ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MASTER IN THE ELECTRIC POWER INDUSTRY

DEVELOPMENT OF RISK ASSESSMENT TOOLS IN A EUROPEAN GAS AND POWER UTILITY

Author: Javier Goas Martín

Supervisor: Gabriel Sánchez Pizarro

Co-supervisor: Daniel Fernández Alonso

MADRID

June 2017

Master's Thesis Presentation Authorization

THE STUDENT:

Javier Goas Martín

....

THE SUPERVISOR

Gabriel Sánchez Pizarro

Signed: .... Date: 30/ 06/ 17

THE CO-SUPERVISOR

Daniel Fernández Alonso

Signed: .... Date: 30/ 06/ 17

Authorization of the Master's Thesis Coordinator

Dr. Luis Olmos Camacho

Signed: .... Date: ....../ ....../ ......

ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA (ICAI)

MASTER IN THE ELECTRIC POWER INDUSTRY

DEVELOPMENT OF RISK ASSESSMENT TOOLS IN A EUROPEAN GAS AND POWER UTILITY

Author: Javier Goas Martín

Supervisor: Gabriel Sánchez Pizarro

Co-supervisor: Daniel Fernández Alonso

Madrid

June 2017

ABSTRACT

In the context of their daily operations, energy utilities have to manage the risk related to power and gas activities. In this context, it is necessary to develop tools which can help to decide at what price the company should enter into different products and how can market prices affect the performance of the utility.

In this thesis, a flexible Monte Carlo evaluation tools has been implemented in MS Excel. The software is equipped with various price models, such as Geometric Brownian Motion, Vasicek and ARIMA, which can help predict the dynamics of the spot prices.

A study of the suitability of the GBM and Vasicek price models has been carried out for spot power and gas markets, having obtained better results the GBM model for gas markets and the Vasicek model for power markets.

Strong emphasis has been made on the valuation of Asian options with the Monte Carlo model, offering a comparison of the prices determined with the three spot price models together with the Vorst approximation. Very consistent results have been obtained for the valuation of caps and floors with the Vasicek and ARIMA, comparing the actual payoffs which would have delivered these options during past periods, and improving the Vorst benchmark pricing.

Finally, the Monte Carlo tool has been employed to offer risk indicators such as Mark-to-Market and Value at Risk.

ACKNOWLEDGEMENT

A Gabriel y a Daniel, por su inagotable paciencia.

A José Manuel, por sus ideas y sugerencias.

A mis padres, por ser el mejor ejemplo.

A Ana y a Jorge.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Motivation

On its daily basis, the energy utilities have to take decisions oriented to manage the risk related to their power and gas activities. The aim of this project is to develop in two stages, contract valuation and risk analysis tools.

The resulting tools should serve the company to take decisions on its power and gas activities and to assess the value and the risk of its portfolio of contracts. Different methodologies are assessed for this purpose: Monte Carlo procedures with naïve and antithetic approaches have been developed in MS Excel. Different econometric price models have been created such as GBM and Vasicek in MS Excel or ARIMA in R software.

1.2 Objectives

The main objectives of the master thesis are:

- Development of option valuation models based on different spot price generators for power and gas, determining their fair value.

- Develop a flexible and user-friendly software to compute the price of the contracts and suitable risk metrics in order to manage the risks of a contract.

- Allow to assess the risks of a trading position.

1.3 Structure

This document is organized as follows. Chapter 2 describes the main features of energy trading, and describes briefly power and gas markets and the main price drivers for each commodity. In chapter 3 are reviewed the main concepts of risk management, and different products to hedge this risk, as well as the methods to valuate these instruments. In Chapter 4 are presented the steps followed to develop the software, as well as the valuation methods to assess the risk of entering into option contracts. In Chapter 5, the suitability of spot prices models is assessed as well as the results of the option valuation methods, as well as the VaR and CVar calculations.

Chapter 6 summarizes the conclusions and Chapter 7 contains the annexes, which explain further the obtention of the ARIMA models and detail the code employed.

2. INTRODUCTION TO ENERGY TRADING

2.1 Commodities

Oxford Dictionary defines 'commodity' as a "raw material or primary agricultural product that can be bought and sold, such as copper or coffee" (Oxford). However, this definition seems to leave out many products which are widely regarded as commodities. This is the case for materials such as oil or energy commodities as power and gas, which do not fall into one of the categories specified in the definition. Nowadays, even greenhouse gas emissions are considered as commodities.

