Politecnico di Milano
School of Industrial and Information Engineering
Master of Science in Mathematical Engineering

TITLE: very interesting subject, ain't it?

Supervisors: Prof. Daniele Marazzina
Prof. Ferdinando M. Ametrano

Master thesis by:
Samuele Vianello
ID: *********

Academic year 2017-2018

The secret to happiness is freedom.
And the secret to freedom is courage.

Thucydides

Contents

List of Tables

List of Figures

List of Algorithms

Abstract

Including Bitcoin in an investment portfolio increases portfolio diversification.

Acknowledgements

First of all, I would start to thank the supervisors of the present work: my gratitude goes to Prof. Ametrano, who introduced me to Bitcoin and its nature, inspired me with his passion for the subject and assisted me in the learning process, and to Prof. Marazzina, who gave me countless advices and tips during the drawing up of the thesis. Then I would like to thank the experts whose work is the backbone of mine: they are too many to fit in few lines, but hopefully I will give them credit throughout the thesis. Furthermore, I would like to express my immense gratitude to my family and friends, for always having supported me during my years of study.

Finally, I would like to thank my mother and my girlfriend, Ludovica: thank you for having been always by my side; thank you for having inspired me more than anybody else. This journey would not have been possible without you.

Thank you.

Chapter 1

Introduction

1.1 Problem under analysis

Bitcoin [**?**] is a decentralized network whose aim is to provide a peer-to-peer electronic payment system. In recent years it has astonished people who had the opportunity to study it, with an increasing number of supporters and detractors worldwide: it has been called a bubble, a Ponzi scheme, it has been labelled as sound money and store of value. But labels can be dangerous, and excessive labeling is usually useless: for this reason the present work does not aim at settling the dispute. Bitcoin is hard to understand, being at the crossroad of various disciplines, going from networking theory to cryptography, from economics to game theory. Among this multitude, mathematics and cryptography are perhaps the easiest way to approach Bitcoin, being difficult to misinterpret these two disciplines: of course there are engineering tradeoffs and debates around them, but the technical foundations are sound. For this reason, the present work wants to give a brief but satisfactory introduction to the mathematical structures and the cryptographic primitives behind Bitcoin, focusing in particular on the role played by the digital signature scheme. We will outline the problems presented by ECDSA, the scheme actually implemented, and show how they could be solved by Schnorr, a scheme that might be introduced in the protocol in the coming years. On top of that, we will delve into the technicalities of higher level constructions enabled by Schnorr, that will help in solving some of the key problems of Bitcoin, namely privacy and fungibility[1]. Bitcoin was intended to be anonymous, but it is not: it is pseudonymous, in the sense that it is possible to link transactions to a unique identity through blockchain analysis. If this digital

[1]Fungibility is the property of a good or a commodity whose individual units are interchangeable.

identity is unveiled, connecting to a real identity, that person could be persecuted for his involvement in Bitcoin (e.g. in authoritarian regime). The lack of privacy contributes to the lack of fungibility, allowing to distinguish between different bitcoins[2]. These problems affects heavily the monetary use case and in general have to be properly addressed: their improvement will be the *leitmotiv* of this present work.

The present thesis has been written during the author's fellowship at the Digital Gold Institute, a research and development center focused on teaching, consulting, and advising about scarcity in digital domain (Bitcoin and crypto-assets) and the underlying blockchain technology. The author has actively contributed to the development of the Python library that can be found at `https://github.com/dginst/BitcoinBlockchainTechnology`: for this reason, some optimization procedures are presented throughout the thesis.

1.2 Thesis structure

In Chapter 11212113 we will start presenting the mathematical foundations of the cryptography underpinning the Bitcoin protocol: in particular we will recall the concepts of group and field (Section **??**), we will introduce the general idea of elliptic curve (EC, Section **??**) and show how to induce a group structure on an EC through a proper definition of the addition operation (Section **??**).

In Chapter **??** we will give an overview of the public key cryptography based on elliptic curves: thanks to the mathematical introduction of Chapter **??** we will be able to understand what an elliptic curve over a finite field is (Section **??**); then we will discuss elliptic curves' parameters selection and validation (Section **??**), we will present the basic idea behind asymmetric cryptography based on EC (Section **??**) and we will show how to improve the core operation of asymmetric elliptic curve cryptography (ECC), scalar multiplication, through a proper change of coordinates (Section **??**). As a practical use case the section will be concluded with the presentation of the Bitcoin curve (secp256k1 in Section **??**). The chapter will be concluded by the description of the hard problem at the basis of elliptic curves' widespread adoption in recent years, the so called discrete logarithm problem (DLP in Section **??**).

