshi and Dilip Madan. Stock return characteristics, skew laws and the differential pricing of individual equity options. *Review of Financial Studies*, 16:101–143, 2003.

[3] Nicolas P. B. Bollen and Robert E. Whaley. Does net buying pressure affect the shape of implied volatility functions? *The Journal of Finance*, LIX, NO. 2:711–753, 2004.

[4] Nicole Branger and Christian Schlag. Why is the index smile so steep? *Review of Finance*, 8:109–127, 2004.

[5] Sander Sepp. Modeling stock return of correlation. *Master thesis, Universiteit van Amsterdam*, 2011.

[6] Chun Chou Wu. The garch option pricing model: A modification of lattice approach. *Review of Quantitative Finance and Accounting*, 26:55C–66, 2006.

[7] YaoChun Liu ShihFeng Huang and JingYu Wu. An empirical study on implied garch models. *Journal of Data Science*, 10:87–105, 2012.

[8] Robert J. Elliott. A discrete time equivalent martingale measure. *Mathematical Finance*, 8:127C–152, 1998.

[9] Robert J. Elliott Alex Badescu and Tak Kuen Siu. Esscher transforms and consumption-based models. *Insurance: Mathematics and Economics*, 45:337–347, 2009.

[10] Alexandru M. Badescu and Reg J. Kulperger. Garch option pricing: A semiparametric approach. *Insurance: Mathematics and Economics*, 43:69–84, 2008.

[11] Takayuki Morimoto Koichi Maekawa, Sangyeol Lee and Ken ichi Kawai. Jump diffusion model with application to the japanese stock market. *Mathematics and Computers in Simulation*, 78:223–C236, 2008.

[12] Chongfeng Wu Weidong Xu and Hongyi Li. Foreign equity option pricing under stochastic volatility model with double jumps. *Economic Modelling*, 28:1587–1863, 2011.

[13] E. Miglio and C. Sgarra. A finite element framework for option pricing with the bates model. *Working paper,Politecnico di Milano*, 2008.

[14] Jin Chuan. Duan. The garch option pricing model. *Mathematical Finance*, 5:13–32, 1995.

[15] Steven L. Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *The Review of Financial Studies*, 6:327–343, 1993.

[16] Jonathan E.Ingersoll John C. Cox and Stephen A.Ross Jr. A theory of the term structure of interest rates. *Econometrica*, 53:385–408, 1985.

[17] David S. Bates. Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options. *The Review of Financial Studies*, 9:69–107, 1996.

[18] Wim Schoutens Hansjörg Albrecher, Philipp Mayer and Jurgen Tistaert. The little heston trap. *Working paper, Radon Institute*, 2006.

[19] Luc Bauwens. Multivariate garch models: A survey. *Journal of applied economics*, 21:79–109, 2006.

[20] Robert Engle. Dynamic conditional correlation: A simple class of multivariate generalized autoregressive conditional heteroskedasticity models. *Journal of Business and Economic Statistics*, 20:339–350, 2002.

[21] A.W. van der Vaart. Time series version 122012. *class notes of Time Series course*, pages 337–347, 2012.

[22] Alexey Medvedev and Olivier Scaillet. Approximation and calibration of short-term implied volatilities under jump-diffusion stochastic volatilit. *The Review of Financial Studies*, 20:427–459, 2007.

[23] Pierre Gauthier and Dylan Possamai. Efficient simulation of the wishart model. *Working paper*, 20:1–56, 2009.

Chapter 7

Appendix (matlab code)

1. Call option pricing formula of Bates model

```
function ...
    call_price=callBatescf(s0,K,tau,r,q,v0,theta,rho,kappa,...
                                sigmav,lambda,muJ,vJ)
%s0:      starting stock price (stock price at time zero)
%K:       strike price
%tau:     time to maturity
%r:       risk free rate
%q:       dividend yield
%v0:      initial variance
%theta:   long run variance
%kappa:   mean reverting parameter
%sigmav: volatility of volatility
%lambda:  jump rate
%muJ:     mean of jump size
%vJ:      variance of jump size
% fai is  Characteristic function of Bates model
%N1, N2 is the probability that stock ends in the money
 function f2=fai(y)
d=sqrt((rho*sigmav*y*1i-kappa).^2+sigmav^2*(y*1i+y.^2));
g=(kappa-rho*sigmav*y*1i-d)./(kappa-rho*sigmav*y*1i+d);
A=1i*y.*(log(s0)+(r-q)*tau);
B=((theta*kappa)/sigmav^2)*((kappa-rho*sigmav*y*1i-d)*tau-...
                          2*log((1-g.*exp(-d*tau))./(1-g)));
C=((v0/sigmav^2)*(kappa-rho*sigmav*y*1i-d).*(1-exp(-d*tau)))...
                                        ./(1-g.*exp(-d*tau));
D=-lambda*muJ*y*tau*1i+lambda*tau*(((1+muJ).^(y*1i))...
                        .*exp(0.5*vJ*(1i*y).*(1i*y-1))-1);
```

```
f2=exp(A+B+C+D); %f2 is the Characteristic function of N2.

 end
%y1 is the real function in N1 to be integrated.
 function y1=fun1(y)
    y1=real(exp(-1i*y*log(K)).*fai(y-1i)./(1i*y*fai(-1i)));
 end
%y2 is the real function in N2 to be integrated.
 function y2=fun2(y)
    y2=real(exp(-1i*y*log(K)).*fai(y)./(1i*y));
 end
P1=0.5+(1/pi)* quadl(@fun1,0,200,[],[]); %P1 refers N1
P2=0.5+(1/pi)* quadl(@fun2,0,200,[],[]); %P2 refers N2
call_price=exp(-q*tau)*s0*P1-exp(-r*tau)*K*P2;
%call option price
end
```