In economics, commodities are considered all those goods and services which are assumed to have essentially uniform quality independently of the producer (Investopedia). Therefore, several types of commodities can be identified, such as energy, agricultural crops, livestock or metals.

Commodities, being a central part in human life, have been traded since ancient times. During Bronze Age, trading in metals such as bronze, gold and copper occurred in the Mediterranean Basin and Asia Minor (Cartwright, 2012). Due to the difficulties in transport and high delivering times, natural forward markets arose for agricultural crops and metals.

In 1848 commodities trading became first formally established with the creation of the Chicago Board of Trade (CBOT). But it was not until 1864 that the CBOT standardized future contracts.

More recently, the need for risk hedge promoted the development of these markets, creating new products, techniques and clearing schemes. In the 1970's, thanks in part to the development of computers, the derivative markets appeared worldwide. (Realmarkits)

2.2 Energy trading

During much of the 20th century, electricity utilities in nearly every country had the form of integrated monopolies. They were in charge of all the activities related from generation to supply to customers. Energy trading became relevant after the 1973 oil crisis, when derivative markets appeared (Millán, 1996). Later, other electricity markets were created with the start of the liberalization process in Chile and UK in the 1980’s, and in many other countries afterwards. The liberalization in the natural gas sector and especially the development of well-functioning markets has taken more time especially in Europe. Until recently, some European markets were not considered to be liquid enough, leading to unstable prices that encouraged parties to take shelter on indexations of other commodities, mainly Brent. In recent years, there has been an increasing trend to move towards TTF or NBP indexations, as they are the most liquid hubs in Europe.

Among the most recognized energy markets are the CME group, and Intercontinental Exchange (ICE). The CME group operates the NYMEX (New York Mercantile Exchange) market, as well as other non-energy markets such as CME (Chicago Mercantile Exchange) and CBOT (Chicago Board of Trade), which focus mainly and agricultural financial products.

NYMEX most famous commodities are WTI crude oil and Henry Hub natural gas, which are references in America of the respective commodities. ICE Brent is the reference in Europe for crude oil.

2.3 Power markets

2.3.1. Introduction

Since the discovery of the principles of electromagnetic induction by Michael Faraday in 1831 and the creation of the modern electric utility by Thomas Edison in the late 19th century, the efforts to create, develop and modernize the electricity power sector have never ceased. In the early days, electricity was generated and sold locally, much in part to the low power capacity of the electric generators at the time. The benefit of the economies of scale played a big role in the improve of technologies and the power lines, boosting the possibilities of selling electricity to the reach places located far from the generating plants. (UT Austin)

The integrated monopoly scheme that was predominant in all European countries during most of the 20th century helped creating advanced transmission systems to supply energy to remote places. However, state investment has historically been much smaller in the lines interconnecting countries. While private investment has helped renew electricity assets especially on the generation side, electricity networks have continued being regulated as a monopoly due to the inefficiency of creating competition in this area. Besides, the concerns of exporting countries of the fact that expanding interconnection capacity could elevate prices for their national consumers, along with the usually problematic agreement of the share of network investment costs and revenues, has lead in many cases in Europe to independent national power markets with limited capacity interconnection. In the last years, there have been strong initiatives in the European Union to move towards a single electricity market. While it is not a reality so far, there have been major advances in regional integration, leading to several electricity transnational markets, being the most important Nord Pool and EPEX.

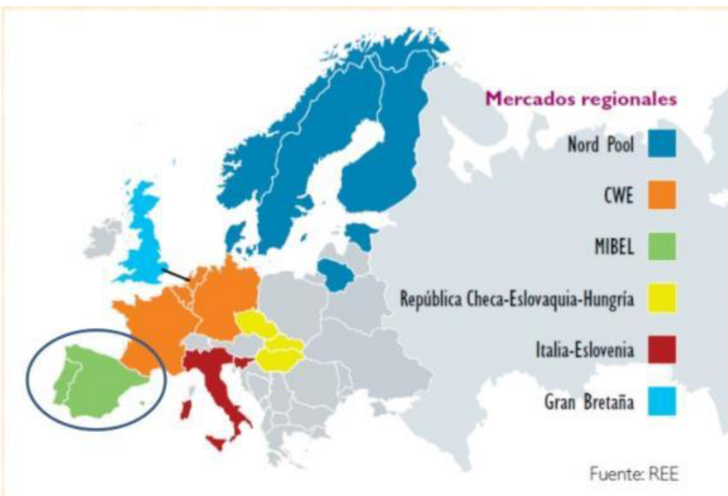


Figure 1: Regional electricity markets in the European Union (REE)

