EMLYON business school

EDUCATING ENTREPRENEURS FOR THE WORLD

MSc MANAGEMENT / Double Degree

Master thesis

ASSET PRICE JUMPS AND THEIR IMPACT ON OPTION PRICING

Oxana USHAKOVA

Supervisor

Olivier LE COURTOIS

Lyon, France, 2014

ABSTRACT

In this Master's thesis we price options using Fast Fourier transform and Monte-Carlo simulation. Our goal is to see the impact of the jump component in the asset price process on accuracy of pricing. First we introduce diffusion, jump-diffusion and pure jump models concerned. Then, using data, calibrated on S&P 500 index, we price European options via FFT. At last, we use Monte-Carlo simulation to price European and Barrier options.

CONTENTS

Introduction 4
Jump Component Framework 5
  Black-Scholes and Merton Model 5
  Levy Processes 8
Option Pricing 18
  Fast Fourier Transform 18
  Monte Carlo Simulation 24
    Jump–Diffusion Models 25
    Pure Jump Models 26
Conclusion 31
References 32
Appendix 33

INTRODUCTION

Louis Bachelier's thesis "Théorie de la spéculation"(1900) is the foundation stone of the modern financial mathematics, thus he is considered as a pioneer in the study of option pricing and stochastic processes. Bachelier suggested to model the asset price using Brownian motion:

$$S(t) = S(0) + \sigma W(t),$$

where $\sigma > 0$ and $W(t)$ is a Brownian motion.

However, this model, being additive, can give negative values as output. Then, in 1965 the first multiplicative model was derived: Samuelson proposed Geometric Brownian motion for stock returns. Eight years later Black and Scholes (1973) and Merton (1973) showed how to price option with the Bachelier-Samuelson model:

$$S(t) = S(0)e^{\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W(t)}$$

In their seminal paper Black and Scholes give a theoretically consistent framework for option pricing. This paper initiated a strong growth of derivatives markets, i.e. the Chicago Board Option Exchange, first exchange to list options, was opened the same year. The Bachelier-Samuelson model is therefore called Black-Scholes model.

The Black-Scholes model has several drawbacks: first, the model assumes that log-increments of the stock price are normally distributed, however the empirical evidence shows that these increments are not normal, secondly, the Black-Scholes model is derived under the assumption that the underlying asset's price process is continuous, i.e. there are no price jumps, so to improve the Black-Scholes model's performance Lévy models were proposed in late 1980s.

In this paper we'll show that Lévy models fit better empirically observed properties of real world data. We'll cover

- diffusion Black-Scholes model & stochastic volatility (SV) Heston model,
- jump-diffusion Merton & Bates model,
- pure jump models based on NIG, CGMY (VG) and Meixner processes.

JUMP COMPONENT FRAMEWORK

BLACK-SCHOLES AND MERTON MODEL

The Black-Scholes model is one of the most important concepts in modern option pricing theory. Before introducing the underlying idea, we list the model's assumptions.

1. The stock price follows the Geometric Brownian motion with μ and σ constant.
2. The short selling of securities with full use of proceeds is permitted.
3. No transaction costs or taxes. All securities are perfectly divisible.
4. There are no dividends during the life of derivative.
5. No arbitrage.
6. Security trading is continuous.
7. Risk-free rate is constant and the same of all maturities.

The (Geometric) Brownian motion is the dynamic form of Normal distribution. The Normal distribution is symmetric around its mean, with zero skewness and kurtosis equal to 3.

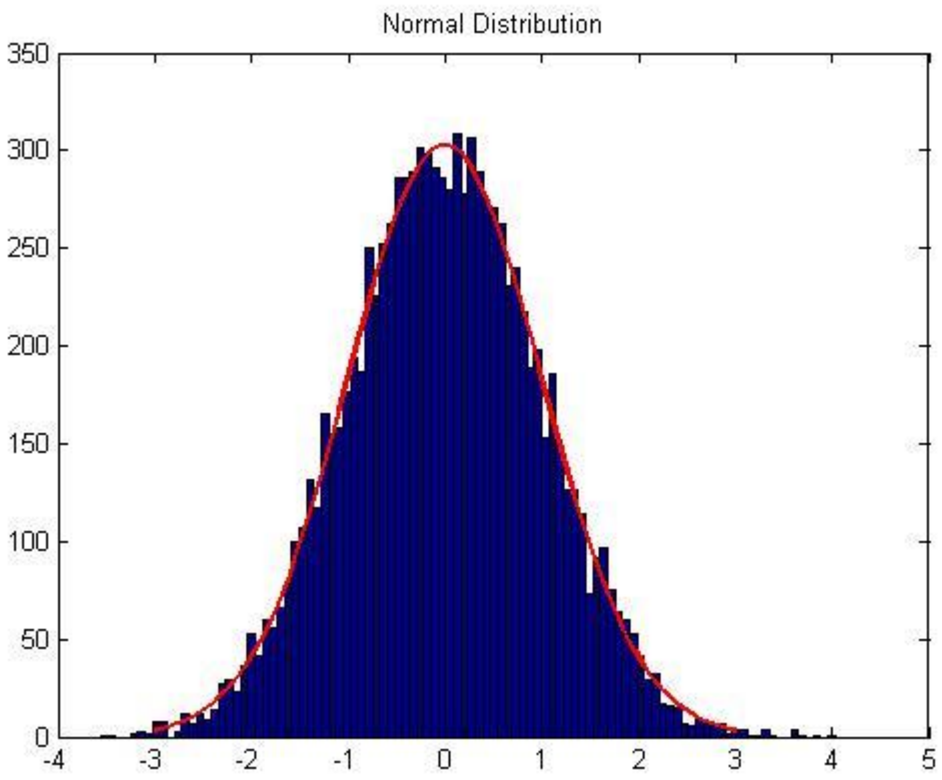

Figure 1. Simulation of Normal Distribution (μ, σ²)

However, empirical market data usually doesn't respect Normal Distribution properties.

Following figures show the imperfection of Black-Scholes model in fitting market data. We observe S&P 500 close prices from 3 Jan, 2005 to 7 Dec, 2011 (1748 ticks).

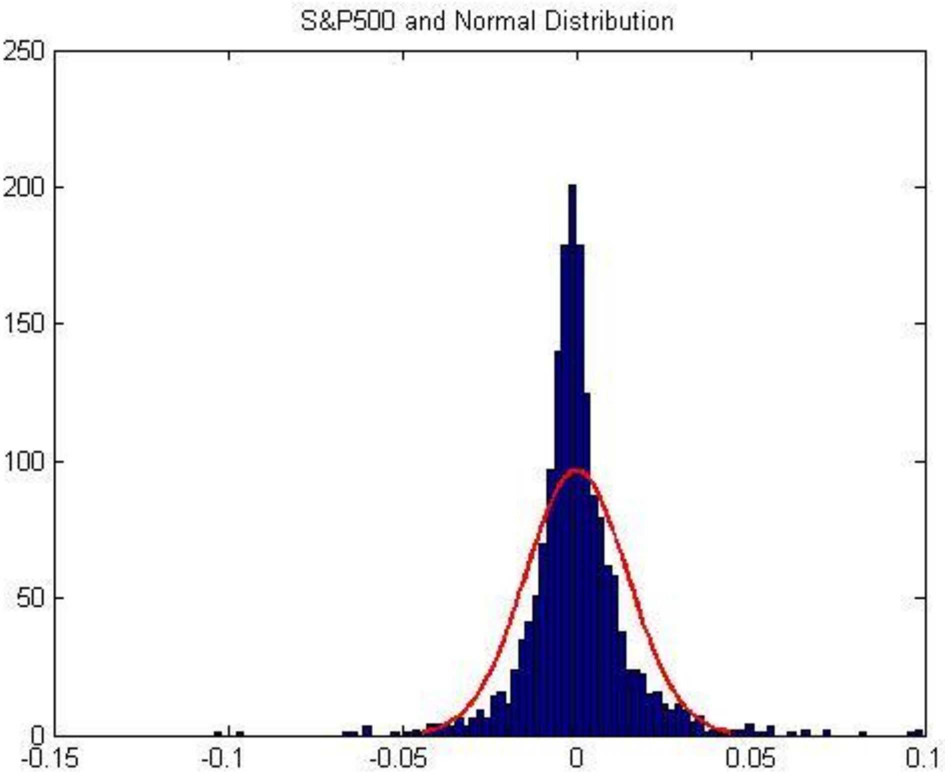

Figure 2. Market data S&P 500 and its fitting to Normal Distribution (μ, σ²)

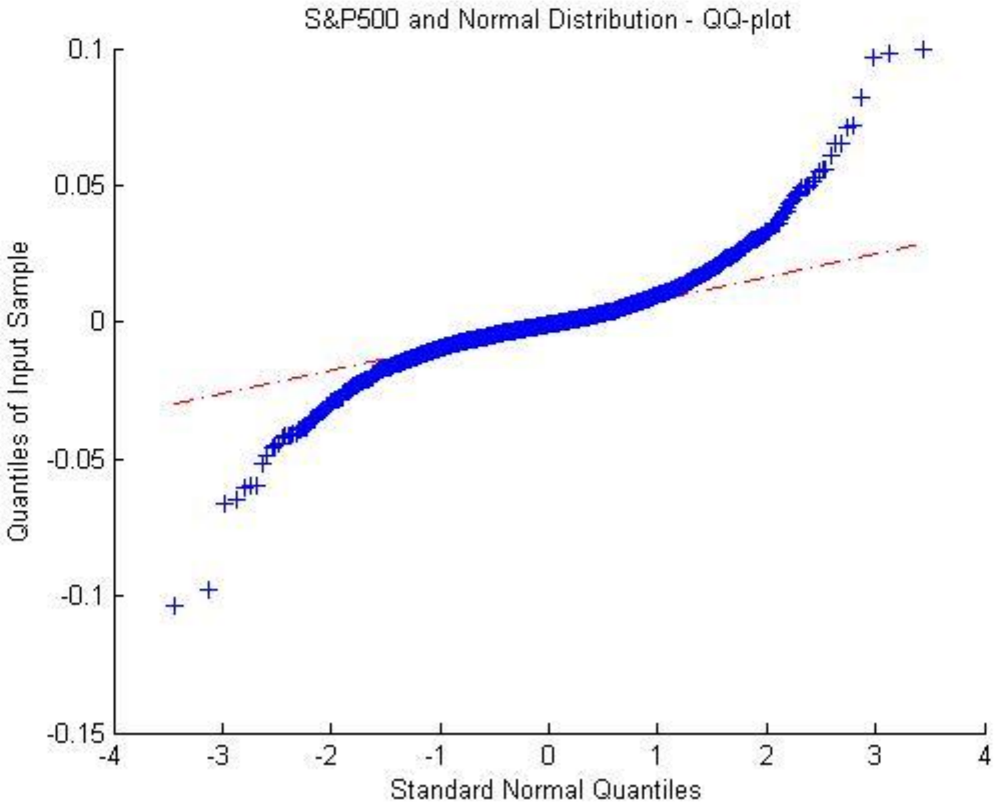

Figure 3. Market data S&P 500 fitting to Normal Distribution (μ, σ²) on QQ-plot

Here the kurtosis is 12.0293 and skewness is 0.5281. The higher kurtosis causes heavy tails, which means a higher chance of large price changes, i.e. jumps. The negative skewness shows the asymmetry.

According to Schoutens W. [15], another imperfection of Black-Scholes model is the volatility clustering: parameters of uncertainty change stochastically over time.