2. Fitting ING by minimizing $\sum_{K<S_0}(Put_{thoretic} - Put_{market})^2 + ...$ $\sum_{K>S_0}(Call_{thoretic} - Call_{market})^2$. In the following code, the "options" for fminsearchbnd do not need to set Algorithm.

```
function[par,output,bscall,batescall,bsput,batesput,...
callerro,puterro,bsimpvol,batesvol,volerro,time]=INGcpT2fbnd()
tic;
format long
s0=5.87;r=0.013088666;q=0;
t0=datestr('10/10/2011','dd/mmm/yyyy');  %starting date
T=datestr('12/16/2011','dd/mmm/yyyy');    %maturity date
num=datenum(T)-datenum(t0);  %number of days
numT=num-2*floor(num/7);     %trading days
tau=numT/252;              %annulized number of days
kvec=[9.948115203,8.289615964,7.204066372,6.124477245,...
                  4.948775628,3.76562343,2.247195106];
%kvec is the strike price
len=length(kvec);

bsimpvol=[59.32674263,57.18438803,58.6675566,65.9186029,...
78.77273049,95.41717773,123.9269735]/100;
[Call,Put]=blsprice(s0,kvec,r,tau,bsimpvol,0);
bscall=Call; %Black-Scholes call option price
bsput=Put;

%x0=[v0,theta,rho,k,sigma,lambda,muJ,vJ];
par0=[0.5,0.7,-0.9,3,2,0.001,-0.001,0.002];%fitting for long
```

```
lb=[0,0,-1,0,0,0,-20,0];
ub=[5,5,1,10,5,10,10,1];

%the following three line is for no jump case
%par0=[0.5,0.5,-0.7,3,2];
%lb=[0,0,-1,0,0];
%ub=[5,5,1,10,5];
    batescall=zeros(1,len);
    batesput=zeros(1,len);
    batesvol=zeros(1,len);

 options=optimset('MaxIter',500000,'TolX',1e-10,...
'MaxFunEvals',50000,'Algorithm','interior-point');

%alternative optimization methods 'fminsearchbnd'
%[par,output]=fminsearchbnd(@BatesDifferences,par0,
%                                  lb,ub,options);

%A=(-1)*eye(5); b=[0,0,1,0,0]; %for no jump case
A=(-1)*eye(8); b=[0,0,1,0,0,-0.001,20,0]; %jump case
[par,output]=fmincon(@BatesDifferences,par0,A,b,[],[],...
lb,ub,[],options);

 function y=BatesDifferences(par)
 % parameters for jump case
 v0=par(1);theta=par(2);rho=par(3);k=par(4);
 sigma=par(5);lambda=par(6);muJ=par(7);vJ=par(8);

 %parameters for no jump case
 %v0=par(1);theta=par(2);rho=par(3);k=par(4);sigma=par(5);
    u1=0;
    u2=0;
    cont=0;
     for i=1:len
        X=kvec(i);
        cont=cont+1;
%batescall(cont)= callBatescf(s0,X,tau,r,q,v0,theta,rho,k,...
%                                            sigma,0,0,0);
batescall(cont)= callBatescf(s0,X,tau,r,q,v0,theta,rho,k,...
                                       sigma,lambda,muJ,vJ);
batesput(cont)=batescall(cont)+X*exp(-r*tau)-s0;
          if(X>s0)
batesvol(cont)=fzero(@(v)cimpvol(v,s0,X,tau,r,...
                           batescall(cont)),0.6);

```

```
    u1=u1+(bscall(cont)−batescall(cont))^2;
          else
batesvol(cont)=fzero(@(v)pimpvol(v,s0,X,tau,r,...
                            batesput(cont)),0.9);
    u2=u2+(bsput(cont)−batesput(cont))^2;
          end
     end
    y=u1+u2;
 end
    callerro=abs(bscall−batescall);
    puterro=abs(bsput−batesput);
    volerro=abs(bsimpvol−batesvol);

%cimpvol is to get implied volatility for out of money call
%by inversing Black−Scholes formula.
function y1=cimpvol(v,B,k1,T1,r1,c0)
    d1=(log(B/k1)+(r1+v^2/2)*T1)/(v*sqrt(T1));
    d2=d1−v*sqrt(T1);
    call=B*normcdf(d1,0,1)−k1*exp(−r1*T1)*normcdf(d2,0,1);
    y1=c0−call;
 end

%pimpvol is to get implied volatility for out of money put
% by inversing Black−Scholes formula.
 function y2=pimpvol(v,B,k1,T1,r1,p0)
  d1=(log(B/k1)+(r1+v^2/2)*T1)/(v*sqrt(T1));
  d2=d1−v*sqrt(T1);
  put=k1*exp(−r1*T1)*normcdf(−d2,0,1)−B*normcdf(−d1,0,1);
  y2=p0−put;
 end
figure(1)
plot(kvec,bscall,'−.r*'),xlabel('strike price'),...
ylabel('call option price'),hold on
plot(kvec,batescall,'−−b'),xlabel('strike price'),...
ylabel('call option price')
legend('market option price','model option price');
figure(2)
plot(kvec,bsput,'−.r*'),xlabel('strike price'),...
ylabel('put option price'),hold on
plot(kvec,batesput,'−−b'),xlabel('strike price'),...
ylabel('put option price')
legend('market option price','model option price',...
'location','NorthWest');
figure(3)
plot(kvec,bsimpvol,'−.r*'),xlabel('strike price'),...
```

```
ylabel('implied volatility'),hold on
plot(kvec,batesvol,'--b'),xlabel('strike price'),...
ylabel('implied volatility')
legend('market vol','model vol');
toc;
time=toc;
end
```