Nord Pool is an electricity spot market created in 1993 which performs operations in nine European countries. First established in the Scandinavian countries, it later expanded to the Baltic countries, and recently incorporated Germany and the United Kingdom. It is the largest electricity market in volume traded in Europe. It has been appointed as Nominated Electricity Market Operator in 15 countries, including Poland and those currently operated by EPEX, with the aim of increasing integration in the market operations. (Nord Pool)

EPEX is currently the second biggest electricity spot market in Europe. It operates in Central West European countries, including France, Germany, Austria, Switzerland and Luxembourg. EPEX is currently owned by APX, which itself operates the electricity markets in Netherlands, Belgium and the UK. The future and derivatives platform in all these countries in Central and Northern Europe is the European Energy Exchange (EEX).

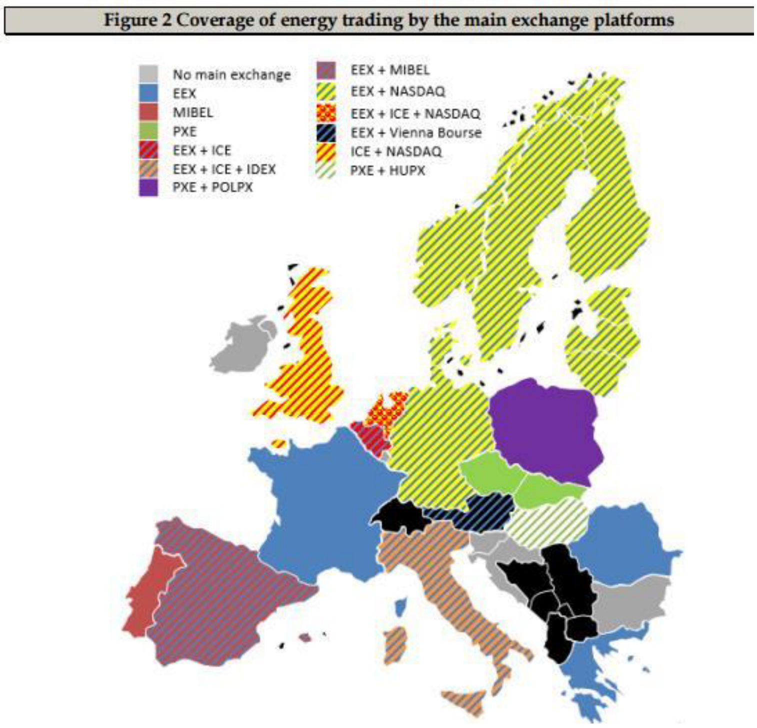


Figure 2: Coverage of energy trading by the main exchange platforms (ECA, 2015)

The Iberian Peninsula market, MIBEL, also has its operations divided, being the spot market (OMIE) in Spain and the derivatives and futures markets in Portugal (OMIP). It was in 2007 when the operations between Portugal and Spain started being performed jointly, by using the market splitting mechanism. This technique consists in consider as a single market both countries in case that the interconnection capacity is not saturated. Otherwise, the two countries are treated as separated markets and there are differences in prices (MIBEL, 2009). According to OMIE, in 2014 the price was the same in both countries for 90% of the hours. There are financial auctions on interconnection rights, which allow to perform trading of the spread between the prices of both countries.

Since 4 February 2014, a market coupling technique, Price Coupling of Regions (PCR), was implemented in the NWE and SWE regions, allowing to efficiently manage the interconnection capacity. As of today, the PCR algorithm couples the prices of 24 European countries.