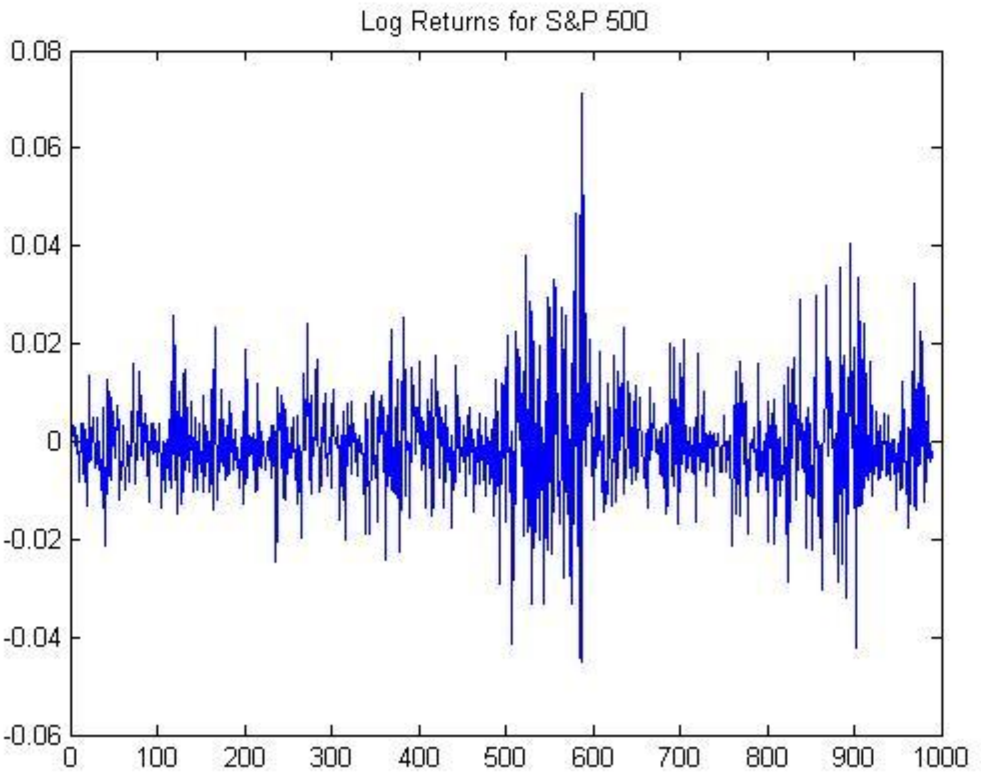

Figure 4. Log-returns on S&P 500

The best way to demonstrate the volatility clustering is plotting autocorrelations of squared log-returns. This method helps see if the heteroscedasticity took place. Heteroscedasticity is a violation of the constant error variance assumption. It occurs if different observations' errors have different variances. Obviously, in case of financial market series we take volatility as the source of error.

Figure 5 shows high autocorrelations, i.e. our data is heteroscedastic, thus we have volatility clustering.

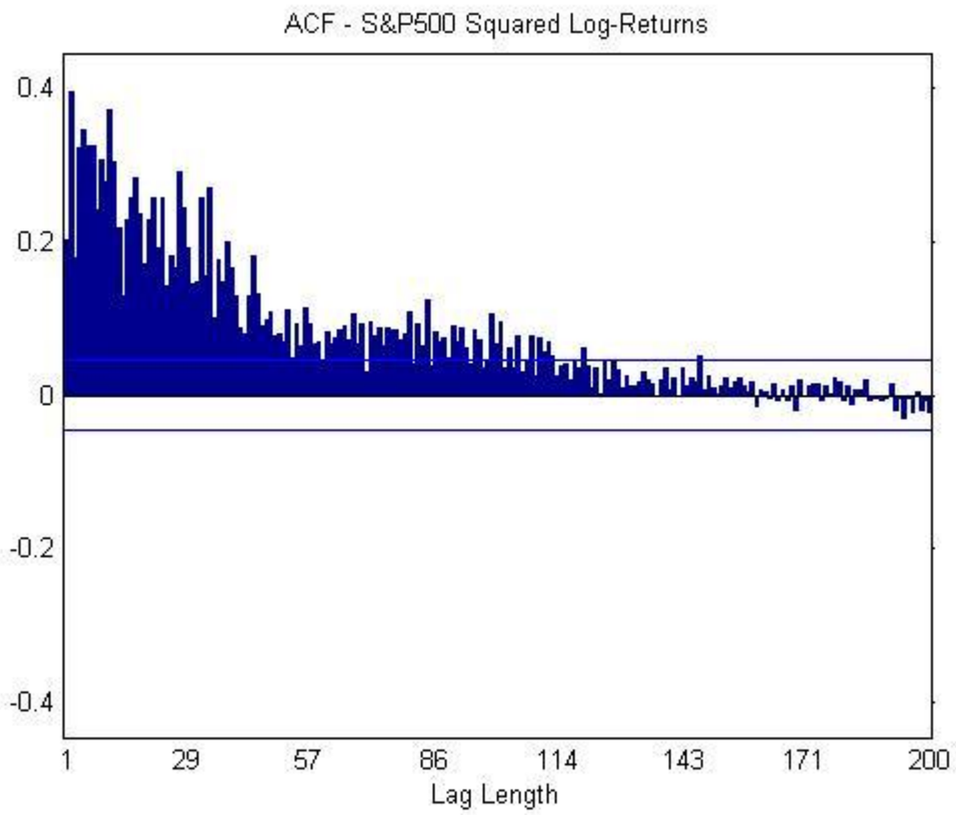

Figure 5. ACF of squared log-returns on S&P 500

Due to excess of kurtosis & non-zero skewness observed and continuity, that eliminates jumps, we may conclude that Normal distribution doesn't fit empirical data, thus Black-Scholes model can't be considered as accurate pricing model.

LEVY PROCESSES

Here we introduce processes and models, driven by these processes. The processes are classified by model type they generate:

- A process where stock prices change continuously forms as a diffusion model.
- A process where continuous changes are overlaid with jumps forms as jump-diffusion model, JD models.
- A process where all stock price changes are jumps forms as a pure jump model, PJ models.

DIFFUSION PROCESSES

WIENER PROCESS

Brownian motion, named after botanist Robert Brown, was first used in physics to describe the random motion of particles. Its application in mathematics, introduced by Norbert Wiener, gave rise to the study of continuous time martingales.

A stochastic process X_t satisfying the following stochastic differential equation:

$$X_t = \mu t + \sigma W_t$$

is called a Wiener process with drift μ and infinitesimal variance σ^2.

Paths of Wiener process, or Brownian Motion, are continuous and its increments are normally distributed with mean 0 and variance t-s, i.e. for all $0<s<t$ we have

$$W_t - W_s \sim N\,(0, t - s)$$

Because of $W_0 = 0$, it follows immediately that

$$W_t \sim N\,(0, t)$$

i.e. W_t is normally distributed with mean 0 and variance t.

GEOMETRIC BROWNIAN MOTION

As it was mentioned above that Wiener process is the mathematical interpretation of Brownian motion. Now we introduce Geometric Brownian motion (GBM), used by Black-Scholes-Merton in option pricing model.

Instead of assuming that stock price follows Brownian motion, Black-Sholes-Merton model uses Geometric Brownian motion for stock price, where the logarithm of stock price follows Brownian motion. The reason is that the expected return is independent of the stock's price. So the assumption of constant expected drift rate is should be replaced by the constant expected return, i.e. expected drift divided by the stock price.

Geometric Brownian motion is a continuous-time stochastic process where the logarithm of the randomly varying quantity follows a Brownian motion with drift, satisfying the following stochastic differential equation

$$dS_t = \mu S_t dt + \sigma S_t dW_t$$

where W_t is a Wiener process and μ and σ are constants.

WHY GEOMETRIC BROWNIAN MOTION

- The expected returns of GBM are independent of the value of the process (stock price), which agrees with what we would expect in reality
- A GBM process only assumes positive values, just like real stock prices.
- A GBM process shows the same kind of 'roughness' in its paths as we see in real stock prices.
- Calculations with GBM processes are relatively easy.

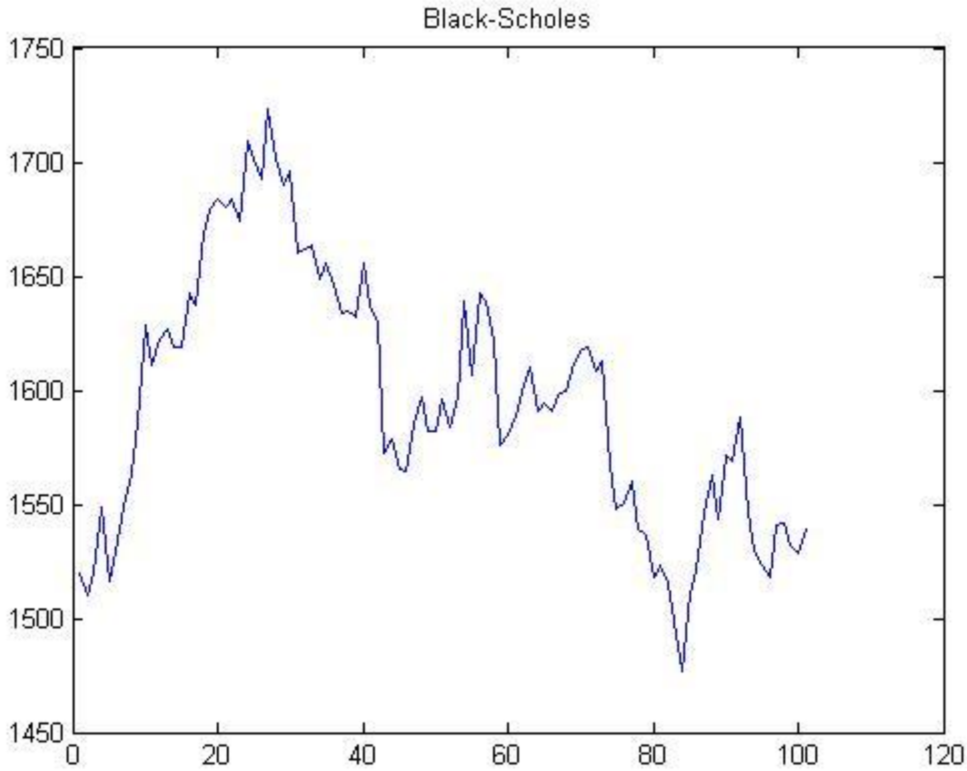

Figure 6. Asset path simulation in Black-Scholes model

POISSON PROCESS

The trajectories of a Poisson process are piecewise constant and right-continuous and left-limited, RCLL, with jumps of constant size S. The jumps occur at times T_i and the intervals between jumps (the waiting times) are exponentially distributed. At every date $t > 0$, N_t has the Poisson distribution with parameter λ_t, where $\lambda \geq 0$ if for $k \in N_0$ that is, it is integer-valued and

$$p(k) = P(X = k) = e^{-\lambda}\frac{\lambda^k}{k!}$$

The Poisson process shares with the Brownian motion the very important property of independence and stationarity of increments, that is, for every $t > s$ the increment $N_t - N_s$ is independent from the history of the process up to time s and has the same law as N_{t-s}.

CHARACTERISTIC FUNCTION

$$\mathbb{E}[e^{iuX_t}] = \exp\{\lambda t(e^{iu} - 1)\}$$

COMPOUND POISSON PROCESS

For financial applications, it is of little interest to have a process with a single possible jump size. The compound Poisson process is a generalization where the waiting times between jumps are exponential but the jump sizes can have an arbitrary distribution. More precisely, let N be a Poisson process with parameter λ and $\{Y_i\}_{i\geq 1}$ be a sequence of independent random variables with law f. The process

$$X_t = \sum_{i=1}^{N_t} Y_i$$

is called compound Poisson process. Its trajectories are RCLL and piecewise constant but the jump sizes are now random with law f. The compound Poisson process has independent and stationary increments.

CHARACTERISTIC FUNCTION

$$\mathbb{E}\left[e^{iuX_t}\right] = \exp\{t\lambda \int \left(e^{iux} - 1\right) f(dx)\}$$

HESTON PROCESS

The Heston model is again diffusion model, but in contrast to Black-Scholes model the Heston model considers that volatility follows a stochastic process. This model can be seen as an extension of Black-Scholes model, where second source of randomness is added.

The Heston model is given the following set of SDEs:

$$dS(t) = \mu S(t)dt + \sqrt{V(t)}S(t)dW_1(t)$$

$$dV(t) = \kappa\big(\Theta - V(t)\big)dt + \upsilon\sqrt{V(t)}dW_2(t)$$

$$S(0) = S_0$$

$$V(0) = V_0$$

$$\langle dW_1, dW_2 \rangle = \rho dt,$$

where

κ – mean reversion speed of the variance, Θ - long-term variance, V_0 - spot variance, υ – volatility of the variance, ρ- the correlation between $\langle dW_1, dW_2 \rangle$.

CHARACTERISTIC FUNCTION

$$\mathbb{E}\left[e^{iuX_t}\right] = \exp\{A_H(u,t,T) + B_\sigma(u,t,T)V(t) + iuX(t)\}$$

Functions $A_H(u,t,T)$ and $B_\sigma(u,t,T)$ are given in Appendix with MATLAB syntax.