3. Fitting skew surface

```
function [par,output,s,bsput,batesput,priserro,...
    bsimpvol,impbatesvol,absvolerro,runtime]=UNAmulT6surface()
%s: Adjusted spot price by deducing the dividend
%bsput: put option price from Black-Schole's model
%batesput: put option price from Bates model.
%impbatesvol:implied volatility from Bates model
%absvolerro: absolute value of volatility error

%p0:spot price
%r:risk free rates of 6 different maturity
%FDIV: Comulative dividents
%kmat:market strike
%bsimpvol: market volatility
tic;
format long
p0=23.61;q=0;
r=importdata('UNA6Tr.dat');
FDIV=importdata('UNA6TFDIV.dat');
kmat=importdata('UNA6Tkmat.dat');
bsimpvol=importdata('UNA6Tvol.dat');
t0=datestr('10/10/2011','dd/mmm/yyyy');  %starting date
%six different maturities
T1=datestr('11/18/2011','dd/mmm/yyyy');
T2=datestr('12/16/2011','dd/mmm/yyyy');
T3=datestr('3/16/2012','dd/mmm/yyyy');
T4=datestr('6/15/2012','dd/mmm/yyyy');
T5=datestr('9/21/2012','dd/mmm/yyyy');
T6=datestr('12/21/2012','dd/mmm/yyyy');
vecT=char(T1,T2,T3,T4,T5,T6);
numvecT=datenum(vecT);
[row,col]=size(bsimpvol);
bsput=zeros(row,col);
Call=zeros(row,col);
Put=zeros(row,col);
```

```
tau=zeros(row,1);
s=zeros(row,1);
for i=1:row
    tau(i)=(datenum(numvecT(i))-datenum(t0))/365;
     s(i)=p0-exp(-r(i)*tau(i))*FDIV(i);
     for j=1:col
 [Call(i,j),Put(i,j)]=blsprice(s(i),kmat(i,j),...
                      r(i),tau(i),bsimpvol(i,j),0);
     bsput(i,j)=Put(i,j);
    end
end
lb=[0,0,-1,0,0,0,-20,0];
ub=[10,10,1,20,20,20,10,1];
%par0=[v0,theta,rho,k,sigma,lambda,muJ,vJ];
par0=[0.1,0.1,-0.1,5,0.7,0.1,-0.1,0.2];
A=(-1)*eye(8); b=[0,0,1,0,0,-0.02,20,0];
options=optimset('MaxIter',500000,'TolX',1e-10,...
'MaxFunEvals',50000,'Algorithm','interior-point');
%alternative optimization methods 'fminsearchbnd'
%[par,output]=fminsearchbnd(@BatesDifferences,par0,
%                                 lb,ub,options);
[par,output]=fmincon(@BatesDifferences,par0,A,b,[],[],...
                                 lb,ub,[],options);
%function BatesDifferences returns the parameters through
%minimizing the premium over all maturities.
function y=BatesDifferences(par)
        u1=0;
        u2=0;
        batescall=zeros(row,col);
        batesput=zeros(row,col);
        priserro=zeros(row,col);
        v0=par(1);theta=par(2);rho=par(3);k=par(4);sigma=par(5);
       lambda=par(6);muJ=par(7);vJ=par(8);
for j1=1:row

     for j2=1:col
    batescall(j1,j2)= callBatescf(s(j1),kmat(j1,j2),...
    tau(j1),r(j1),q,v0,theta,rho,k,sigma,lambda,muJ,vJ);

    batesput(j1,j2)=batescall(j1,j2)+kmat(j1,j2)*...
                         exp(-r(j1)*tau(j1))-s(j1);
    priserro(j1,j2)=abs(bsput(j1,j2)-batesput(j1,j2));
          if(kmat(j1,j2)>s(j1))

    u1=u1+(Call(j1,j2)-batescall(j1,j2))^2;
```

```
            else

    u2=u2+(Put(j1,j2)-batesput(j1,j2))^2;
            end
      end
end
    y=u1+u2;
end
%function impvol is to derive the volatility...
%in Black-Scholes formula,given the put option price...
%from Bates model
function f=impvol(v,s,kmat,tau,r,row,col,batesput,i1,i2)
     bsoption=zeros(row,col);
     d1=(log(s(i1)/kmat(i1,i2))+...
     (r(i1)+v^2/2)*tau(i1))/(v*sqrt(tau(i1)));
%bsoption is the put price from Black-sholes model
     d2=d1-v*sqrt(tau(i1));
     bsoption(i1,i2)=kmat(i1,i2)*exp(-r(i1)*tau(i1))...
     *normcdf(-d2,0,1)-s(i1)*normcdf(-d1,0,1);
     f=batesput(i1,i2)-bsoption(i1,i2);
end
impbatesvol=zeros(row,col);
absvolerro=zeros(row,col);
for i1=1:row
    for i2=1:col
impbatesvol(i1,i2)=fzero(@(v)impvol(v,s,kmat,tau,r,...
                  row,col,batesput,i1,i2),0.3);
absvolerro(i1,i2)=abs(bsimpvol(i1,i2)-impbatesvol(i1,i2));
    end
end
%plots for figure 3.2 in Chapter 3.
i3=1;
count=0;
while(i3<12)
i4=i3+1;
count=count+1;
figure(i3)
plot(kmat(count,:),bsput(count,:),'-.r*'),xlabel('strike'),...
ylabel('option price');
hold on
plot(kmat(count,:),batesput(count,:),'--b');xlabel('strike'),...
ylabel('option price');
grid on
legend('market price','model price','Location','NorthWest');
figure(i4)
```

```
plot(kmat(count,:),bsimpvol(count,:),'-.m*'),xlabel('strike'),...
ylabel('implied volatility');
hold on
plot(kmat(count,:),impbatesvol(count,:),'--k');xlabel('strike'),...
ylabel('implied volatility');
grid on
legend('market impvol','model impvol');
i3=i4+1;
end
% Surface plots for figure 3.3 in Chapter3
taumat=(ones(7,1)*tau')';
figure(i3)
surf(taumat,kmat,bsimpvol),xlabel('Maturity(years)'),...
ylabel('Strike'),title('implied volatility plot'),....
zlabel('Implied Volatility'); hold on;
surf(taumat,kmat,impbatesvol),xlabel('Maturity(years)'),...
ylabel('Strike'),title('implied volatility plot'),...
zlabel('Implied Volatility');
figure(i3+1)
surf(taumat,kmat,impbatesvol-bsimpvol),xlabel('Maturity(years)'),...
ylabel('Strike'),title('skew erro surface plot'),...
zlabel('Volatility difference');
figure(i3+2)
surf(taumat,kmat,bsput),xlabel('Maturity(years)'),...
ylabel('Strike'),title('option price surface'),...
zlabel('put option price'); hold on;
surf(taumat,kmat,batesput),xlabel('Maturity(years)'),...
ylabel('Strike'),title('option price surface'),...
zlabel('put option price');
figure(i3+3)
surf(taumat,kmat,bsput-batesput),xlabel('Maturity(years)'),...
ylabel('Strike'),title('price erro plot'),...
zlabel('price difference');
toc;
runtime=toc;
end
```