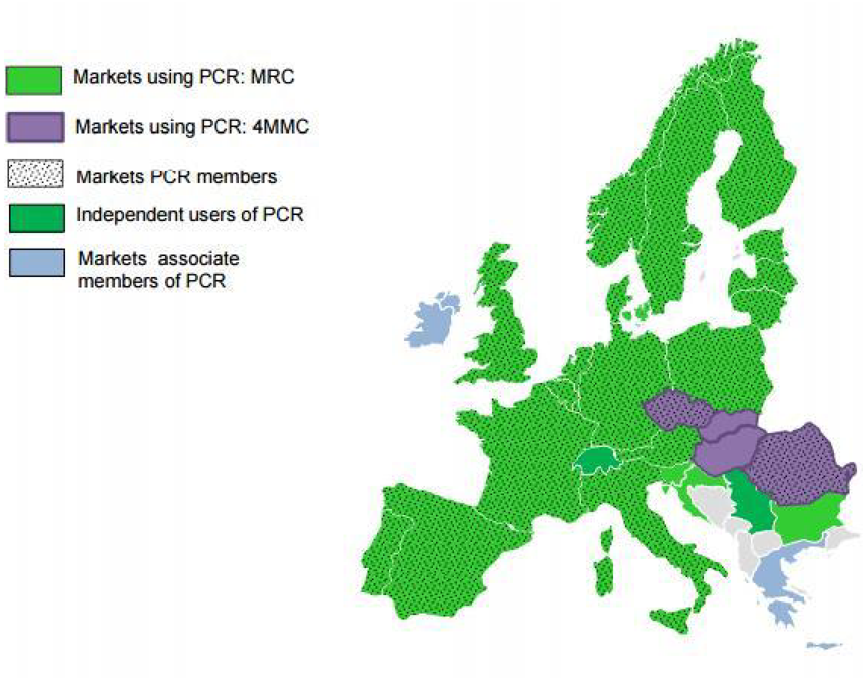


Figure 3: Countries using PCR algorithm (PCR)

2.3.2. Fundamentals of the Spanish electricity market

Spain's electricity market, as the rest of the EU power markets, is based on marginal pricing, receiving all dispatched power units the marginal bid of the most expensive plant which is allowed to produce. The marginal bids of each plant, due to economic theory, is very similar to the variable costs of the power units. This allows the generators to recover their variable costs and an important part of the fixed costs. To guarantee security of supply, technology-discriminatory capacity payments are made to generators.

In the Spanish market, there is a high percentage of non-manageable renewables. In 2016, wind energy alone generated 19.3% of the total generated power in Spain. Other non-manageable renewable production came from solar PV (3.1%), thermal solar (2.1%) and run-of-river hydro power plants (REE, 2017). These plants generally bid at very low or even zero prices as their variable costs are very low. In Spain there are also 7573 MW of nuclear plant capacity which usually offer at zero prices when available in order to avoid stops that are not profitable due to their lack of flexibility in production (REE, 2017). As to reservoir hydroelectric power plants, their strategy consists in using the water in periods where the marginal prices are higher. Discounting the production of these technologies from the demand yields the so-called thermal gap, the part of the demand that has to be covered with fossil fuel power plants. Due to lower coal prices with respect to natural gas in Spain and low carbon emission prices in the EU in the last years, the variable cost of coal plants is usually lower than the variable cost of natural gas power plants, thus making coal precede CCGTs in the merit order for dispatch. The marginal cost in the Spanish power system is usually set by coal or natural gas power plants, depending mainly on the demand and the renewable production.

It is therefore observable that the main variables which explain the electricity prices in Spain are the cost coal and natural gas and the thermal gap. This last variable is affected by several factors including renewables production, yearly hidraulicity and availability of power units, especially nuclear.

2.3.3. Evolution of the Spanish electricity prices

Since 2012 the Spanish electricity mix has barely changed, due to several reasons. First, these years there has been system power overcapacity with respect to national demand, after its drastic decrease during the financial crisis which hit Spain severely between 2008 and 2015. As a consequence, there has not been more space in the electricity mix for new nuclear, gas and coal plants. Regarding renewables, the approval of the RD Law 1/2012 eliminated all the subsidies in their installation from 2012, which led to a stop in investments in these technologies in Spain.

However, in order to comply with the 2020 European energy objectives several auctions for installation of renewable technologies have taken place in 2016 and at the beginning of 2017, or are planned for the next months. The January 14, 2016 auction allowed the installation of 500 MW of wind energy installed capacity and 200 MW of biomass energy. The May 17, 2017 auction allowed the installation of 2,979 MW of wind energy, and there is another auction for 3,000 MW of renewable power which will take place in 2017. Therefore, in the following years, there will be an increase in renewable installed capacity when these new installations are completed.