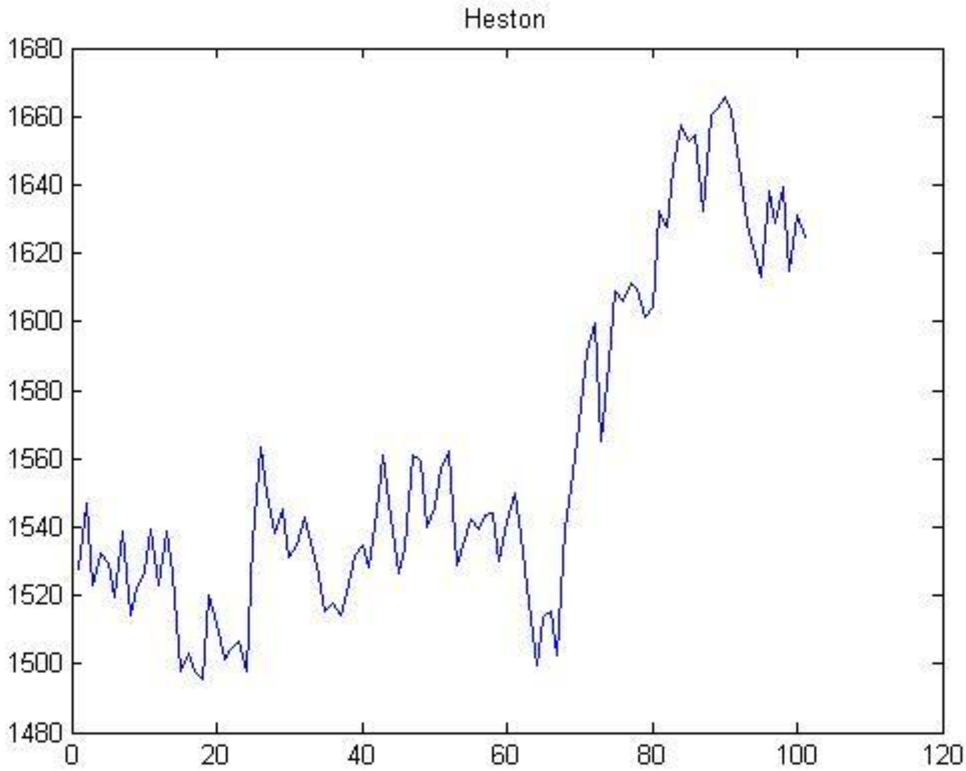

Figure 7. Asset path simulation in Heston model

JUMP-DIFFUSION MODELS

MERTON'S MIXED JUMP-DIFFUSION PROCESS

Combining the Brownian motion with drift and the compound Poisson process, we obtain the simplest form of jump diffusion model based on a process which has sometimes and random sized jumps and has continuous but random evolution between the jumps times:

CHARACTERISTIC FUNCTION

Its characteristic function, can be computed by simple multiplication of corresponding characteristic functions since these two parts are independent:

$$\mathbb{E}\left[e^{iuX_t}\right] = \exp\left\{t\left(i\mu u - \frac{(\sigma u)^2}{2} + \lambda \int_{\mathbb{R}} \left(e^{iux} - 1\right) f(dx)\right)\right\}$$

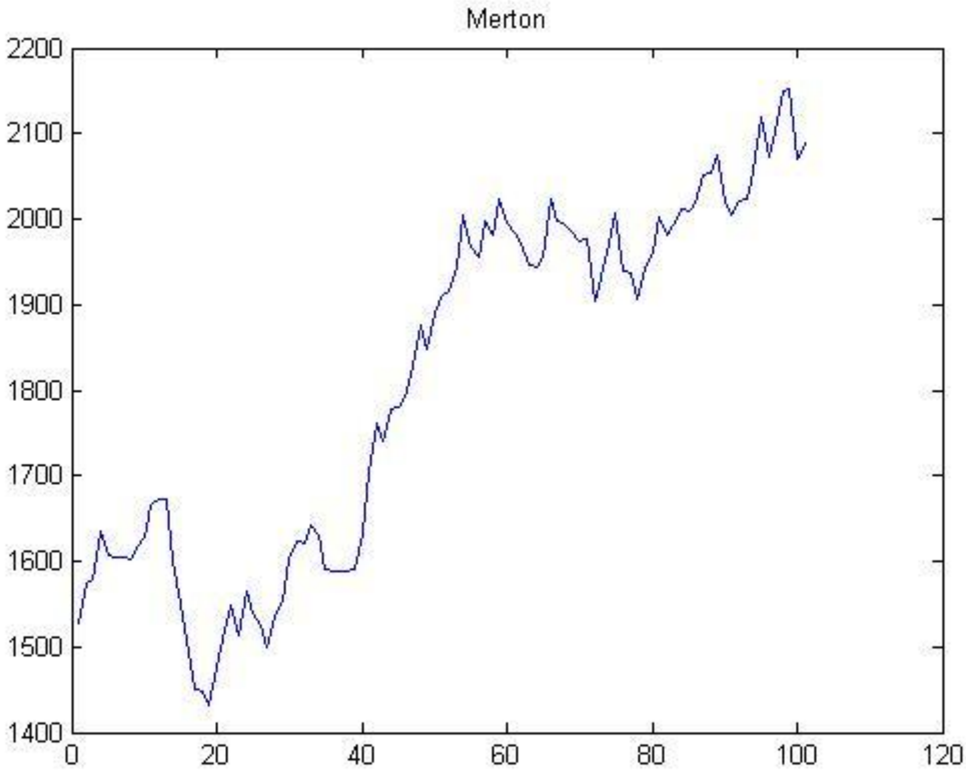

Figure 8. Asset path simulation in Merton model

BATES MODEL

The Bates model combines properties of Heston and Merton models, thus the Bates model consider that asset price process has both stochastic volatility and jumps.

CHARACTERISTIC FUNCTION

Due to the independence of jump and stochastic volatility components we consider that the characteristic function can be obtained by simple multiplication of corresponding characteristic function of Heston and Merton models processes as in Merton model.

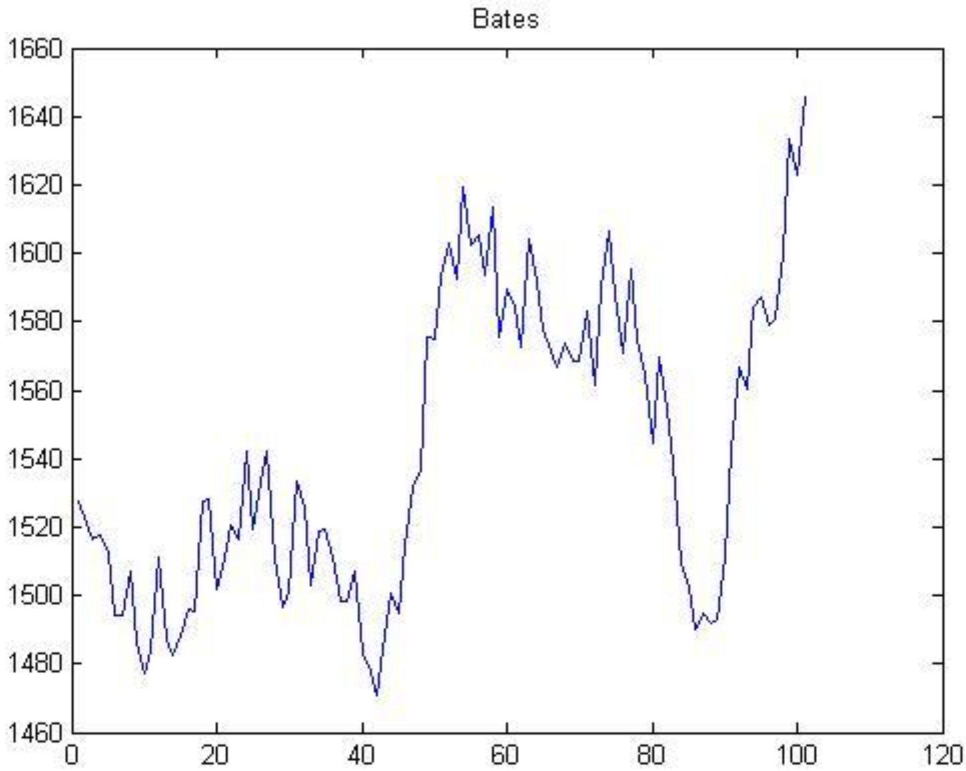

Figure 9. Asset path simulation in Bates model

PURE JUMP MODELS

GENERALIZED HYPERBOLIC PROCESS

The Generalized Hyperbolic process is a wide class of five-parameter processes, which is related to the Generalized Inverse Gaussian distribution as the NIG process does.

Generalized Hyperbolic distributions can be represented as normal mean-variance mixtures

$$d_{GH}(x;\ \lambda,\beta,\delta,\mu) = \int_0^\infty d_{N\,(\mu+\beta_{y,y})}(x) d_{GIG}(y;\ \lambda,\delta,\sqrt{a^2-b^2})dy$$

where the mixing distribution is the Generalized Inverse Gaussian distribution with *p=0.*

CHARACTERISTIC FUNCTION

$$\mathbb{E}\left[e^{iuX_t}\right] = \left(\frac{\alpha^2-\beta^2}{\alpha^2-(\beta+iu)^2}\right)^{\frac{v}{2}} \frac{K_v(\delta\sqrt{\alpha^2(\beta+ui)^2})}{K_v(\delta\sqrt{\alpha^2-\beta^2})}$$

where K_v is the modified third-kind Bessel, or MacDonald, function

$$K_v(z) = \frac{1}{2}\int_0^\infty u^{v-1} exp\left\{-\frac{1}{2}z(u+u^{-1})\right\}du$$

NORMAL INVERSE GAUSSIAN PROCESS (NIG)

The NIG process is a subclass for Generalized Hyperbolic process with no Brownian component. This process is used when heavy tails, skewness and jumps must be taken in account, i.e. commodity markets.

The NIG distribution is the special case of Normal Variance-Mean Mixture distribution,

$$Y = \alpha + \beta V + \sigma\sqrt{V}X$$

where α and β are real numbers and σ > 0. The random variables X and V are independent, X is normal distributed with mean zero and variance one, and V is continuously distributed on the positive half-axis with mixing probability density function g.

The NIG distribution as a normal variance-mean mixtures when the mixture distribution is an inverse Gaussian distribution, special case of Generalized Inverse Gaussian distribution.

It has three parameters – NIG (α, β, δ) where α refers to the tails size, β is the skewness and δ, as σ is Normal distribution, is the volatility. If β = 0 then the NIG distribution is symmetric.

CHARACTERISTIC FUNCTION

$$\mathbb{E}\left[e^{iuX_t}\right] = \exp\left\{-\delta(\sqrt{\alpha^2 - (\beta + iu)^2} - \sqrt{\alpha^2 - \beta^2})\right\} e^{i\mu u}$$

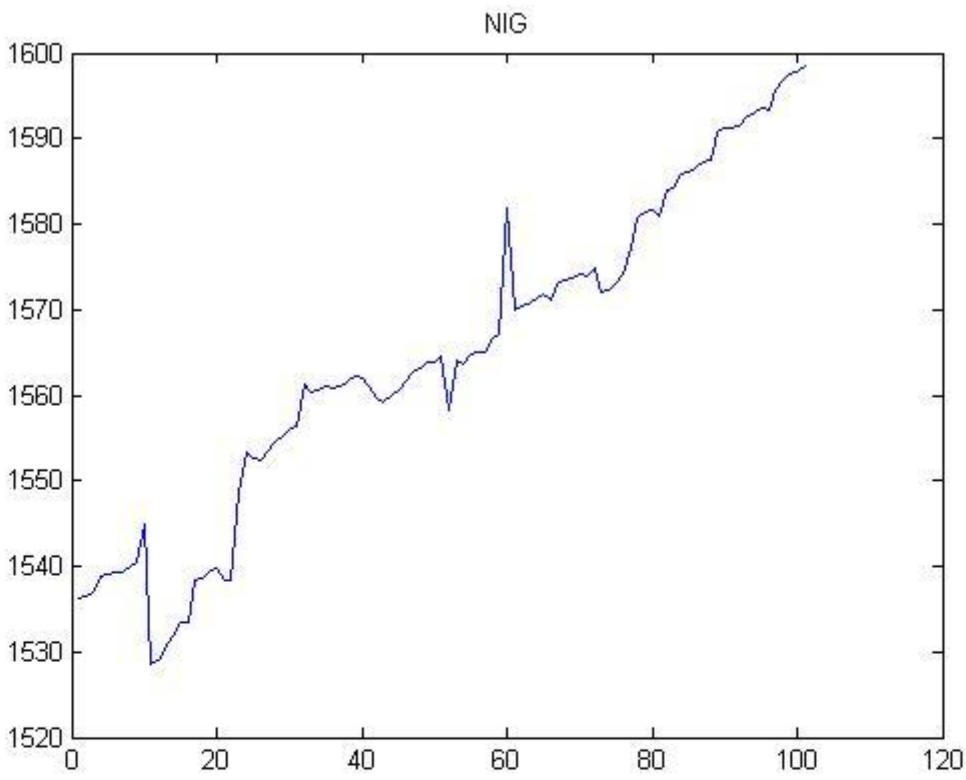

Figure 10. Asset path simulation in NIG model

CGMY PROCESS

The CGMY process is a pure jump process and its path depends on Y:

- If Y < 0, then paths have finite jumps in any finite interval, otherwise, paths have infinitely many jumps in any interval.