4. Simulate AEX with Correlation model 1, 2, 3.

```
%monte carlo simulation with 24 stocks except TNTE
function [bsdexput,montedexput,priceerro,...
 indexvol,volatility,volerro,time]=cor_model_put()
tic;
t0=datestr('10/10/2011','dd/mmm/yyyy');  %starting date
```

```
T=datestr('12/16/2011','dd/mmm/yyyy');    %maturity date
num=datenum(T)-datenum(t0);  %number of days
numT=num-2*floor(num/7);%number of trading days
tau=numT/252; %annulized time to maturity
dt=1/252; %\Delta_t
r=0.013088666*ones(25,1);%risk free rate
indexr=0.013088666;
%AEXFDIV=1.513154636;%AEX comulative dividend
indexs0=293.5-1.513154636*exp(-indexr*tau);
%indexk:index strikes
indexk=[357.4372334,343.752007,333.2139058,316.6557355,...
295.4245611,268.4832617,237.9895254,216.3888022,187.4924704];
%indexvol:index volatilities
indexvol=[21.7344,21.6778,22.1872,24.3946,28.3,...
          33.7902,40.2992,45.28,53.204]/100;
[Call, Put]= blsprice(indexs0,indexk,indexr,tau,indexvol,0);
bsdexcall=Call;
bsdexput=Put;
%bsdexput:Black-Scholes put option price of AEX.
len=length(indexk);
volatility=zeros(1,len);
%s0:spot price of 25 stocks
s0=importdata('s0.dat');
%numShares:number of shares of 25 stocks
numShares=importdata('numShares.dat');
divisor=sum(numShares.*s0)/indexs0;
newnumshares=numShares/divisor;
%par:fitted parameters from minimizing premium
par=importdata('AEXpar.dat');
v0=par(:,1);theta=par(:,2);rho=par(:,3);kappa=par(:,4);
sigma=par(:,5);lambda=par(:,6);muJ=par(:,7);vJ=par(:,8);
%the following is for calculating implied correlation...
%according formula in section 4.3 (page 44)
sigmai=importdata('sigmai.dat');
%sigmai: at the money volatilities of 25 stocks
weights=importdata('weights.dat');
%weights: real weights of 25 stocks on '10/10/2011'
avevol=sum(sigmai.*weights);
%avevol:average of the volatilities
avesqvo=sum((sigmai.^2).*(weights.^2));
Ivol=0.283;
%Ivol:at the money volatility of AEX
imprho=(Ivol^2-avesqvo)/(avevol^2-avesqvo);
%simulation part:
g=zeros(1,numT);
```

```
%g:value of the constant correlation
m=500000;%number of simulations
indexsT=zeros(1,len);
%indexsT:AEX stock price at maturity time
n=25;
for j1=1:m   %number of simulation times
     g(1)=imprho;
     indp=zeros(1,(numT+1));
%indp:vection of AEX S_t(stock price at each t)
     a=s0;
     indp(1)=indexs0;
%z(:,1): Brownian motion random variables of stock...
%price processes
%z(:,2):Brownian motion random variables of volatility...
%processes
%correlation must be in (0,1) to guarantee the positive...
%definiteness of the correlation matrix. If the correlation...
%larger than 1, we set it to 1. If the correlation ...
%less than 0, we set it to zero.
for j2=1:numT
     if(0<g(j2)&&g(j2)<1)
correlation=g(j2)*ones(n,n)-g(j2)*eye(n)+eye(n);
      L = chol(correlation,'lower');
      epsilon=normrnd(0,1,n,2);
       z(:,1)=L*epsilon(:,1);
       z(:,2)=epsilon(:,2);
     elseif(g(j2)≥1)
       z(:,1)=ones(n,1).*normrnd(0,1);
       z(:,2)=normrnd(0,1);
     elseif(g(j2)==0)
       epsilon=normrnd(0,1,n,2);
       z(:,1)=epsilon(:,1);
       z(:,2)=epsilon(:,2);
     end
zJ=normrnd(log(1+muJ)-vJ/2,vJ,n,1);
J=exp(zJ)-1; N=poissrnd(lambda*dt);
%s=a.*exp((r-lambda.*muJ-v0/2)*dt+...
  %sqrt(v0).*z(:,1)*sqrt(dt)+log(1+J).*N);
s=a.*exp((r-lambda.*muJ-v0/2)*dt+...
          sqrt(v0).*z(:,1)*sqrt(dt)+J.*N);
vol=v0+kappa.*(theta-v0)*dt+sigma.*sqrt(v0)...
.*(rho.*z(:,1)+sqrt(1-rho.^2).*z(:,2))*sqrt(dt);
       a=s;
v0=max(vol,0);
indp(j2+1)=sum(newnumshares.*a);
```

```
      beta=0.4;
      alpha=0.6;
   if(indp(j2+1)<indexs0)
%in the following are model1, model2, model3 (thesis page ...
    49-50)
g(j2+1)=max(imprho+beta*(indexs0-indp(j2+1))/indexs0,0);
%g(j2+1)=max(imprho+beta*sqrt((indexs0-indp(j2+1))/indexs0),0);
%g(j2+1)=max(imprho+beta*((indexs0-indp(j2+1))/indexs0)^(1/4),0);
   else
g(j2+1)=max(imprho-alpha*sqrt((indp(j2+1)-indexs0)/indexs0),0);
%g(j2+1)=max(imprho-alpha*((indp(j2+1)-indexs0)/indexs0),0);
%g(j2+1)=max(imprho-alpha*((indp(j2+1)-indexs0)/indexs0)^(1/4),0);
   end
end
indexsT=indexsT+max(indexk-(indp(numT)*ones(1,len)),0);
end
%montedexput:simulated AEX put option price
%from Monte-Carlo simulation.
  montedexput=(indexsT/m)*exp(-indexr*tau);
  priceerro=abs(montedexput-bsdexput);
for j3=1:len
x0=indexvol(j3);
volatility(j3)=fzero(@(v)impliedvol(v,indexs0,indexk,tau,...
indexr,j3,montedexput(j3)),x0,...
optimset('TolX',1e-7,'TolFun',1e-7));
end
function y=impliedvol(v,B,k,T,r1,N3,p0)
 d1=(log(B/k(N3))+(r1+v^2/2)*T)/(v*sqrt(T));
 d2=d1-v*sqrt(T);
 put=k(N3)*exp(-r1*T)*normcdf(-d2,0,1)-B*normcdf(-d1,0,1);
 y=put-p0;
end
volerro=abs(volatility-indexvol);
%figure(1):put option price plot
figure(1)
plot(indexk,montedexput,'--b'),xlabel('strike'),...
ylabel('put price');hold on
plot(indexk,bsdexput,'-.r*');xlabel('strike'),...
ylabel('put price');grid on
legend('monte put','market put','Location','NorthWest');
%figure(2):skew plot
figure(2)
plot(indexk,volatility,'--b'),xlabel('strike'),...
ylabel('implied volatility');hold on
plot(indexk,indexvol,'-.r*');xlabel('strike'),...
```

```
ylabel('implied volatility');grid on
legend('monte vol','market vol','Location','NorthEast')
toc;
time=toc;
end
```