[2]Typically Bitcoin identifies the protocol, while bitcoins are the coins transferred using the protocol.

Chapter ?? will deal with ECDSA (Section ??) and Schnorr (Section ??). Throughout the chapter many topics will be touched, among which the issues of ECDSA, Schnorr's solutions and multi-signature schemes.

Finally in Chapter ?? we will delve in the technicalities of Schnorr's applications: we will study a multi-signature scheme that recovers key aggregation (MuSig in Section ??), a threshold signature scheme (Section ??) and the benefits arising from the use of adaptor signatures (Section ??) for cross-chain atomic swaps (Section ??) and for the Lightning Network (Section ??).

Chapter 6 will conclude the thesis with a summary in which we will draw interesting conclusions.

1.3 Notation

- Prime numbers: the lowercase letter p is used to represent an odd prime number;
- Fields: in Chapter ?? for a general field the letter K is used, while the finite fields of order $q = p^k$, where p is an odd prime and k an integer, are represented as $\mathbb{F}_q$;
- Elliptic curves: in general an elliptic curve over a field K is denoted by $E(K)$, which represents the set of points satisfying the generalized Weierstrass equation with coefficients in K plus the point at infinity. From Chapter ?? onward, we will deal with EC over finite fields: in this case lowercase letters are used to denote scalars, while the uppercase equivalent denotes the linked EC point, e.g. $qG = Q = (x_Q, y_Q)$ (G is reserved to the generator of the group). Whenever a second generator is needed we use the capital letter H: this does not constitute a conflict with the cofactor h of the group since typically the two generators are NUMS (*nothing up my sleeves*), meaning that we do not know the discrete logarithm of one with respect to the other and viceversa;
- Elliptic curves' key pair: the pair of private and public key is denoted as $\{q, Q\}$, where $Q = qG$. Whenever a second point is needed we use the couple $\{r, R\}$; if more key pairs are needed subscripts are used, e.g. $q_1G = Q_1 = (x_1, y_1)$, $q_2G = Q_2 = (x_2, y_2)$ and $q_3G = Q_3 = (x_3, y_3)$. Notice that, for the sake of clarity, also the coordinate representation is adapted.

- Algorithms:
  - $||$ refers to byte array concatenation;
  - $a \leftarrow b$ refers to the operation of assignment;
  - $z \xleftarrow{\$} Z$ denotes uniform sampling from the set Z and assignment to z;
  - The function bytes(x), where x is an integer, returns the byte encoding of x;
  - The function bytes(Q), where Q is an EC point, returns $bytes(\text{0x02} + (y_Q \& 1)) \,||\, bytes(x_Q)$[3];
  - The function int(x), where x is a byte array, returns the unsigned integer whose byte encoding is x;
  - The function hash(x), where x is a byte array, returns the hash of x. In particular, when dealing with the Schnorr signature, it returns the 32 byte SHA-256 of x;
  - The function jacobi(x), where x is an integer, returns the Jacobi symbol $\left(\frac{x}{p}\right)$. In general we have (gcd here stands for greatest common divisor):

$$\left(\frac{x}{p}\right) = \begin{cases} 0, & \text{if } \gcd(x,p) \neq 1 \\ \pm 1, & \text{if } \gcd(x,p) = 1 \end{cases}$$

    Moreover we have that if $\left(\frac{x}{p}\right) = -1$ then x is not a quadratic residue modulo p (i.e. it has not a square root modulo p) and that if x is a quadratic residue modulo p and $\gcd(x,p) = 1$, then $\left(\frac{x}{p}\right) = 1$. However, unlike the Legendre symbol, if $\left(\frac{x}{p}\right) = 1$ then x may or may not be a quadratic residue modulo p. Fortunately enough, since p is an odd prime we have that the Jacobi symbol is equal to the Legendre symbol. Thus we can check only whether $\left(\frac{x}{p}\right) = 1$ to assess if x is or is not a quadratic residue modulo p.
    Moreover, by Euler's criterion we have that $\left(\frac{x}{p}\right) = x^{\frac{p-1}{2}} \pmod p$, so that we have an efficient way to compute the Jacobi symbol.