The range of parameters is restricted to C, G, M > 0 and 1≤ Y < 2.

The Variance Gamma (VG) process, being the special case of the CGMY process, is a three-parameter process, where Y=0.

CHARACTERISTIC FUNCTION

$$\mathbb{E}\left[e^{iuX_t}\right] = \exp\{Ct\Gamma(-Y)((M-iu)^Y - M^Y + (G+iu)^y - G^Y)\}$$

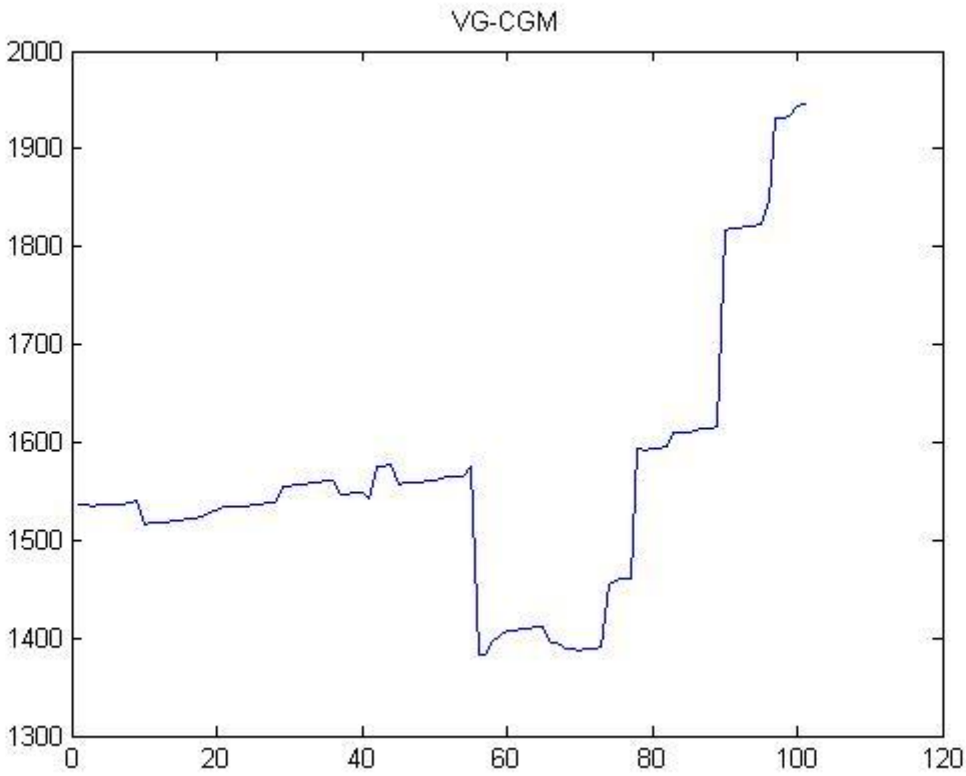

Figure 11. Asset path simulation in VG model

MEIXNER PROCESS

The Meixner process is also a pure jump process, i.e. has no diffusion part. It's based on Meixner distribution and its density is given by:

$$\mathbb{E}\left[e^{iuX_t}\right] = \frac{(2\cos\left(^{\beta}/_{2}\right))^{2\delta}}{2\alpha\pi\Gamma(2d)} \exp(\frac{\beta x}{\alpha}) \left|\Gamma(\delta + \frac{ix}{\alpha})\right|^2 e^{i\mu u}$$

where $\alpha > 0, -\pi < \beta < \pi, \delta > 0$.

CHARACTERISTIC FUNCTION

$$f_{Meixner}(x;\ \alpha,\beta,\delta) = \left(\frac{\cos(^{\beta}/_{2})}{\cosh(\frac{(\alpha u - i\beta)}{2})}\right)^{2\delta}$$

PROPERTIES OF LEVY PROCESSES

All the processes presented refer to Lévy processes - cadlàg continuous-time stochastic processes with stationary independent increments and such that *X(0)* = 0.

Lévy process represents the motion of a point whose successive displacements are random and independent, and statistically identical over different time intervals of the same length.

Any Lévy process is a Markov process, i.e. respects the Markov property where the future value X (t+1) only depends on current position X (t) and not on the past X (t+1) so Brownian motion is a memoryless process;

In other words a Lévy process *X* is a general type of processes such that:

1. $X_0 = 0$
2. X has independent and stationary increments
3. The paths of *X* are RCLL, or cadlàg.

CHARACTERISTIC FUNCTION

An important feature of Lévy process *X* is its infinite divisibility, i.e. for every positive integer t, there exist *t* independent identically distributed random variables $X_{t1}, \ldots, X_{tt}$ whose sum $X_t = X_{t1}, \ldots, X_{tt}$ has the distribution *X*. From this concept it follows that the characteristic function of Lévy process is given by:

$$\mathbb{E}[e^{iuX_t}] = e^{t\psi(u)}$$

Recall that a characteristic function of any real-valued random variable completely defines its probability distribution. If a random variable admits a probability density function, then the characteristic function is the inverse Fourier transform of the probability density function.

Every Lévy process can be presented as:

$$X_t = \gamma t + \sigma B_t + Z_t$$

Where Z_t is a jump process with/without infinitely many jumps.

OPTION PRICING

This chapter covers European call pricing via fast Fourier transform method, or FFT, that was introduced in 1998 by Carr P. and Madan D.H.

FAST FOURIER TRANSFORM

For more general Lévy process models than those based on Brownian motion we typically cannot find analytical solutions in the classic form as in Black Scholes model. We will therefore now introduce a pricing method based on characteristics function – Fourier transform.

This method has the following advantages:

1. The risk neutral density is rarely known, nevertheless we know from the Lévy-Khintchine representation the equation for the Fourier transform of *S(t)*.

2. The algorithms used for the inversion of the Fourier transform are well known, fast and optimized, because of the huge use of Fourier theory in many fields like signal theory.

3. If we use the algorithm named Fast Fourier Transform, then we will be able to valuate options with different strikes in a single calculation.

Here we will evaluate a European call option price based on the price asset price process *S(t),* maturity time T and strike price *K.* Write $k = log(K)$ and $s(T) = log(S(T))$. Let $C_T(k)$ denote the option price and q_T the risk-neutral probability density function of the log price s_T .

The characteristic function of the density q_T is given by

$$\phi_T(u) = \int_{-\infty}^{\infty} e^{ius} q_T(s)ds.$$

The option value which is related to the risk-neutral density q_T is given by

$$C_T(k) = \int_{k}^{\infty} e^{-rT}(e^s - e^k) q_T(s)ds$$

Here $C_T(k)$ is not square integrable because when $k \to -\infty$ so that $K \to 0$, we have $C_T \to S(0)$. To obtain a square integrable function, Carr and Madan suggested consideration of the modified price $c_T(k)$ given by

$$c_T(k) = e^{\alpha k} C_T(k)$$

for a suitable $\alpha > 0$. Here Carr and Madan suggested to choose ≈ 0.25, while Schoutens suggests $\alpha \approx 0.75$. The value of α affects the speed of convergence.

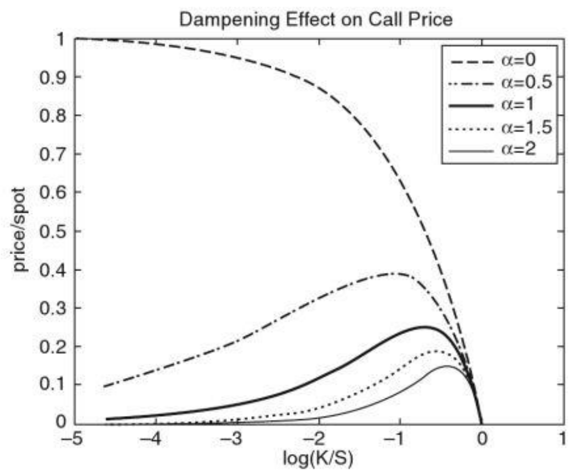

Figure 12. Dampening effect on Call Price [11]

The Fourier transform of $c_T(k)$ is given by

$$\psi_T(v) = \int_{-\infty}^{\infty} e^{ivk}\, c_T(k)dk$$

The inverse corresponding inverse Fourier transform takes the form

$$c_{T(k)} = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{-ivk}\psi_{T(v)}dv$$

Option price formula for $C_T\,(k)$

$$C_T(k) = \frac{e^{(-\alpha k)}}{\pi}\int_0^{\infty} e^{-ivk}\psi_{T(v)}dv$$

In this paper the Fast Fourier transform will be used. FFT is a discrete approximation technique of Fourier transform. This method is not that time consuming and reduces computational labor.

FFT is an efficient algorithm to compute the following sum:

$$w(k) = \sum_{j=1}^{N} e^{(-i2\pi(j-1)(k-1))/N} x(j)$$

Where N must be a power of 2, i.e. we take 2^{14}.

MODEL CALIBRATION

Before implementing FFT to a data set, concerned models should be calibrated to the same data set.

While the pricing problem is concerned with computing option values given the model parameters, the calibration problem is concerned with computing the model parameters given the option prices. Thus the calibration problem is the inverse problem to the pricing problem.

There a lot of calibration methods for nonlinear optimization as Nedler-Mead method, Levenberg – Marquardt method, SQP for non-linear optimization. The calibration process is very labour-intensive and time-consuming, so we use already calibrated parameters as in [3]. The calibration was done on call option prices for S&P 500 index from June 2007 to December 2008. [See Appendix 1 for data used].

	RMSE	ESTIMATED PARAMETERS			
NORMAL	9.8827	sigma			
		0,1531			
CGMY	1.9305	C	G	M	Y
		0,0156	0,0767	7,55	1,2996
NIG	1.6064	alfa	beta	theta	
		5,0364	-3,3199	0,0881	
MEIXNER	1.7084	alfa	beta	theta	
		0,34	-1,49	0,29	

These parameters cover only pure-jump models. In order to see the impact of jumps on jump-diffusion models (with SV Heston model), we consider a *base scenario*. Looking on the graphical illustration of obtained results, we'll see if any of concerned models is beyond the common trend.

BASE PARAMETERS

T	[0.5; 2]
K	(80:10:120)
S	100
r	0.05
d	0.019
sigma	0.15
a (μ_j)	0.1
b (σ_j)	0.2
lambda	0.2
V0	0.02
theta	0.02
omega	0.2
kappa	0.1
rho	0.5

FOURIER TRANSFORM RESULTS

We've price European calls via FFT using MATLAB. Results are presented separately for jump-diffusion and pure jump models.

JUMP –DIFFUSION MODELS

Applying FFT on jump-diffusion models, we see that results are varying w.r.t. maturity chosen. Thus, for maturity T = 0.5 all the models show similar results, however Heston and Merton models are usually more closed to each other than to others.