Regarding coal plants, on April 28, 2017, a committee of Member States agreed setting stricter limits for NOx and SOx emissions, to be applied from 2021 (Wynn, 2017). This will force most of the plants to either invest especially in denitrification systems or be withdrawn from the power system. In Spain, some coal power plants are expected to be shut down. Endesa has shown intentions of closing Compostilla (1,200 MW) and Teruel (1,102 MW) and make investments in As Pontes (1,468 MW) and Litoral (1,159 MW). It will also shut down Anllares (365 MW) and install denitrification systems in La Robla (655 MW), both shared with Gas Natural Fenosa. EDP has installed denitrification systems in Aboño (916 MW), and plans to do so in Soto de Ribera (600 MW) although reducing their capacity. Viesgo has also concluded with the intention of expanding Los Barrios (589 MW) lifetime to 2030. In total, there are expectations of removal of at least 3256 MW from the Spanish mix, however this could change if the Government grants subsidies to the necessary investments to survive after 2020. Other power plants for which there is higher uncertainty have not been mentioned. Therefore in the near horizon there can be important changes in the electricity mix. (REE, 2013) (Sources: news)

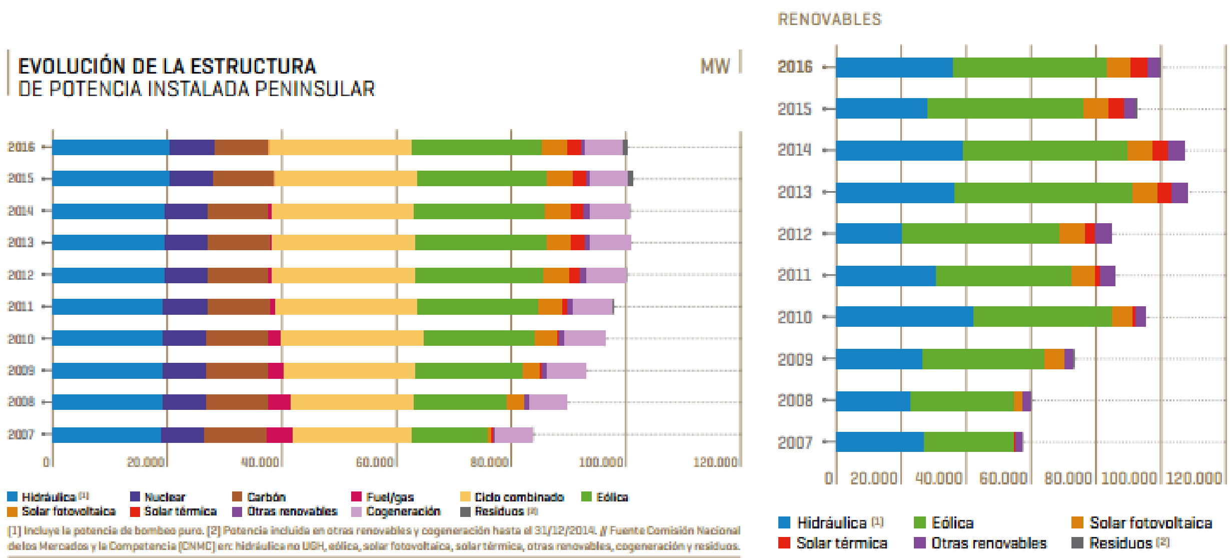


Figure 4: Installed power capacity and renewable production in Spain 2007-2016 (REE, 2017)

The dynamics of the prices reflects how much Spain's power spot prices depend on renewable production. 2013, 2014 and 2016 were with high renewable production, mainly due to hydroelectricity and wind. These average price in these years were 44.25 €/MWh, 44.12

€/MWh and 40.32 €/MWh, respectively, with periods of prices below 30€/MWh especially in spring. On the other hand, renewable production was shorter in 2012 and 2015, contributing to higher average prices of 47.23 €/MWh and 50.32 €/MWh, respectively.