[3]Here & denotes the bitwise AND operator, thus $y_Q \& 1$ checks whether y_Q is even (prepend 0x02) or odd (prepend 0x03), following the standard proposed in [**?**] about the compressed representation of an EC point. An EC point can be expressed in compressed or uncompressed form: the compressed form requires the x coordinate and one additional byte to make explicit the y coordinate, while the uncompressed form requires both the coordinates.

Chapter 2

Correlation Analysis

In order to get an initial insight on how Bitcoin is correlated with other assets, we will perform a correlation analysis based on the empirical time series of our data. We will focus our attention on the logarithmic returns it is the standard practice. We will often refer to logarithmic returns simply as returns, only specifying their nature when it is necessary to avoid confusion.

2.1 Empirical Correlation of Returns

We first start by performing some statistical analysis on the data in order to estimate the distribution from which they are sampled. For this part, we will consider our data as successive samples of a N-dimensional vector in $\mathbb{R}^N$, where N is the number of assets:

$$\mathbf{x}_j = \begin{pmatrix} x_{1,j} \\ x_{2,j} \\ \vdots \\ x_{N,j} \end{pmatrix}, j = 1 \dots N_{sample}$$

Each element i of the vector $\mathbf{x}_j$ represents the j^{th} realization of the returns for asset i.

Following basic statistics, we can now compute the *sample mean* of our vectors of returns as:

$$\bar{\mathbf{x}} = \frac{1}{N_{sample}} \sum_{j=1}^{N_{sample}} \mathbf{x}_j = \begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \vdots \\ \bar{x}_N \end{pmatrix}$$

where $\bar{x}_i = \frac{1}{N_{sample}} \sum_{j=1}^{N_{sample}} x_{i,j}$ is the sample mean of component i.

Now we compute the *sample covariance matrix* through the following formula:

$$\bar{\Sigma} = \frac{1}{N_{sample} - 1} \sum_{j=1}^{N_{sample}} (\mathbf{x}_j - \bar{\mathbf{x}})(\mathbf{x}_j - \bar{\mathbf{x}})^T$$

where $\bar{\mathbf{x}}$ represent the sample mean of the returns just introduced.

All the information needed to obtain the *correlation matrix* C are already included in $\bar{\Sigma}$, we only need to perform some further calculations:

$$C_{i,j} = \frac{\bar{\Sigma}_{i,j}}{\sqrt{\bar{\Sigma}_{i,i}\bar{\Sigma}_{j,j}}} \tag{2.1}$$

We have thus obtained an empirical estimate of the correlation between our assets returns. The formula in (2.1) is often referred to as *Pearson correlation coefficient*, from the name of the English mathematician Karl Pearson who first formulated it.

Results are reported in the following tables.

******** ADD RESULT TABLES ********

We are mainly interested in the correlation between Bitcoin and other assets returns, so we will now focus on the first row (or equivalently column, by symmetry) of the correlation matrix.

All values are fairly close to zero, never exceeding 10% towards the positive or the negative side. One may thus wonder whether these correlations are *statistically significantly* different from zero. To answer this question, we will introduce two statistical tests to check the correlation significance.

2.2 Correlation Significance

The very core of Inferential Statistics, the branch of statistics that allows to draw conclusions from the information contained in a set of data, is hypothesis testing.

In our case, we are specifically interested in testing if the sample correlation coefficients are significantly different from zero or not. Both of the following tests are presented in the most general form for a sample of two variables, their distribution correlation ρ and their sample correlation $\hat{\rho}$.

Following standard testing procedure, we specify the *null hypothesis* and the *alternative hypothesis*:

$$\mathbf{H_0}: \quad \rho = 0 \quad vs. \quad \mathbf{H_1}: \quad \rho \neq 0$$

These will be common to both presented tests.

2.2.1 *t*-test

Our first test is based on Student's t-distribution and the following t-statistic:

$$t = \hat{\rho}\sqrt{\frac{n-2}{1-\hat{\rho}^2}} \tag{2.2}$$

which under the null hypothesis is distributed as a Student's t with $n-2$ degrees of freedom, where n stands for the cardinality of the sample. We can thus proceed by computing the relative p-value and compare it to a given level of confidence α (usually $\alpha = 95\%$). The result of the test will be deduced as follows:

- $p - value < 1 - \alpha$: we have statistical evidence to state that the correlation is *significantly* different from zero;
- $p - value \geq 1 - \alpha$: there is *no statistical evidence* to state that the correlation is different from zero.