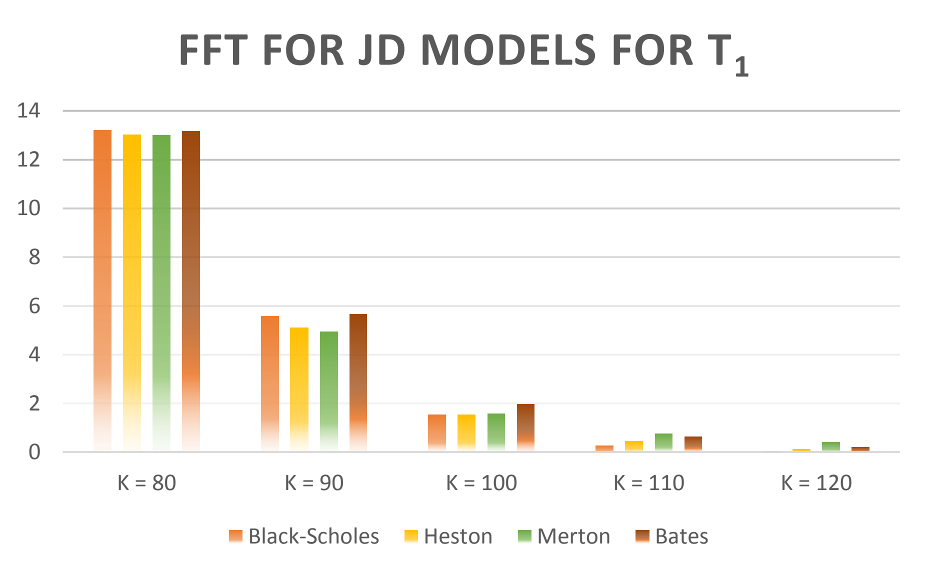

Figure 13. Prices for JD models, computed via FFT for T_1 =0.5

If we consider maturity T=2, then Bates model gives uncommonly high results; at the same time Black-Scholes model shows lowest prices and strong speed of decreasing, while Heston and Merton models are still very close to each other.

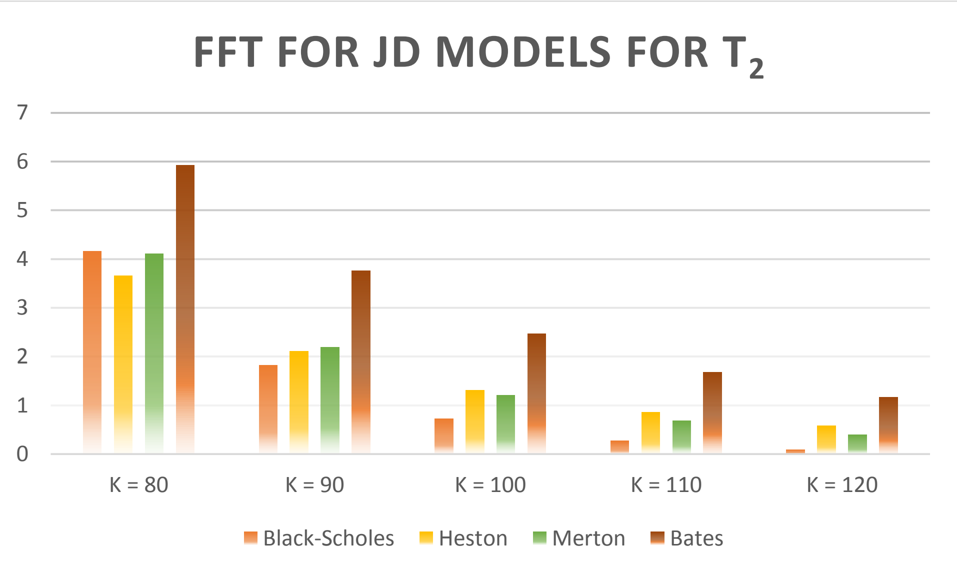

Figure 14. Prices for JD models, computed via FFT for T_2 =2

Simulation of base scenario for jump-diffusion models via FFT couldn't give the definitive answer if any of considered models fits better the market. However, Heston and Merton models set their own trend that was not respected by Black-Scholes model. We can conclude that inserting jumps or/and stochastic component has a strong impact on option pricing.

PURE-JUMP MODELS

Looking at root-mean-square error by process we can already conclude that Lévy processes as NIG, CGMY and Meixner seem to be better in real data fitting than the Black-Scholes model. The reason is that Normal Distribution has only one parameter to calibrate, sigma σ, so the calibration cannot be as accurate as for multy-parametric processes.

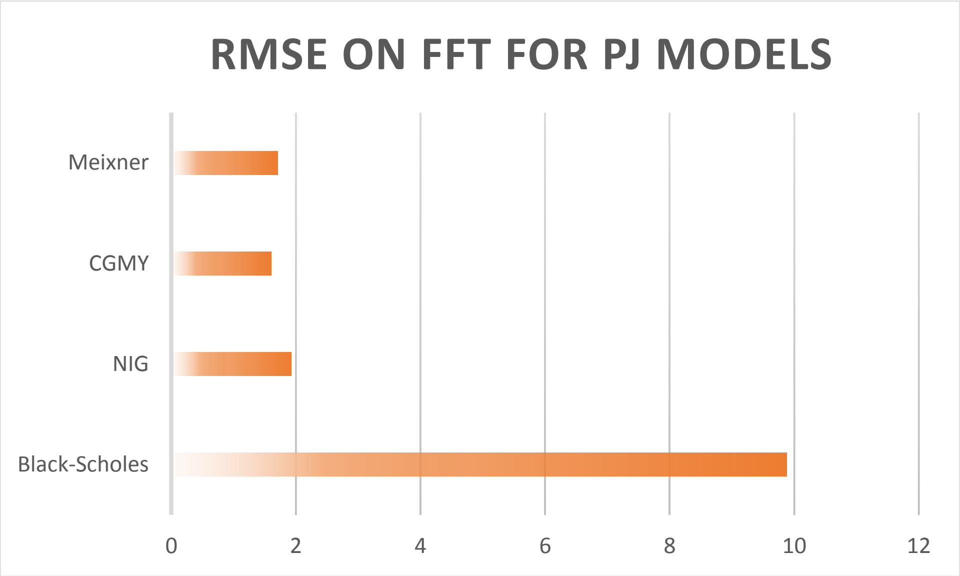

Figure 15. RMSE on FFT for calibrated PJ models

The prices on any process that FFT algorithm returns have a strong common trend for all the maturities. Moreover, Lévy processes as NIG, VG and Meixner fit the market price almost perfectly, while Normal distribution is always a bit different.

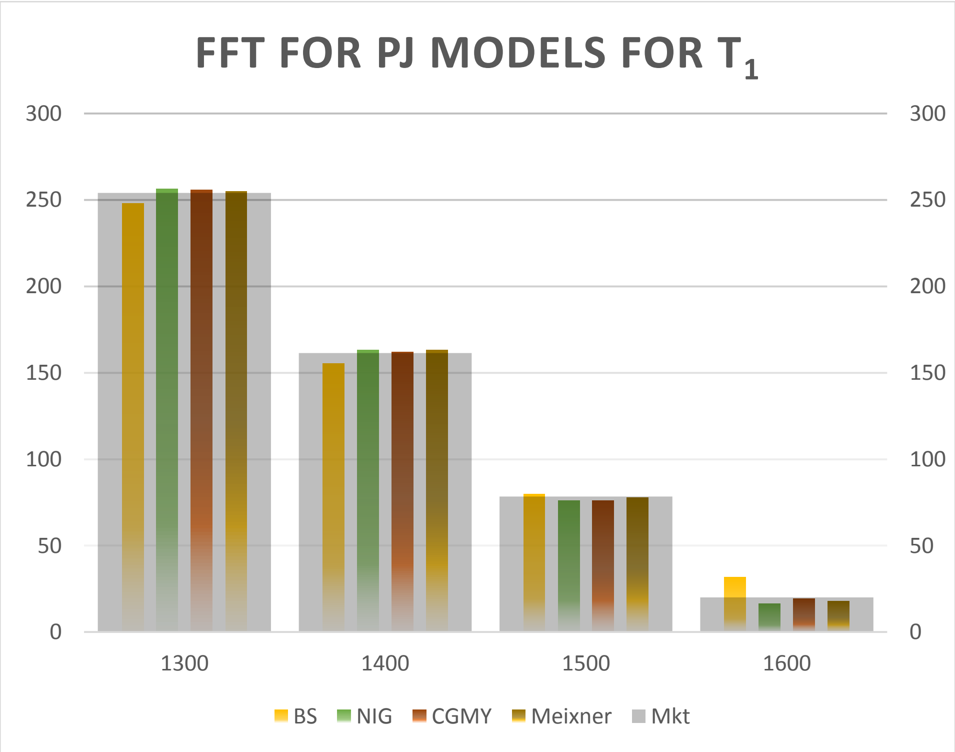

Figure 16. Prices for PJ models, computed via FFT for T_1 = Sept 21, 2007

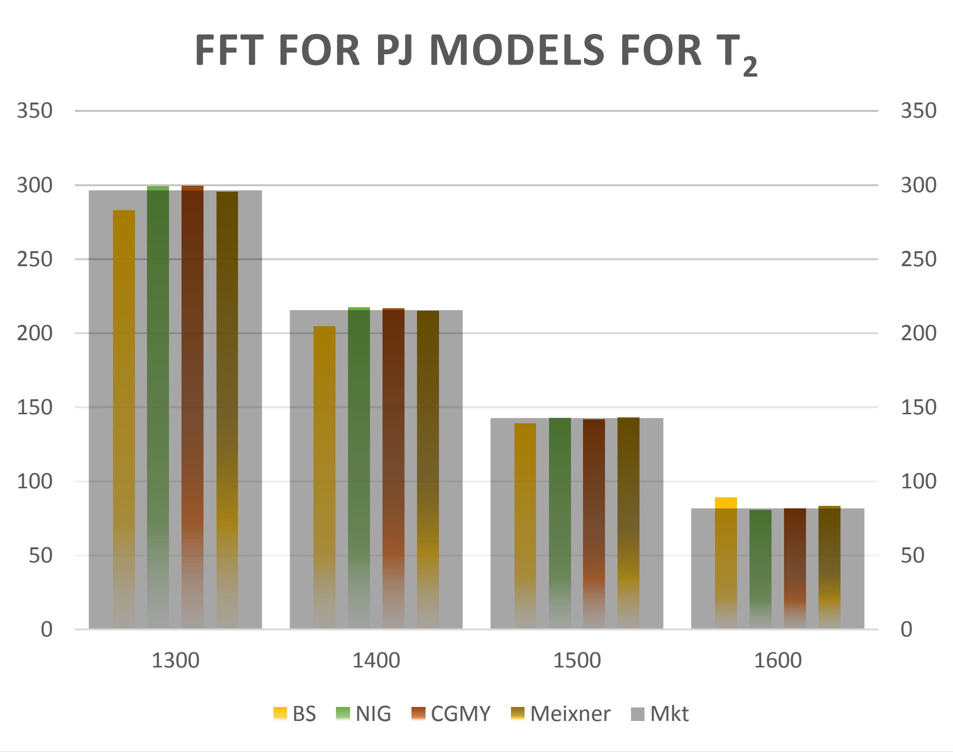

Figure 17. Prices for PJ models, computed via FFT for T_2 = Jun 20, 2008

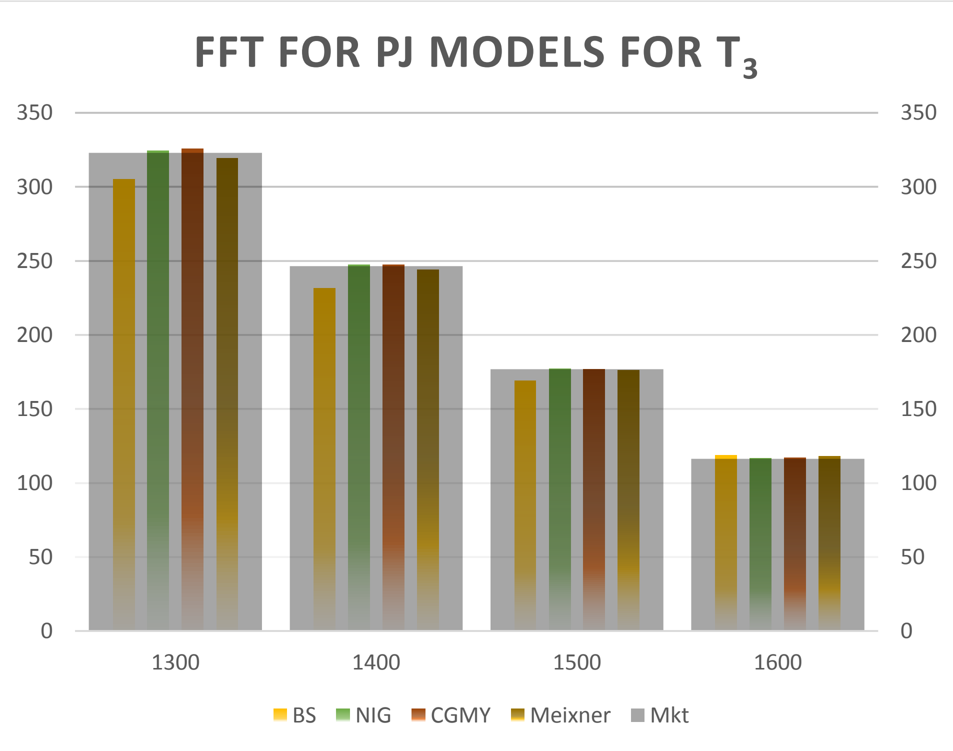

Figure 18. Prices for PJ models, computed via FFT for T_3 = Dec 19, 2008

As we've seen impact of jumps is more visible for pure jump models than jump-diffusion models. Another way to see this impact is to price options with Monte-Carlo simulation.