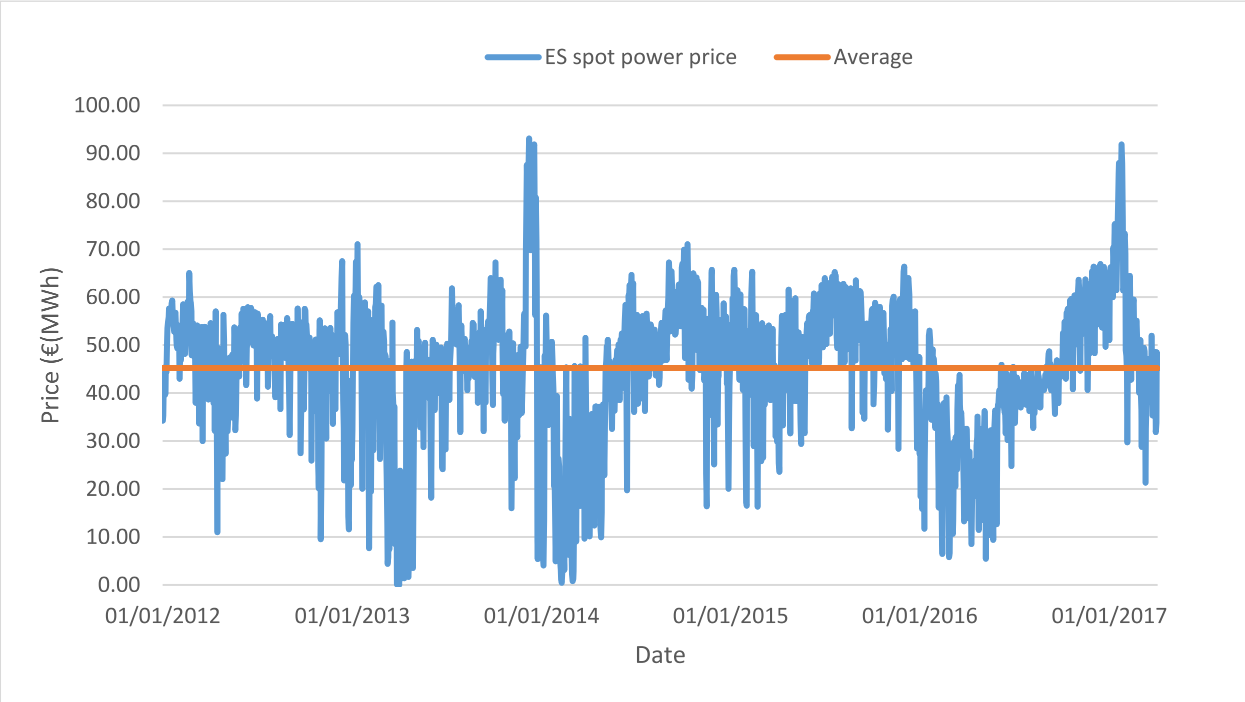


Figure 5: Spot electricity prices in Spain (Thomson Reuters)

2.4 Gas markets

2.4.1. Introduction

Natural gas is a gaseous fossil fuel primarily consisted by methane (CH_4). It can also contain quantities of other hydrocarbons, such as propane and butane or even non-hydrocarbons. According to the IEA, in 2014 natural gas represented 21.2% (2,904.2 Mtoe, approximately 3,226.9 bcm) of total primary energy supplied (13,669 Mtoe), being the third energy source only after oil (31.3%, 4287.7 Mtoe) and coal (28.6%, 3917.4 Mtoe). (IEA, 2016)

In 2015, the main natural gas producers were the United States (767.3 bcm) and Russia (573.3 bcm). Other important producers are some MENA countries, mainly Iran, Qatar, Saudi Arabia and Algeria; as well as China, Norway, and Turkmenistan, all with quotas below 200 bcm. (BP, 2016)

The main natural gas exporters are Russia (192 bcm in 2015, mainly though pipelines), Qatar (115 bcm, mainly by LNG) and Norway (115 bcm, through pipelines and LNG). The exports from the three countries amounts more than half the world exports (50.8%). The biggest natural gas importers are located in the developed Far East countries, mainly Japan, China and South Korea, as well as in the European Union. (IEA, 2016)