2.2.2 Permutation test

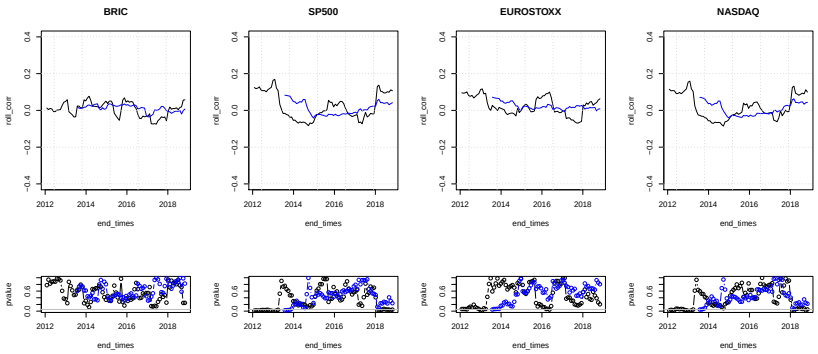
The permutation test is based on building an empirical distribution of values for the correlation by sampling different pairs of X and Y variable and then computing Pearson's correlation. If this is done a large enough number of times, we obtain an empirical distribution of possible values. From this distribution we can then obtain the p-value of the test and thus the final result in the same way as in the previous case.

2.2.3 Significance results

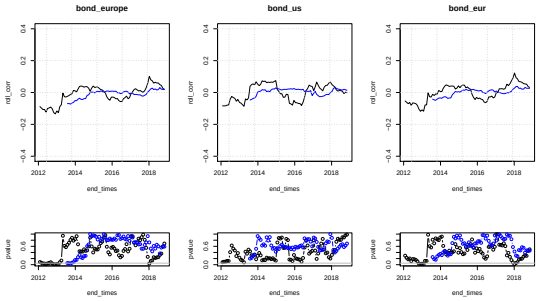
********Comments and numerical results**********

2.3 Rolling Correlation

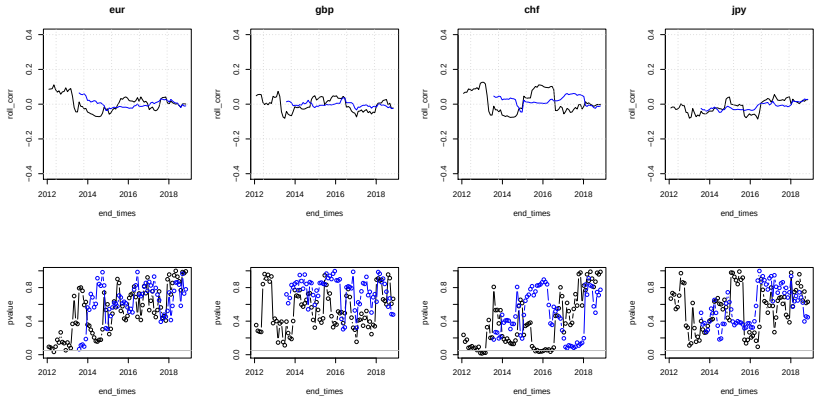
Our study so far has focused on the analysis of the dataset as a whole, with values spanning from 2010 to 2018. This is clearly important if we want to obtain a general overview of the period, but it is also interesting to see how the correlation between the assets has evolved through. Therefore, we present in Figure 2.1 the results obtained from calculating the correlation between Bitcoin and the other assets using rolling windows of 36 and 18 months, updated monthly.