MONTE-CARLO SIMULATION

The pricing method, that we used earlier - FFT, can be applied to both European and path-dependent options. Pricing exotic options with FFT is extremely time and labour-consuming, thus we use Monte-Carlo algorithm, consisting of several steps:

- Calibrate the model on European option prices,
- Simulate N sample paths of the underlying asset price over the time interval,
- Calculate the payoff of the option for each path, where payoff function depends on option type,
- Average the discounted payoffs over sample paths.

BARRIER OPTIONS

The options discussed in previous chapters are so-called vanilla options, where the payoff only depends on the terminal value, i.e. intermediate values aren't taken into consideration. However, modern OTC derivatives market requires more sophisticated products and exotic options become more popular. In this chapter we will price path-dependent options, i.e. Barrier options, using Monte-Carlo simulation.

Before proceeding to pricing process, we describe barrier options in details.

As it was mentioned above, barrier options are path-dependent. The payoff depends on some level asset price (barrier) being attended. There are 2 categories with 2 types per category of such options:

Category A: *knock-in/knock-out*

A knock-in options is activated only when the underlying asset touches the barrier, while a knock-out option is instead deactivated when it touches the barrier.

Category B: *up/down barrier*

Up/down refers to the location of the barrier w.r.t. asset price.

Thus we have four types of barrier call options:

UP-AND-IN BARRIER CALL

The *up-and-in barrier call option* is a standard European call option with strike K when its maximum lies above the barrier H, while it is worthless otherwise.

$$C_{UI} = e^{-rT}\mathbb{E}^{\mathbb{Q}}[S(T) - K)^{+}1_{M(T)\geq H}$$

UP-AND-OUT BARRIER CALL

The *up-and-out barrier call option* is a standard European call option with strike K when its maximum lies below the barrier H, while it is worthless otherwise.

$$C_{UO} = e^{-rT}\mathbb{E}^{\mathbb{Q}}[S(T) - K)^{+}1_{M(T)<H}$$

DOWN-AND-IN BARRIER CALL

The *down-and-in barrier call option* is a standard European call option with strike K when its minimum lies below the barrier H, while it is worthless otherwise.

$$C_{DI} = e^{-rT}\mathbb{E}^{\mathbb{Q}}[S(T) - K)^{+}1_{m(T)\leq H}$$

DOWN-AND-OUT BARRIER CALL

The down-and-out barrier call option is a standard European call with strike K when its lies above some barrier H, while it is worthless otherwise.

$$lmjC_{DO} = e^{-rT}\mathbb{E}^{\mathbb{Q}}[S(T) - K)^{+}1_{m(T)>H}$$

JUMP –DIFFUSION MODELS

Applying Monte-Carlo simulation on jump-diffusion models, we use following parameters to price European and Barrier options.

In order to price Barrier options we have to find the distribution of max/min values of asset price process *M(T)* and *m(T)*. The explicit formula for *M(T)*, *m(T)* exists if these prices follow Geometric Brownian motion. However, for more general Lévy process we have to use the Monte-Carlo simulation as the distributions are usually not known explicitly.

JD EXTRA-PARAMETERS	
T	0.5
U (BARRIER)	90
K	80

Unfortunately Monte-Carlo simulation on jump-diffusion models for European and Barrier options isn't representative, as the models seem to have equal consistent gaps. However, later we'll look at autocorrelations, where the advantage of applying Heston model is presented.

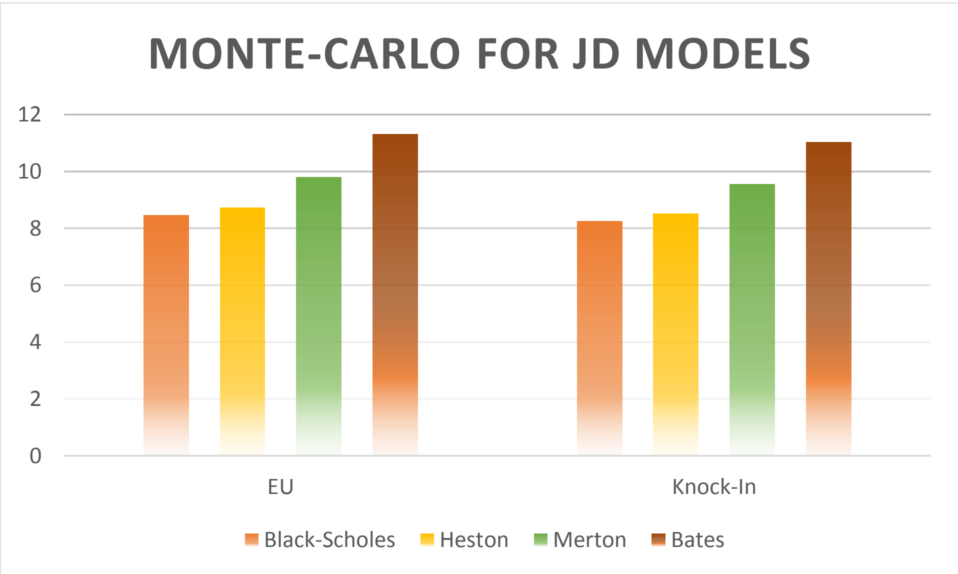

Figure 19. Prices for European calls in JD models, computed via Monte-Carlo algorithm

PURE JUMP MODELS

For pure jump models we've chosen maturity T = Jun 20, 2008 and several strikes K = {1300; 1400; 1500; 1600}. First, we price European options.

The root-mean-square error shows that NIG and VG processes adjust better the market data than Black-Scholes model.

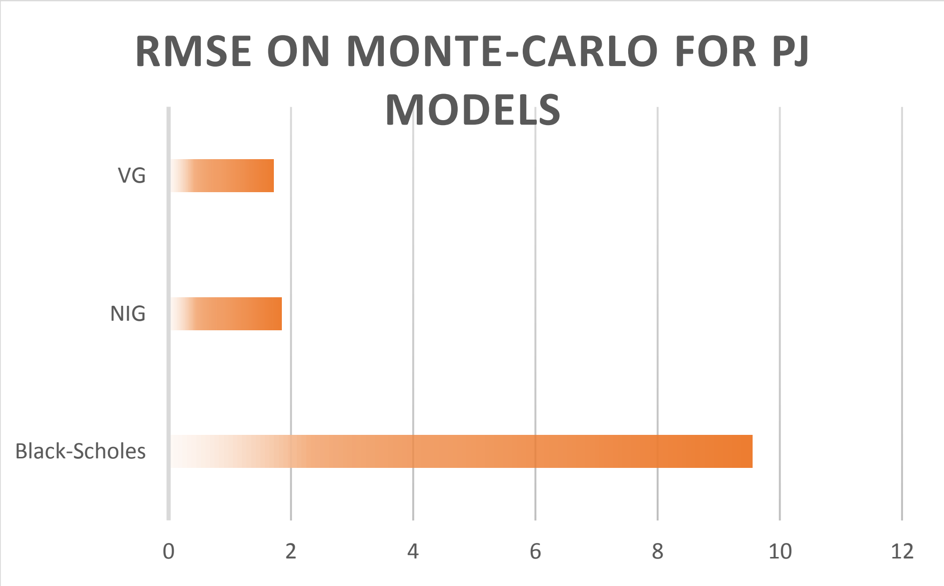

Figure 20. RMSE on Monte-Carlo algorithm for calibrated PJ models

Following figures illustrate the obtained results: as expected, NIG and VG processes adjust better the real market prices.

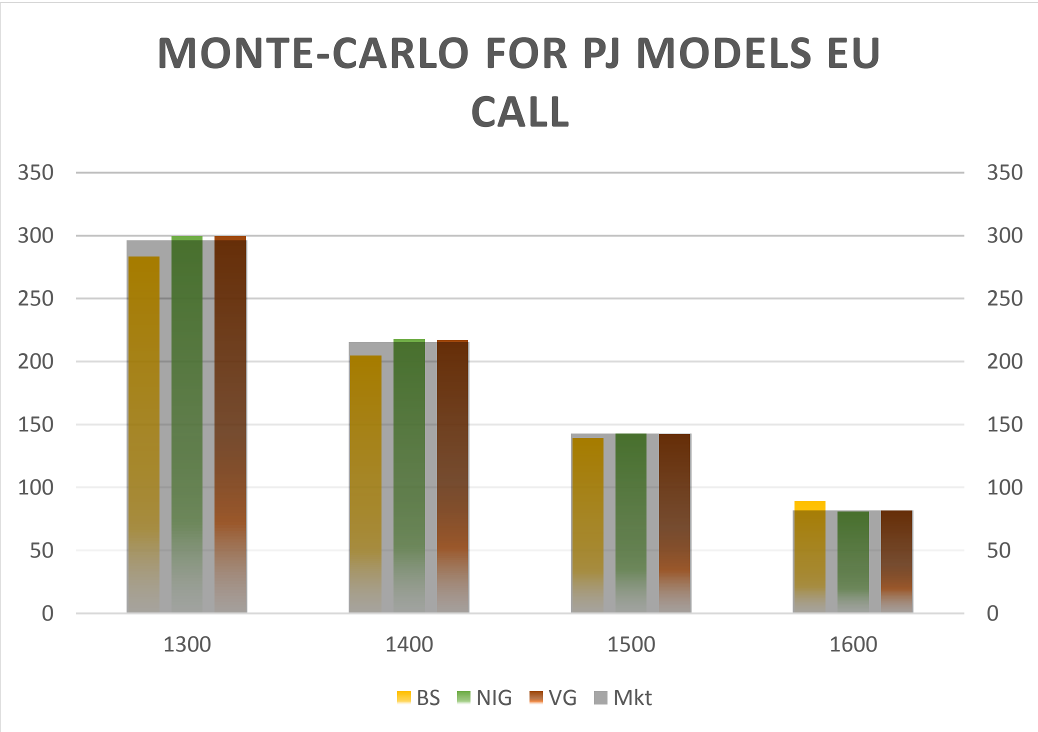

Figure 21. Prices for European calls in PJ models, computed via Monte-Carlo algorithm for Sept 21, 2007

We've chosen 3 barriers for 2 strikes and same maturity. The results vary with barrier's value, but NIG and VG processes are again always very close to each other, that let us think of them as better alternative to Black-Scholes model.