5. Simulate AEX with stochastic correlation model (Correlation Model 5).

```
function [bsdexput,montedexput,indexvol,volatility,...
            volerro,eigvalues2]=threestocks_stccor_1()
format long
t0=datestr('10/10/2011','dd/mmm/yyyy');  %starting date
T=datestr('12/16/2011','dd/mmm/yyyy');    %maturity date
num=datenum(T)-datenum(t0);%number of days
numT=num-2*floor(num/7);
tau=numT/252;
dt=1/252;

r=0.013088666*ones(3,1); %risk free rates
indexr=0.013088666;
%AEXFDIV=1.513154636;
indexs0=293.5-1.513154636*exp(-indexr*tau);
%index strike prices
indexk=[357.4372334,343.752007,333.2139058,316.6557355,...
295.4245611,268.4832617,237.9895254,216.3888022,187.4924704];
%index market volatilities
indexvol=[21.7344,21.6778,22.1872,24.3946,...
    28.3,33.7902,40.2992,45.28,53.204]/100;
[Call, Put]= blsprice(indexs0,indexk,indexr,...
                            tau, indexvol,0);
bsdexput=Put;
len=length(indexk);
volatility=zeros(1,len);
%INGS_0=5.87; PHIS_0=14.44; RDSAS_0=24.06158
s0=[5.87;14.44;24.06158];
%nING=3830.227;nPHI=986.0788;nRDSA=1748.224;
numShares=[3830.227;986.0788;1748.224;];
divisor=sum(numShares.*s0)/indexs0;
%adjust the number of shares of stocks
newnumshares=numShares/divisor;

INGpar=[0.438495,0.616692,-0.578492,3.589774,...
    3.023439,0.194891,-0.526817,0.604418];
```

```
PHIpar=[0.096358,0.281223,-0.549169,3.530070,...
    1.092620,0.598306,-0.204061,0.106495];
RDSApar=[0.036639,0.047743,-0.510405,1.313757,...
    0.474227,0.435591,-0.178816,0.044928];
par=[INGpar;PHIpar;RDSApar];
v0=par(:,1);theta=par(:,2);rho=par(:,3);kappa=par(:,4);
sigma=par(:,5);lambda=par(:,6);muJ=par(:,7);vJ=par(:,8);

len=length(indexk);
%number of simulation times
m=50000;
%index sumed payoffs at maturity T from 50000 sample paths
indexsT=zeros(1,len);
%set values for the parameters in the model
alpha=0.1;
Beta=0.9;
imprho=0.6;
sigmarho=0.3;
eta1=0.5;
eta2=0.6;
lambda1=0.9;  %lambda_rho
%rho1,rho2,rho3 form the starting correlation
%matrix.
rho_21=0.1;  %correlation between stock 2 and 1
rho_31=0.2;  %correlation between stock 3 and 1
rho_32=rho_21*rho_31+sqrt(1-rho_21^2)*sqrt(1-rho_31^2);

%the vector of the rho_21 at different time step
cor_21=zeros(1,numT);
%the vector of the rho_31 at different time step
cor_31=zeros(1,numT);
%the vector of the rho_32 at different time step
cor_32=zeros(1,numT);
%eigvalues2 is to check positive definiteness...
%of the correlation matrix
eigvalues2=zeros(3,numT);
cor_21(1)=rho_21;
cor_31(1)=rho_31;
cor_32(1)=rho_32;
%rhomat is the correlation matrix
rhomat=zeros(3,3,numT);
rhomat(:,:,1)=[1,rho_21,rho_31;rho_21,1,rho_32;rho_31,rho_32,1];

for j1=1:m
        a=s0;
```

```
        a1=rhomat(:,:,1);
    for j2=1:numT
%nearest_mineig.m is the already developed program
%to guarantee the positive definiteness of the
%correlation matrix
 aa=nearest_mineig(a1,0.001);
%change the above matrix to become a correlation
%matrix with diagonal elements equaling to 1
 bb=aa-diag([aa(1,1),aa(2,2),aa(3,3)])+eye(3);
 eigvalues2(:,j2)=eig(bb);
 rhomat(:,:,j2)=bb;
%rhomat is the correlation matrix

       L = chol(rhomat(:,:,j2),'lower');
       epsilon=normrnd(0,1,3,2);
       z(:,1)=L*epsilon(:,1);
       z(:,2)=epsilon(:,2);
       zJ=normrnd(log(1+muJ)-vJ/2,vJ,3,1);
       J=exp(zJ)-1;
       N=poissrnd(lambda*dt);
       s=a.*exp((r-lambda.*muJ-v0/2)*dt+...
       sqrt(v0).*z(:,1)*sqrt(dt)+J.*N);

 vol=v0+kappa.*(theta-v0)*dt+sigma.*sqrt(v0).*...
 (rho.*z(:,1)+sqrt(1-rho.^2).*z(:,2))*sqrt(dt);
       a=real(s);
       v0=max(vol,0);

    %new add for correlation
    Z=normrnd(0,1,1,2);
    dW1=sqrt(dt)*Z(1);
    dW2=(cor_21(j2)*Z(1)+sqrt(1-cor_21(j2)^2)*Z(2))*sqrt(dt);
 %Z(1) is for brownian motion w1,
 %Z(2) is for the common brownian motion.
  dW3=(cor_31(j2)*Z(1)+sqrt(1-cor_31(j2)^2)*Z(2))*sqrt(dt);
  cor_21(j2+1)=alpha*(imprho-cor_21(j2))*dt-sigmarho*...
 (eta1*dW1+eta2*dW2)+(Beta-cor_21(j2))*poissrnd(lambda1*dt);
  cor_31(j2+1)=alpha*(imprho-cor_31(j2))*dt-sigmarho*...
 (eta1*dW1+eta2*dW3)+(Beta-cor_31(j2))*poissrnd(lambda1*dt);
  cor_32(j2+1)=cor_21(j2)*cor_31(j2)+sqrt(1-cor_21(j2)^2)*...
                                     sqrt(1-cor_31(j2)^2);
  rhomat(:,:,j2+1)=[1,cor_21(j2+1),cor_31(j2+1);...
      cor_21(j2+1),1,cor_32(j2+1);cor_31(j2+1),cor_32(j2+1),1];
   %finish new add for correlation
    end
```

```
  %indp is the index stock price at maturity T
      indp=sum(newnumshares.*a);
      indexsT=indexsT+max(indexk-indp*ones(1,len),0);
 end
montedexput=(indexsT/m)*exp(-indexr*tau);
%PUT option price of the synthetic AEX from Monte-Carlo
%simulation
    for j3=1:len
volatility(j3)=fzero(@(v)impliedvol(v,indexs0,indexk,...
tau,indexr,j3,montedexput),0.3,...
optimset('TolX',1e-7,'TolFun',1e-7));
%the above 3 lines is for getting the implied volatility
    end
function y=impliedvol(v,B,k,T,r1,N2,p0)
        d1=(log(B/k(N2))+(r1+v^2/2)*T)/(v*sqrt(T));
        d2=d1-v*sqrt(T);
        put=k(N2)*exp(-r1*T)*normcdf(-d2,0,1)-B*normcdf(-d1,0,1);
        y=put-p0(N2);
end
    volerro=abs(volatility-indexvol);
    figure(1)
    plot(indexk,montedexput,'--b');
    xlabel('strike'),ylabel('put price');
    hold on
    plot(indexk,bsdexput,'-.r*');
    xlabel('strike'),ylabel('put price');
    grid on
legend('monte put','market put','Location','NorthWest');

    figure(2)
    plot(indexk,volatility,'--b');
    xlabel('strike'),ylabel('implied volatility');
    hold on
    plot(indexk,indexvol,'-.r*');
    xlabel('strike'),ylabel('implied volatility');
    grid on
    legend('monte vol','market vol')
 end
```