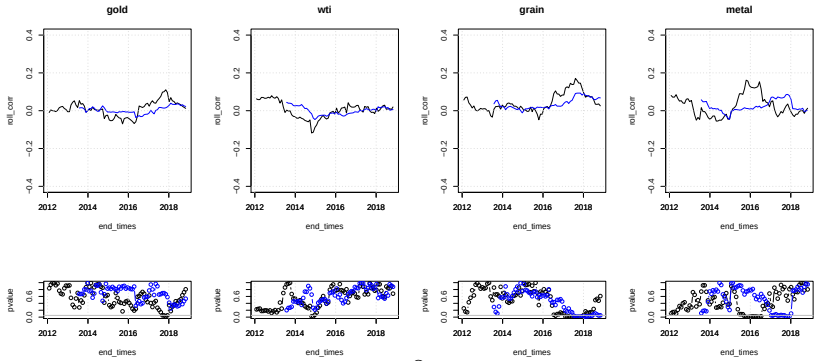
(a) Stocks



(b) Bonds



(c) Currency exchange



(d) Commodities

Figure 2.1: Plots of rolling correlation for the different asset classes (on top) and significance for each value (on bottom). Blue lines are the 3-year rolling correlations, while the black ones have a window of 18 months. Both computations are updated monthly.

There are two graphs for each asset: in the top plots levels of the rolling correlations are represented using two different colours, blue for the 3-year and black for the 18-month windows; in the bottom plots we included the significance of each rolling correlation through its p-value. The grey horizontal line represents the 5% level of significance[1].

The main conclusion we can draw from these images is that the correlation of any asset with Bitcoin is hardly ever significantly different from zero, and when it is, its absolute level is never more than greater than 20% for a small period of time.

To confirm the fact that Bitcoin is not correlated with any asset, we can also take a look at the path of the rolling correlations: there is no line that is always above zero, nor below. This indicates that there is no underlying trend, whether positive or negative, and the correlation one might find is only temporary.

**** maybe add more comments on the results ****

[1]As we explained in the previous section, to check that a sample correlation is *significantly* non zero, we compare the p-value of the test to a given level, here $1 - \alpha = 5\%$. Graphically, whenever the dots are above the grey line in Figure 2.1, the corresponding correlation is *not* significantly different from zero.

Chapter 3

Presentation of the Models

In this chapter we will present the stochastic frameworks in which we developed our analysis. We first introduce the Merton Model presented in 1976 by R.C. Merton: he added log-normal jumps to the simple B&S dynamics of the asset price. Then we move to the *stochastic volatility* model of Heston 1993. Heston introduced a new stochastic process that accounts for the variance of the underlying which evolves as a B&S with a stochastic volatility term. The last model we will present was introduced by Bates in 1996 and it is the combination of the former two: an asset dynamics which include jumps and is driven by a stochastic volatility. All models are first introduced in the one dimensional case and then generalised to the n dimensional case which was then implemented in our code.

3.1 Preliminary Notions

In this section we will briefly present the Black&Scholes framework and asset dynamics, introduce the notion of Poisson process and present the CIR process. All of these building blocks will be required to fully understand the models to follow.

3.1.1 B&S Model

3.1.2 Poisson Process

3.1.3 CIR Process

3.2 Merton Model

******maybe add some words right here ********

3.2.1 Original Univariate Model

The first jump diffusion model was originally introduced in [5] in order to account for the leptokurtic distribution of real market returns and to model sudden fall (or rise) in prices due to the arrival of new information. The asset price dynamics is modelled as follows:

$$\frac{dS_t}{S_t} = \alpha dt + \sigma dW_t + (Y_t - 1)dN \tag{3.1}$$

where α and σ are respectively the drift and the diffusion of the continuous part, Y_t is a process modelling the intensity of the jumps and $N(t)$ is the Poisson process driving the arrival of the jumps and has parameter λ.

We can rewrite (3.1) in terms of the log-returns $X_t = log(S_t)$ and obtain, following the computations in [4] and using theory from [6]:

$$dX_t = (\alpha - \frac{\sigma^2}{2})dt + \sigma dW_t + \log(Y_t) \tag{3.2}$$

that has as solution:

$$X_t = X_0 + \mu t + \sigma W_t + \sum_{k=1}^{N(t)} \eta_k \tag{3.3}$$

where X_0 is the initial value of the log-returns, $\eta_k = \log(Y_k) = \log(Y_{t_k})$ and t_k is the time when the k^{th} Poisson shock from $N(t)$ happens. We use $\mu = \alpha - \frac{\sigma^2}{2}$ for ease of notation throughout the paper. Following [5], we take η_k *i.i.d.* (independent and identically distributed) and Gaussian, in particular $\eta \sim \mathcal{N}(\theta, \delta^2)$. Another choice for the distribution of η is given in [3].