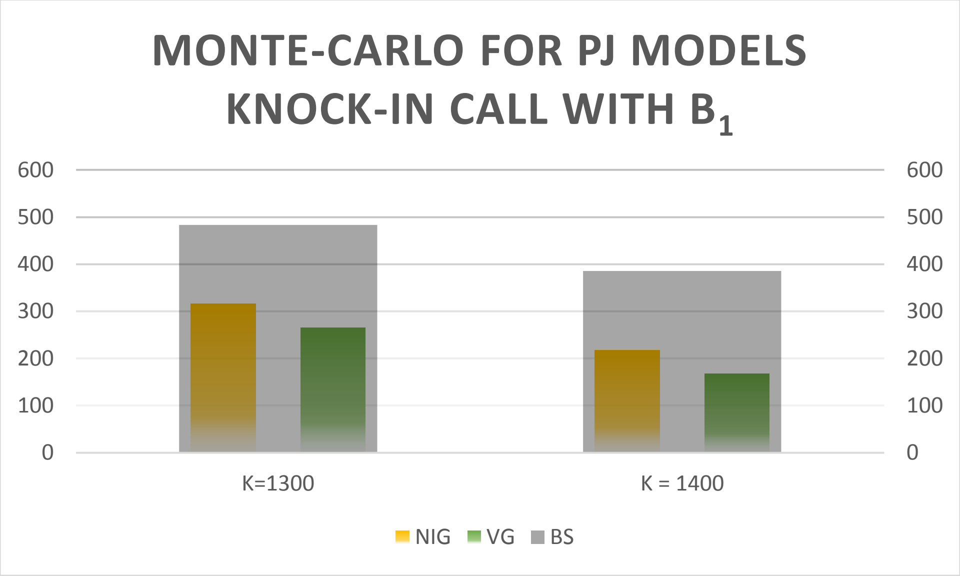

Figure 22. Prices for Knock-In Barrier calls in PJ models computed via Monte-Carlo algorithm for Sept 21, 2007 and barrier = 1550

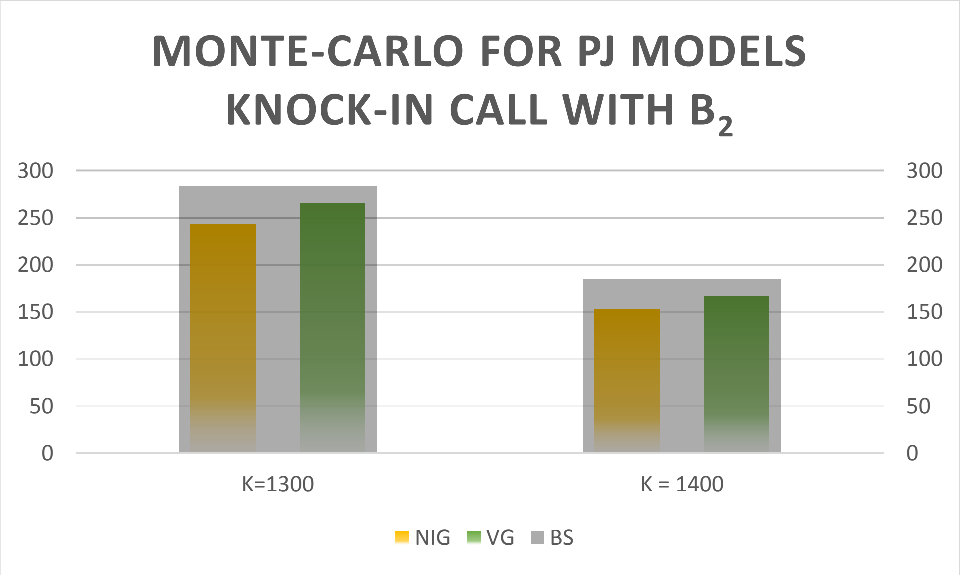

Figure 23. Prices for Knock-In Barrier calls in PJ models computed via Monte-Carlo algorithm for Sept 21, 2007 and barrier = 1660

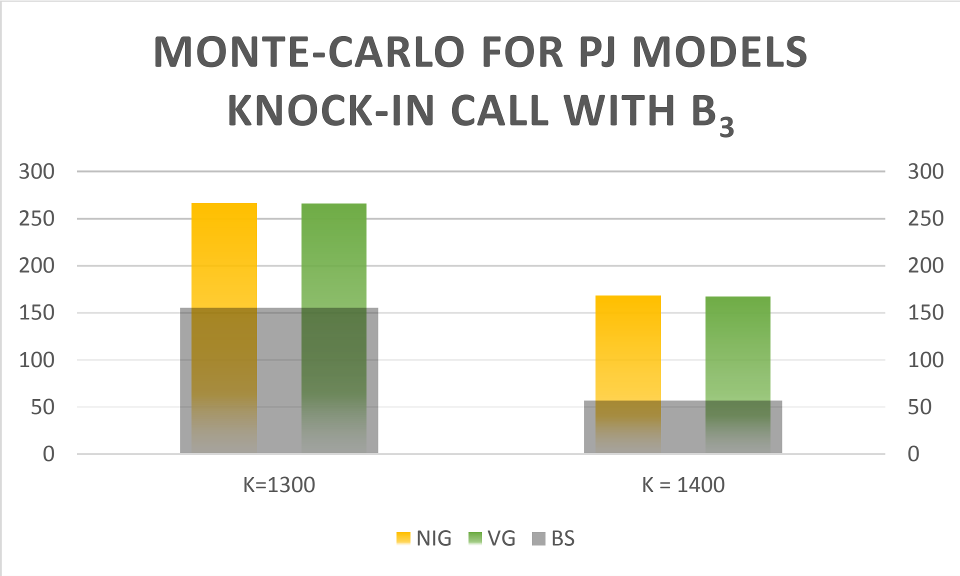

Figure 24. Prices for Knock-In Barrier calls in PJ models computed via Monte-Carlo algorithm for Sept 21, 2007 and barrier = 1650

VOLATILITY CLUSTERING IN SIMULATED PATHS

In first chapter we looked at autocorrelations of real market data. When plotting same graphs for simulated paths we should get following results:

- Any simulation based on Heston & Bates models is supposed to show non-zero ACF's results, as Heston model assumes volatility as a stochastic process where error is not constant;
- Any simulation based on Lévy processes considered should have zero ACF's results, as when pricing with Lévy processes as NIG or CGMY we use constant value for volatility, found via calibration.

This test allows to demonstrate the specific feature of Heston model that allows to fit better highly volatile data prone to clustering.

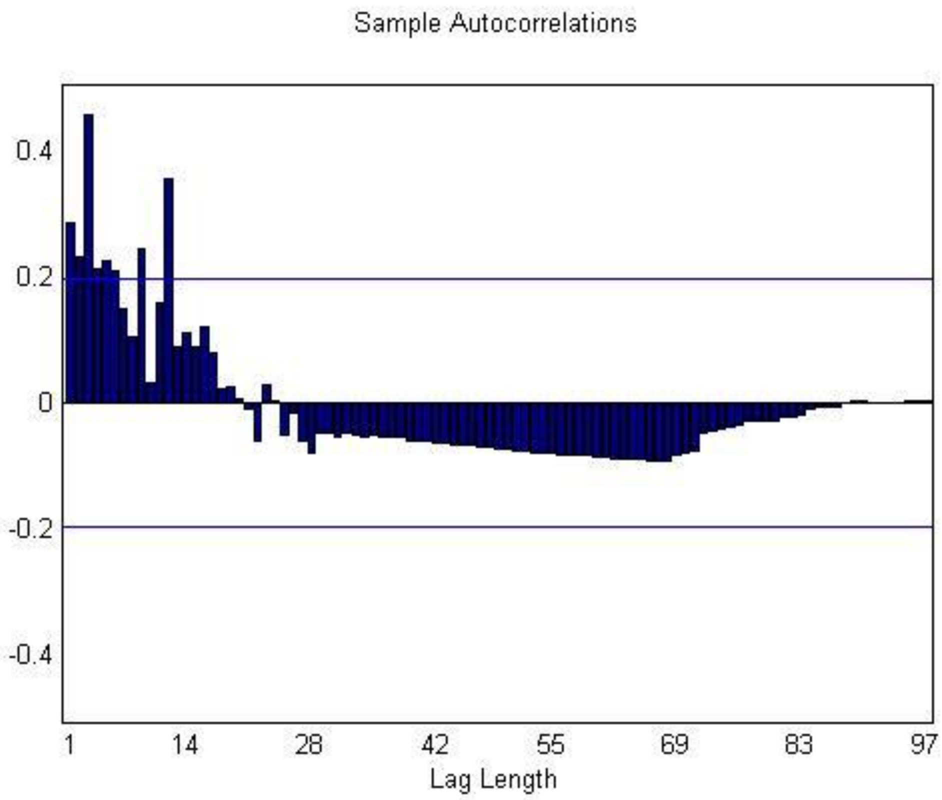

Figure 25. ACF for squared returns, simulated with Heston model

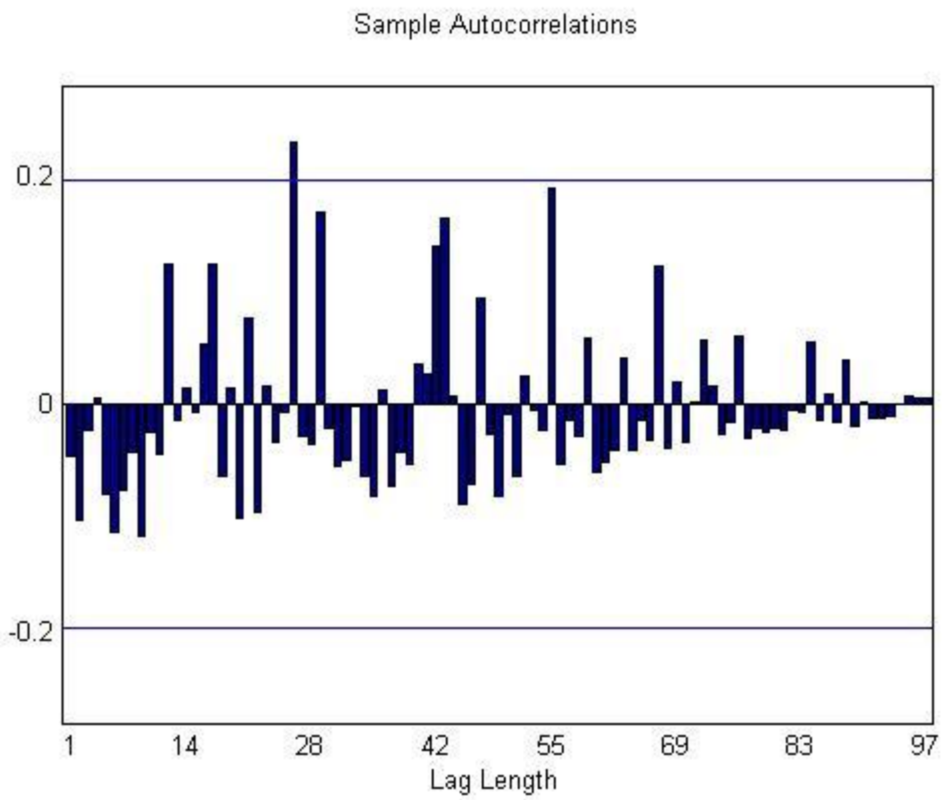

Figure 26. ACF for squared returns, simulated with NIG model

We've also checked Bates model, as another example of heteroscedastic model, and CGMY model, as example of homoscedastic model, but results aren't as representive as those shown above.