6. Fitting GARCH(1,1) for individual stocks (DCC model step 1).

```
function [H,var,X,Y,par,output]=mygarch()
format long
data=xlsread('select3stocks.xlsx',2,'B2:D249');
```

```
data1=data(1:150,:);
%data1 for calculating the starting volatility...
%\sigma_{t-1} in GARCH(1,1)model.
[n1,n2]=size(data1);
data2=data(151:200,:);
%the 150th number is from 02/08/2011,...
%sigma_t starting from 152th number 02/08/2011,...
%size of data2 is 49
[L1, L2]=size(data2);
C0=[0.0003,0.0003,0.0001];
%C0 is the C in GARCH(1,1)
A0=[0.4,0.001,0.002];
%A0 is the alpha
B0=[0.3,0.03,0.02];
%B0 is the beta
matr=[C0,A0,B0];
par0=reshape(matr,1,9);
X=zeros(L1,L2);
%X is the residuals
var=zeros(L1,L2);
var(1,:)=(std(data1)).^2;
m=mean(data2);
X(1,:)=data2(1,:)-m;
%X(1,:) is x_{t-1}, our sigma_t start from data(152,:)
Y(1,:)=data2(1,:)-mean(data1);
%Y is the conditional mean
s=0;
%initial value for the likelihood
H=zeros(L1,L2);
%H: the volatility
H(1,:)=(std(data1)).^2;
%H(1,:):starting volatility by taking...
%the standard deviation of data 1
a1=-1*eye(6,9);
a2=[0,0,0,-1,0,0,-1,0,0;0,0,0,0,-1,0,0,-1,0;0,0,0,0,0,-1,0,0,-1];
b=ones(9,1);
A1=[a1;a2];
lb=zeros(1,9);
ub=ones(1,9);
options =optimset('Display','on','MaxIter',5000000,'TolX',...
1e-10,'MaxFunEvals',10000000,'Algorithm','interior-point');
%[par,output]=fminsearchbnd(@maxlikelyhood,par0,lb,ub,options);
[par,output]=fmincon(@maxlikelyhood,par0,A1,b,[],[],...
                                   lb,ub,[],options);
 function y=maxlikelyhood(par)
```

```
    C=par(1:3); A=par(4:6); B=par(7:9);
for i=2:L1
    for j=1:L2
    H(i,j)=C(j)+A(j)*X(i-1,j)^2+B(j)*H(i-1,j);
    end
    Y(i,:)=data2(i,:)-mean(data(1:n1+i-1,:));
    X(i,:)=data2(i,:)-m;
    var(i,:)=std(data(1:n1+i,:)).^2;
    s=s+3*log(2*pi)+log(H(i,1))+log(H(i,2))+...
    log(H(i,3))+X(i,:).*(H(i,:).^(-1))*X(i,:)';
end
y=s;
%plot(X(:,1),'-r*'),xlabel('day'),...
%ylabel('residuals'),hold on;
%plot(Y(:,1),'-b*');
%set(gca,'YLim',[-0.1 0.1])
%set(gca,'YTick',[-0.1:0.05:0.1])
%legend('unconditional case',...
%'conditional case','location','NorthEast')
end
end
```