It is often useful when dealing with market data that are by nature discrete, to consider a *discretized* version of (3.3) in which the values are sampled at intervals of Δt in $[0, T]$. We thus get that for $X_i = \log(\frac{S_{i+1}}{S_i})$:

$$X_i = \mu\Delta t + \sigma\sqrt{\Delta t}\, z + \sum_{k=1}^{N_{i+1}-N_i} Y_k \tag{3.4}$$

where we denote $X_i = X_{t_i}$, $N_i = N(t_i)$ and $t_i = i\Delta t$ with $i = 0 \dots N$, $t_N = N\Delta t = T$, z is distributed as a standard Gaussian $z \sim \mathcal{N}(0,1)$.

The Poisson process $N(t)$ in (3.4) is computed at times t_{i+1} and t_i and these quantities are subtracted. Following basic stochastic analysis, one can prove that the resulting value $N_{i+1} - N_i$, is distributed as a Poisson random variable N of parameter $\lambda\Delta t$. This allows us to provide an explicit formulation for the transition density of the returns using the theorem of total probability:

$$f_{\Delta X}(x) = \sum_{k=0}^{\infty} \mathbb{P}(N = k) f_{\Delta X|N=k}(x) \tag{3.5}$$

This is an infinite mixture of Gaussian distributions, due to the infinite possible realization of the Poisson variable, and renders the estimation of the model through MLE technique intractable, see [2]. To solve this problem we introduce a first order approximation, as it's been proposed in [1]. Considering small Δt, so that also $\lambda\Delta t$ is small, we obtain that the only relevant terms in (3.5) are the one for $k = 0, 1$. The formula for the transition density becomes:

$$f_{\Delta X}(x) = \mathbb{P}(N = 0) f_{\Delta X|N=0}(x) + \mathbb{P}(N = 1) f_{\Delta X|N=1}(x)$$

expressing it explicitly:

$$f_{\Delta X}(x) = (1 - \lambda\Delta t)\ f_{\mathcal{N}}(x; \mu, \sigma^2) + (\lambda\Delta t)\ f_{\mathcal{N}}(x; \mu + \theta, \sigma^2 + \delta^2) \tag{3.6}$$

where $f_{\mathcal{N}}(x; \mu, \sigma^2)$ is the density of a Gaussian with parameters $\mathcal{N}(\mu, \sigma^2)$.

3.2.2 Multivariate Model

Starting from the univariate model introduced in [5], we developed a generalization to n assets including only idiosyncratic jumps:

$$\frac{dS_t^{(j)}}{S_t^{(j)}} = \alpha_j dt + \sigma_j dW_t^{(j)} + (Y_t^{(j)} - 1) dN_t^{(j)} \tag{3.7}$$

where $\mathbf{S}_t$ are the prices of the assets, $j = 1...n$ represents the asset, α_j are the drifts, σ_j are the diffusion coefficients, $W_t^{(j)}$ are the components of an n-dimensional Wiener process $\mathbf{W}_t$ with $dW^{(j)}dW^{(i)} = \rho_{j,i}$, η_j represent

the intensities of the jumps and are distributed as Gaussian: $\eta_j \sim \mathcal{N}(\theta_j, \delta_j^2)$. Finally, $N^{(j)}(t)$ are Poisson processes with parameters λ_j, which are independent of $\mathbf{W}_t$ and of one another.

In order to calibrate the parameters to the value of the market log-returns, we used a Maximum Likelihood approach. We thus maximize:

$$\mathcal{L}(\psi|\Delta\mathbf{x}_{t_1}, \Delta\mathbf{x}_{t_2}, \ldots, \Delta\mathbf{x}_{t_N}) = \sum_{i=1}^{N} f_{\Delta\mathbf{X}}(\Delta\mathbf{x}_{t_i}|\psi) \tag{3.8}$$

where $\psi = \{\{\mu_j\}, \{\sigma_j\}, \{\rho_{i,j}\}, \{\theta_j\}, \{\delta\}_j, \{\lambda_j\}\}$ are the model parameters, $f_{\Delta\mathbf{X}}$ is the transitional density of the log-returns which is computed approximately using the theorem of total probability. For a full insight on the model and the calibration procedure, please refer to the *APPENDIX LINK*

3.3 Heston Model

3.4 Bates Model

Chapter 4

Calibration of the Models

In this chapter we will present the stochastic frameworks in which we developed our analysis. We first introduce the Merton Model presented in 1976 by R.C. Merton: he added log-normal jumps to the simple B&S dynamics of the asset price. Then we move to the *stochastic volatility* model of Heston 1993. Heston introduced a new stochastic process that accounts for the variance of the underlying which evolves as a B&S with a stochastic volatility term. The last model we will present was introduced by Bates in 1996 and it is the combination of the former two: an asset dynamics which include jumps and is driven by a stochastic volatility. All models are first introduced in the one dimensional case and then generalised to the n dimensional case which was then implemented in our code.