In the past years the development of the sector of Liquefied Natural Gas (LNG), which allows to transport gas to far destinations, has caused a decrees in its prices, establishing competition with gas transported through pipelines, especially in Europe. Qatar is the LNG global leader, producing almost one third of LNG worldwide. There are other important suppliers on nearly any continent, such as Australia, Malaysia, Nigeria, Trinidad and Algeria. On the other hand, main importers are the Far East biggest economies as well as the UK and Spain in Europe. (Figure 6)

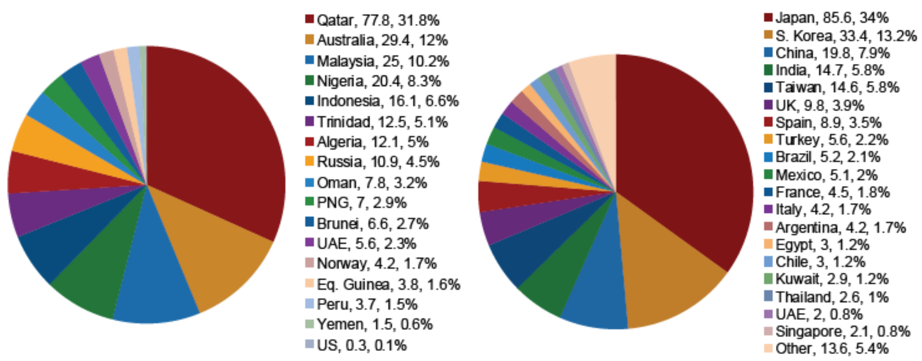


Figure 6: main LNG exporters (left) and importers (right) in MT in 2015 (IGU, 2016)

In the past decade, the development of the unconventional gas industry mainly in the United States and Canada has had a several impact in the world natural gas market structure. This has allowed the US to become a net exporter, while it depended heavily on imports from Canada historically.

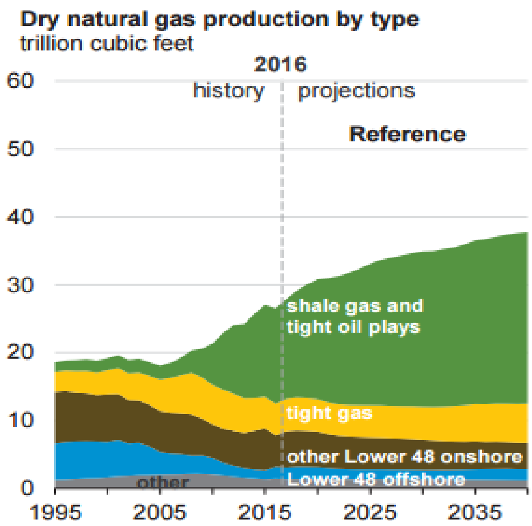


Figure 7: Historical and forecasted unconventional gas production in the US. (US EIA, 2017)

Three world zones can be distinguished from the supply and demand analysis: North America, European countries and Far-East Asian countries. The main references for natural gas each zone are Henry Hub, in USA; NBP in the UK or TTF in the Netherlands and JCC, in Japan.