7. DCC model step 2. Remark: In the following code, Q_t is the V_t and $\overline{Q}$ is the $\overline{V}$ in the thesis.

```
function [R,realR,Dpar,output]=DCCpart2(H,X)
format long
%[H,var,X,Y,par,output]=mygarch();
L=50; %L=size(H)=size(X);
sig=sqrt(H);
%sig: the sigma in GARCH(1,1)
data=xlsread('select3stocks.xlsx',2,'B2:D249');
data3=data(1:150,:);
data4=data(151:200,:);
epsilon=X./sig;
%epsilon:the standard residuals, referring...
%z in the thesis.
Qbar=zeros(3,3);
%Qbar:the Vbar in the thesis
numT=49;  %number of maturity days
Qbar(1,1)=(1/numT)*(sum(epsilon(1:50,1).*epsilon(1:50,1)));
Qbar(2,2)=(1/numT)*(sum(epsilon(1:50,2).*epsilon(1:50,2)));
Qbar(3,3)=(1/numT)*(sum(epsilon(1:50,3).*epsilon(1:50,3)));
Qbar(1,2)=(1/numT)*(sum(epsilon(1:50,1).*epsilon(1:50,2)));
```

```
Qbar(1,3)=(1/numT)*(sum(epsilon(1:50,1).*epsilon(1:50,3)));
Qbar(2,3)=(1/numT)*(sum(epsilon(1:50,2).*epsilon(1:50,3)));
Qbar=[Qbar(1,1),Qbar(1,2),Qbar(1,3);...
      Qbar(1,2),Qbar(2,2),Qbar(2,3);...
      Qbar(1,3),Qbar(2,3),Qbar(3,3)];
Q=zeros(3,3,L);
R=zeros(3,3,L);
Q(:,:,1)=cov(data3);
%Q(:,:,1): V_{t-1} in the thesis
W=diag(1./diag(sqrt(Q(:,:,1))));
R(:,:,1)=W*Q(:,:,1)*W;
%R(:,:,1): R_{t-1} in the thesis
realR(:,:,1)=corr(data(1:151,:));
%realistic correlation
A=[-1,-1;1,1];
b=[0,1];
lb=zeros(1,2);
ub=ones(1,2);
Dpar0=[0.03,0.7];
s=0;
options=optimset('MaxIter',500000,'TolX',1e-10,...
'DiffMinChange',0.00001,'MaxFunEvals',500000,...
'Algorithm','interior-point');
[Dpar,output]=fmincon(@DCC,Dpar0,A,b,[],[],...
lb,ub,[],options);
function y=DCC(Dpar)
alfa=Dpar(1);beta=Dpar(2);
for t=1:49
Q(:,:,t+1)=Qbar*(1-alfa-beta)+...
alfa*(epsilon(t,:)'*epsilon(t,:))+beta*Q(:,:,t);
U=sqrt(diag(Q(:,:,t+1)));
R(:,:,t+1)=diag(1./U)*Q(:,:,t+1)*diag(1./U);
s=s+log(det(R(:,:,t+1)))+epsilon(t+1,:)/(R(:,:,t+1))*...
epsilon(t+1,:)'-epsilon(t+1,:)*epsilon(t+1,:)';
realR(:,:,t+1)=corr(data(1:151+t,:));
end
y=s;
end
r12=reshape(realR(1,2,1:49),[49,1]);
%real correlation between ASML, PHI
R12=reshape(R(1,2,2:50),[49,1]);
%the simulated correlation between ASML, PHI

r13=reshape(realR(1,3,1:49),[49,1]);
%real correlation between ASML, UL
```

```
R13=reshape(R(1,3,2:50),[49,1]);
%the simulated correlation between ASML, UL
r23=reshape(realR(2,3,1:49),[49,1]);
%real correlation between PHI, UL
R23=reshape(R(2,3,2:50),[49,1]);
%real correlation between PHI, UL

figure(1)
plot(r12,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R12,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','NorthEast')
figure(2)
plot(r13,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R13,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','South')
figure(3)
plot(r23,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R23,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','South')
end
```