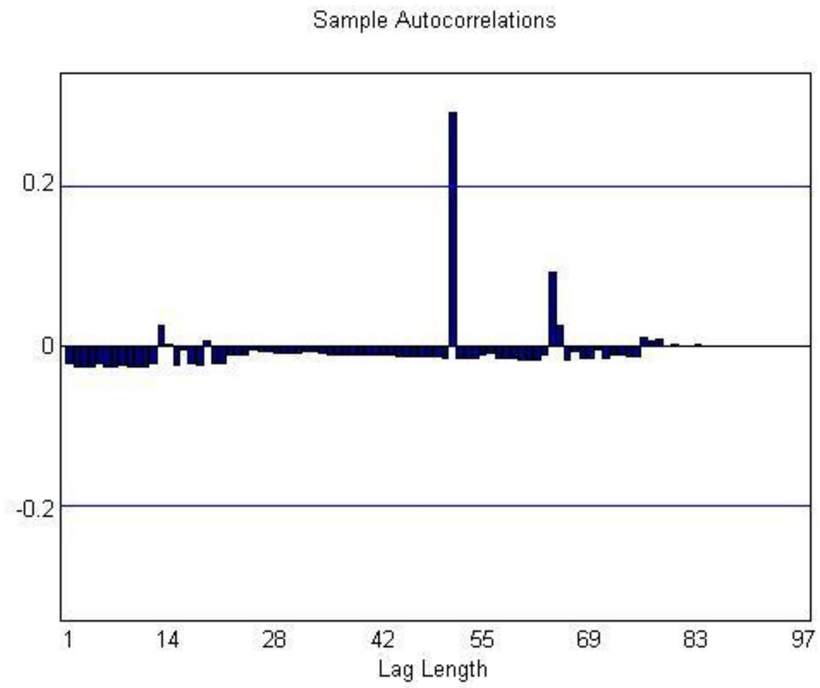

Figure 27. ACF for squared returns, simulated with Bates model

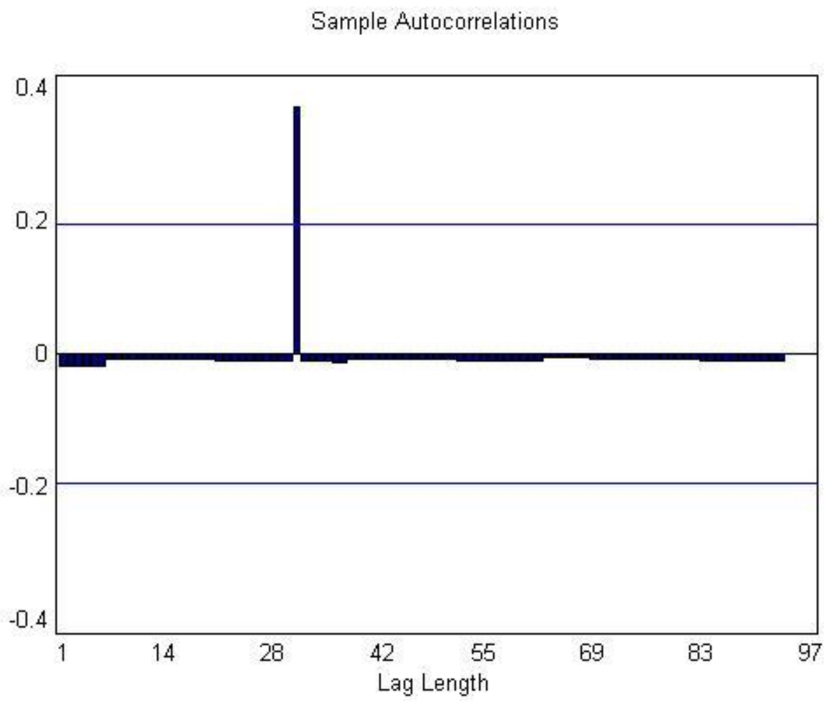

Figure 28. ACF for squared returns, simulated with CGMY model

CONCLUSION

In this Master thesis we focused on the impact of jumps in option pricing. First, we've defined types of models concerned and their underlying processes.

Then, we presented the FFT algorithm that allows to price option w.r.t. any underlying process if characteristic function is derived. We looked at jump-diffusion and pure jump models separately.

After, we applied FFT to price European options and Monte-Carlo simulation to price both European and Barrier options. Even though our results for jump-diffusion models cannot be considered as representative, the results for pure jump models demonstrate that their underlying processes fit better market data than Normal distribution, and thus should be used for option pricing. Moreover, SV models as Heston, should be considered separately due to the role they were build for.

The obtained results let us answer the initial question: jump component considerably increase the accuracy of option pricing. However, this paper does not cover different underlying assets by model, and this topic may be seen as our field of interest in future.

REFERENCES

1. Bernard C., Le Courtois O. and Quittard-Pinon F. (2008). Pricing derivatives with barriers in a stochastic interest rate environment. Journal of Economic Dynamics and Control, 32, pp. 2903-2938.
2. Black, F. and Scholes, M. (1973). The pricing of options and corporate liabilities. Journal of Political Economy, 81, pp. 637-654.
3. Bu, Y. (2007). Option Pricing using Lévy processes. Master's Thesis, Chalmers/Götenborg University, Sweden.
4. Carr, P. and Madan, D.H. (1998). Option valuation using the fast Fourier transform. Journal of Computational Finance, 2, pp. 61-73.
5. Cont, R. and Tankov, P. (2004). Financial Modelling with Jump Processes. Chapman & Hall, New-York.
6. Craine, R., Lochstoer and Syrtveit, K. (2000). Estimation of a Stochastic-Volatility Jump-Diffusion Model. University of California at Berkeley.
7. Haug, E. (2007). The complete Guide to Option Pricing Formulas (2nd Edition). McGraw-Hill, New-York.
8. Hull, J. (2011). Options, Futures, and other Derivatives (8th Edition). Pearson, New-York.
9. Jackson, M. and Staunton, M. (2001). Advanced Modellling in Finance using Excel and VBA. Wiley, Chichester.
10. Jing B.-Y., Kong X.-B. and Liu Z. (2012). Modeling High-Frequency Financial Data by Pure Jump Processes. The Annals of Statistics, 40, pp. 759 – 784.
11. Kienitz, J. and Wetterau, D. (2012). Financial Modelling: Theory, Implementation and Practice (with Matlab source). Wiley, Chichester.
12. Le Courtois, O. and Walter, Ch. (2014). Extreme Financial Risks and Asset Allocation, vol.5. Imperial College Press, London.
13. Le Courtois, O. and Quittard-Pinon, F. (2008). Fair Valuation of Participating Life Insurance Contracts with Jump Risk. Geneva Risk and Insurance Review, 33, pp.106-136.
14. Lewis, A. (2002). Algorithms: Fear of Jumps. Wilmott magazine, 1, pp. 60-67.
15. Schoutens, W. (2003). Lévy Process in Finance: Pricing Financial Derivatives. Wiley, New York.
16. Tankov, P. and Voltchkova, E. (2009). Jump-Diffusion models: a practitioner's guide. Banque et Marchés, 99.

APPENDIX

APPENDIX 1

Prices for 100 Call Options for S&P 500, collected on Jun 1, 2007. The closed price is 1536.34, interest rate is 0.05 and dividend yield id 0.019. These prices are used for pure-jump models calibration in [3].

	JUN 15 2007	JUL 20 2007	SEPT 21 2007	DEC 21 2007	MAR 21 2008	JUN 20 2008	DEC 19 2008
1300	239.1	244.5	254.0	268.5	0	296.4	322.9
1325	214.2	220.0	230.4	246.0	0	275.6	303.2
1350	189.3	195.6	207.1	223.9	239.6	255.2	283.9
1375	164.5	171.4	184.1	202.2	0	235.2	265.0
1400	139.7	147.4	161.5	181.0	198.5	215.6	246.5
1425	114.9	123.7	139.4	160.4	178.8	196.6	228.3
1450	90.4	100.5	118.1	140.4	159.6	178.0	210.7
1475	66.05	78.2	97.7	121.2	141.1	160.1	193.5
1500	42.85	56.9	78.4	103.0	123.4	142.8	176.8
1525	22.25	38.3	60.6	85.8	106.5	126.3	160.8
1550	6.95	22.25	44.4	69.9	90.7	110.6	145.3
1575	1.275	10.75	31.0	55.4	75.9	95.7	0
1600	1.15	4.5	20.1	42.6	62.4	81.9	116.3
1650	0	0	6.6	22.8	39.6	57.4	0
1700	0	0	0	10.3	22.9	37.9	67.8
1800	0	0	0	1.25	0	13.2	34.2
1900	0	0	0	0	0	0	14.4
2000	0	0	0	0	0	0	5.0

APPENDIX 2

MATLAB codes derived from [11] (ex. Heston model):

- characteristic function

```
function y = cf_heston(u,lnS,T,r,d,V0,theta,kappa,omega,rho)
% Heston
alfa = -.5*(u.*u + u*1i);
beta = kappa - rho*omega*u*1i;
omega2 = omega * omega;
gamma = .5 * omega2;

D = sqrt(beta .* beta - 4.0 * alfa .* gamma);

bD = beta - D;
eDt = exp(- D * T);

G = bD ./ (beta + D);
B = (bD ./ omega2) .* ((1.0 - eDt) ./ (1.0 - G .* eDt));
psi = (G .* eDt - 1.0) ./(G - 1.0);
A = ((kappa * theta) / (omega2)) * (bD * T - 2.0 * log(psi));

y = A + B*V0 + 1i*u*(lnS+(r-d)*T);

end
```

- applying FFT

```
function call_price_fft =
CallPricingFFT_Heston(S,K,T,r,d,V0,theta,kappa,omega,rho)
lnS = log(S);
lnK = log(K);

optAlpha = 0.5;

DiscountFactor = exp(-r*T);

FFT_N = 2^18;
FFT_eta = 0.05;

FFT_lambda = (2 * pi) / (FFT_N * FFT_eta);
FFT_b = (FFT_N * FFT_lambda) / 2;

uvec = 1:FFT_N;
ku = - FFT_b + FFT_lambda * (uvec - 1);

jvec = 1:FFT_N;
vj = (jvec-1) * FFT_eta;

tmp = DiscountFactor * psi(vj,lnS,T,r,d,V0,theta,kappa,omega,rho) ...
    .* exp(1i * vj * (FFT_b)) * FFT_eta;
tmp = (tmp / 3) .* (3 + (-1).^jvec - ((jvec - 1) == 0) );
cpvec = real(exp(-optAlpha .* ku) .* fft(tmp) / pi);


indexOfStrike = floor((lnK + FFT_b)/FFT_lambda + 1);
iset = max(indexOfStrike)+1:-1:min(indexOfStrike)-1;
xp = ku(iset);
```

```
yp = cpvec(iset);
call_price_fft = real(interp1(xp,yp,lnK));

end

function ret = psi(u,lnS,T,r,d,V0,theta,kappa,omega,rho)
FFT_N = 2^18;
optAlpha = 0.5;
FFT_eta = 0.05;
jvec = 1:FFT_N;
vj = (jvec-1) * FFT_eta;
ret =exp ( feval(@cf_heston,vj - (optAlpha +
1)*1i,lnS,T,r,d,V0,theta,kappa,omega,rho))./(optAlpha*optAlpha + optAlpha -
vj.*vj + 1i*(2*optAlpha + 1)*vj);

end
```

- Monte-Carlo path simulation (ex. Heston model)

```
function [pathS, pathV] = MC_QE(S0,r,d,T,NTime,NSim,NBatches)

kappa = 0.1;
omega = epsilon = 0.2;
rho = 0;
V0 = Vinst = 0.04;
Theta = Vlong = 0.04;


dT = T/NTime;

pathS = zeros(NSim,NTime+1,NBatches);
pathV = zeros(NSim,NTime+1,NBatches);

lnS1 = zeros(NSim,NTime+1);
lnS1(:,1)=log(S0*exp(-d*T));

V2 = zeros(NSim,NTime+1);
V2(:,1) = Vinst;

k1 = exp(-kappa*dT);
k2 = epsilon^2*k1.*(1-k1)/kappa;
k3 = exp(kappa*dT)*0.5.*k2.*(1-k1).*Vlong;

psiC = 1.5;
gamma1 = .5;
gamma2 = .5;

c1 = (r-d)*dT;
c2 = -rho*kappa*Vlong*dT/epsilon;

K0 = c1 + c2;
K1 = gamma1*dT*(kappa*rho/epsilon - .5)-rho/epsilon;
K2 = gamma2*dT*(kappa*rho/epsilon - .5)+rho/epsilon;
K3 = gamma1*dT*(1-rho^2);
K4 = gamma2*dT*(1-rho^2);

for l = 1:NBatches
```

```
    UV1 = rand(NSim,NTime);
    dW2 = randn(NSim,NTime);

    for i=2:NTime+1
        m = Vlong + (V2(:,i-1)-Vlong)*k1;
        s2 = V2(:,i-1)*k2 + k3;
        psi = s2./m.^2;

        psihat = 1./psi;
        b2 = 2*psihat - 1 + sqrt(2*psihat.*(2*psihat-1));
        a = m ./ (1 + b2);

        I1 = find(psi<=psiC);
        if isempty(I1)
        else
            V2(I1,i) = a(I1).*(sqrt(b2(I1)) + norminv(UV1(I1,i-1))).^2;
        end
        p = (psi - 1)./(psi + 1);
        V2((UV1(:,i-1)<=p) & (psi>psiC),i) = 0;
        I1b = find((UV1(:,i-1)>p) & (psi>psiC));

        beta = (1 - p)./m;
        if isempty(I1b)
        else
            V2(I1b,i) = log((1-p(I1b))./(1-UV1(I1b,i-1)))./beta(I1b);
        end

        lnS1(:,i) = lnS1(:,i-1) + K0 + K1.*V2(:,i-1) + K2.*V2(:,i) ...
                    + sqrt(K3.*V2(:,i-1) + K4.*V2(:,i)).*dW2(:,i-1);
    end
    pathS(:,:,l) = exp(lnS1);
    pathV(:,:,l) = V2;
end
```

- RMSE

```
function r=rmse(data,estimate)

I = ~isnan(data) & ~isnan(estimate);
data = data(I); estimate = estimate(I);

r=sqrt(sum((data(:)-estimate(:)).^2)/numel(data));
```