Chapter 5

Markowitz Mean-Variance Portfolio Optimization

Chapter 6

Conclusions

The aim of this thesis is to give an introduction to the Schnorr signature algorithm, starting from the mathematics and the cryptography behind the scheme, and present some of its amazing applications to Bitcoin, detailing the benefits and the improvements that would arise from its deployment. We started with a brief but thorough description of the mathematical structures (Chapter **??**) and cryptographic primitives (Chapter **??**) that underpin digital signature schemes based on elliptic curve cryptography. In Chapter **??** we presented both ECDSA and Schnorr algorithm, respectively the one actually implemented in Bitcoin and the one that is under development. We compared the two schemes, investigating ECDSA lacks and Schnorr benefits, that ranged from security to efficiency. In particular we focused on the linearity property, that turned out to be the key for the higher level construction presented in Chapter **??**.
We have seen how to traduce utilities already implemented in Bitcoin in terms of Schnorr signatures: multi-signature schemes are implemented through MuSig (Section **??**), whose main advantage is to recover key aggregation; threshold signatures can be deployed through the protocols presented in Section **??**, that makes them indistinguishable from a single signature; the last application we studied has been adaptor signature and its benefits to cross-chain atomic swaps and to the Lightning Network.

The immediate benefits that Schnorr would bring to Bitcoin are improved efficiency (smaller signatures, batch validation, cross-input aggregation) and privacy (multi-signatures and threshold signatures would be indistinguishable from a single signature), leading also to an enhancement in fungibility. All this applications would be possible in a straightforward way after the introduction of Schnorr, that could be brought to Bitcoin through a soft-fork[1]:

[1]Improvements in the protocol have to be made without consesus split.

the fact that Schnorr is superior to ECDSA in every aspect hopefully will ease the process.

The last thing we would like to point out is that, by no means, the applications presented in the present work are the unique benefits that Schnorr could bring to Bitcoin. More complex ideas take the names of Taproot [**?**] and Graftroot [**?**], and are built on top of the concepts of MAST and Pay-to-Contract: through these constructions it would be possible, in the cooperative case, to hide completely the redeem script, presenting a single signature (no matter how complex the script is). For how soft forks need to be implemented after SegWit (i.e. with an upgrade of the version number), there is incentive to develop as many innovations as possible altogether (the presence of too many version numbers with little differences would constitute a lack of privacy): for this reason, it is probable that Schnorr will come to life accompanied by Taproot.
Hopefully, we have convinced the reader that Schnorr (and Bitcoin!) is worth being studied, providing also the tools to properly understand further features and innovations other than the ones presented. Moreover, we hope that you are now motivated not only to delve deeper in the technical side of Bitcoin, but also to approach it from other sides, to fully appreciate its disruptiveness and make yourself an idea of what Bitcoin is and which possibilities it hides.

Appendix A

Bitcoin

Bibliography

[1] Ball, C. A., and Torous, W. N. A simplified jump process for common stock returns. *Journal of Financial and Quantitative Analysis 18*, 01 (1983), 53–65.

[2] Honoré, P. Pitfalls in estimating jump-diffusion models. *SSRN Electronic Journal* (01 1998).

[3] Kou, S. G. A jump-diffusion model for option pricing. *Management Science 48*, 8 (2002), 1086–1101.

[4] Martin, M. A two-asset jump diffusion model with correlation. Master's thesis, University of Oxford, 2007.

[5] Merton, R. Option prices when underlying stock returns are discontinuous. *Journal of Financial Economics 3* (01 1976), 125–144.

[6] Tankov, P., and Cont, R. *Financial Modelling with Jump Processes, Second Edition*. Chapman and Hall/CRC Financial Mathematics Series. Taylor & Francis, 2015.