8. Simulate the correlation matrix for out of sample data after estimating parameters from DCC model with in the sample data.

```
%RUN THIS[H,var,X,Y,par,output]=mygarch()
%take the last row of the H
function [R, realR]=out_sample_R()
data1=xlsread('select3stocks.xlsx',2,'B2:D250');

```

```
data=xlsread('select3stocks.xlsx',2,'B201:D250');
[L1, L2]=size(data);
m=[0.002973366138253  −0.001633168067021  −0.000338188646363];
%the in the sample mean, the same in mygarch.m file
H=zeros(L1,L2); %variance matrix for out of sample data
X=zeros(L1,L2); %residuals for out of sample data
H(1,:)=[0.012407305627240,0.000312796505820,0.000102778453278];
%H(1,:) is the last row from running ...
    [H,var,X,Y,par,output]=mygarch()
X(1,:)=[0.057895122971244,0.045630033664173,0.026250849422132];
%X(1,:) is the last row from running ...
    [H,var,X,Y,par,output]=mygarch()
par=[0.000300122971742,0.000300245945044,0.000100040991970,...
    0.881555391561065,0.001006822647178,0.002032773697887,...
    0.948228510383622,0.039811340588912,0.024913021970721,0.03,0.7];
%par is the optimized parameters from DCC, which is ...
    obtained through
%firstly run [H,var,X,Y,par,output]=mygarch(), then run
%[R,realR,Dpar,Q,Qbar,output]=DCCpart2(H,X) in the command ...
    window.
 C=par(1:3); A=par(4:6); B=par(7:9);alfa=par(10);beta=par(11);
for i=2:L1
    for j=1:L2
    H(i,j)=C(j)+A(j)*X(i−1,j)^2+B(j)*H(i−1,j);
    end
    X(i,:)=data(i,:)−m;
end
sig=sqrt(H); %standard deviation for each stock
epsilon=X./sig;%standized residuals
Qbar=[0.188133559797265   0.440440090043733   ...
    0.567947043750162;
   0.440440090043733   3.298775737862946   3.298037191505298;
   0.567947043750162   3.298037191505298   6.409008803666157];
%Qbar is the Vbar in thesis, which is the unconditional ...
    convariance
%matrix of standardized residuals
Q=zeros(3,3,L1);
%future convariance for standardized residuals
R=zeros(3,3,L1); %future correlation matrix
realR=zeros(3,3,L1);%real correlation matrix from market data
Q(:,:,1)=[0.173647487885433,0.416244434121311,0.535197507438461;
   0.416244434121311,3.247362576662556,3.198833742754284;
   0.535197507438461,3.198833742754284,6.224699493971056];
%Q(:,:,1): take the last value of Q(:,:,50)
%from DCCpart2.
```

```
W=diag(1./diag(sqrt(Q(:,:,1))));
R(:,:,1)=W*Q(:,:,1)*W;
realR(:,:,1)=[1.000000000000000   0.459503338822215   ...
    0.457009495352603;
   0.459503338822215   1.000000000000000   0.554684925594907;
   0.457009495352603   0.554684925594907   1.000000000000000];
for t=1:49
Q(:,:,t+1)=Qbar*(1-alfa-beta)+...
alfa*(epsilon(t,:)'*epsilon(t,:))+beta*Q(:,:,t);
U=sqrt(diag(Q(:,:,t+1)));
R(:,:,t+1)=diag(1./U)*Q(:,:,t+1)*diag(1./U);
realR(:,:,t+1)=corr(data1(1:200+t,:));
end
r12=reshape(realR(1,2,1:49),[49,1]);
%real correlation between ASML, PHI
R12=reshape(R(1,2,2:50),[49,1]);
%the simulated correlation between ASML, PHI

r13=reshape(realR(1,3,1:49),[49,1]);
%real correlation between ASML, UL
R13=reshape(R(1,3,2:50),[49,1]);
%the simulated correlation between ASML, UL
r23=reshape(realR(2,3,1:49),[49,1]);
%real correlation between PHI, UL
R23=reshape(R(2,3,2:50),[49,1]);
%real correlation between PHI, UL
figure(1)
plot(r12,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R12,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','NorthEast')
figure(2)
plot(r13,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R13,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','South')
figure(3)
```

```
plot(r23,'-.r*'),xlabel('day'),
ylabel('correlation'),hold on;
plot(R23,'-.b*'),xlabel('day'),
ylabel('correlation')
set(gca,'YLim',[0.2 0.8])
set(gca,'YTick',[0.2:0.1:0.8])
legend('real correlation',...
'simulated correlation','location','South')
end
```