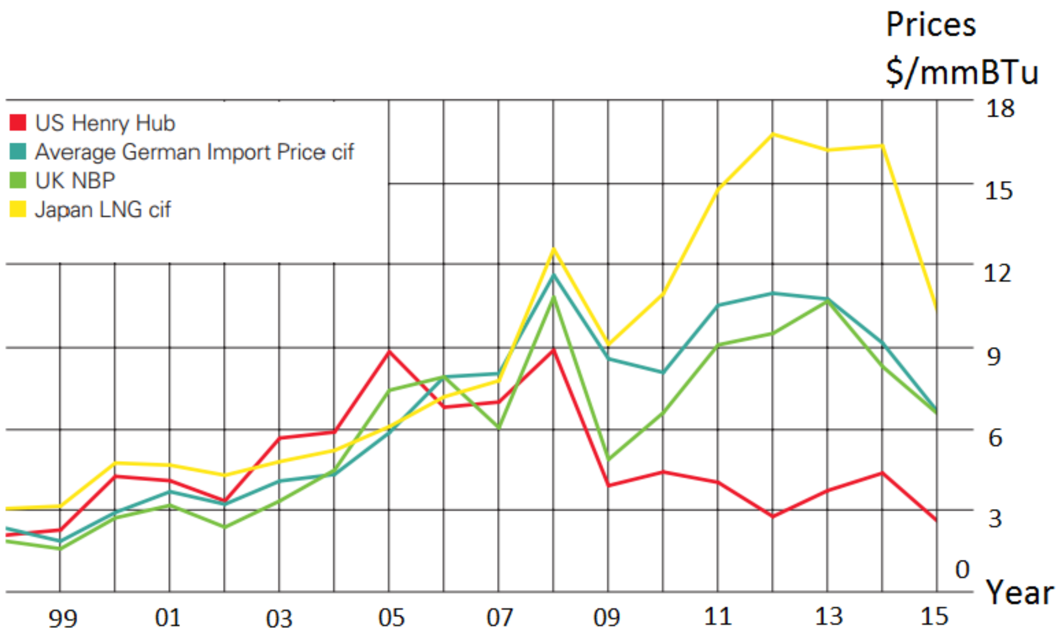


Figure 8: Prices in the reference worldwide hubs 1998-2015 ($/MMBtu) (BP, 2016)

2.4.2. America

In 2015, production in North America summed 984.0 bcm in North America (767.3 bcm in the US, 163.5 bcm in Canada and 53.2 bcm in Mexico). Total consumption in North America accounted for 963.6 bcm (778.0 bcm in the US, 102.4 bcm in Canada and 83.2 bcm in Mexico), thus being the supply very balanced with demand in the region (BP, 2016). The recent development of the shale gas industry in the US and Canada along with the limited liquefaction capacity has caused the Henry Hub price, the reference natural gas benchmark in the United States, to have dropped at certain periods even below 3 $/MMBtu. This is a major decrease from the price spike above 13 $/MMBtu in spot price after hurricane Katrina (August 2005) and the prices around 7 $/MMBtu which were common between 2006 and 2008.

2.4.3. Asia

The Asian advanced economies are among largest natural gas importers. Japan is the top worldwide importing country for natural gas (117 bcm in 2016), being China (56 bcm) and South Korea (43 bcm) also relevant. (BP, 2016)

With the exception of China, which produced 138 bcm in 2015 which serves to cover a 70% of its demand (197.3 bcm), both Japan and South Korea are dependent on LNG imports, coming from countries as Indonesia, Malaysia, the Middle East and Australia (BP, 2016). This dependence as well as the increase in Japanese natural gas consumption following the Fukushima accident has explained the increase in the natural gas prices in Japan in recent years. However, thanks to the development of the worldwide LNG industry and the supply competition, prices have recently converged towards the NBP reference.

2.4.4. Europe

EU countries are also important net gas importers. As of 2015, EU production amounted to 120.1 bcm, only covering 29.9% of total a consumption of 402.1 bcm (BP, 2016). According to projections made by (Teusch, 2012), EU demand will sum 644 bcm, while internal gas production will only account for 165 bcm in 2035, representing only 26% of total consumption.

From a steady growth in natural gas demand in the European countries, there has been a sharp decrease in demand since the beginning of the 2008 financial crisis, which has been sustained due to several factors such as an increase in energy efficiency and especially in electricity efficiency which has decimated demand and an upsurge in renewables penetration. Besides, the low greenhouse gas emission prices during these years have benefited coal over gas in electricity supply.

The top four main suppliers for EU countries are Russia, Norway, Algeria and Qatar, amounting to 93.7% of total imports, according to Eurostat. Russia, Norway and Algeria mainly export gas to the EU through pipelines. These three providers have very large influence in regional markets, especially Russia in Eastern Europe, where it is the sole gas supplier for many countries.

Algeria fundamentally exports to Spain and Italy through pipelines which cross the Mediterranean Sea, but also exports in form of LNG. Other important LNG suppliers are Qatar, which is the main LNG provider and covers a third of the worldwide LNG demand and Nigeria. The main recipients for LNG in Europe are Spain, UK, France and Italy, in descending order. The EU importing LNG capacity is currently underutilized. It allows to cover 43% of gas demand. In the following years, thanks to the development of the LNG global market which is more diversified, Europe could diversify its gas providers and reduce gas prices. (European Commission, 2016)

Partner	Value (Share %)	Net mass (Share %)
Russia	39.1	37.5
Norway	34.8	37.3
Algeria	11.6	10.6
Qatar	7.9	8.3
Libya	2.6	2.3
Nigeria	1.7	1.7
Others	2.3	2.3

Figure 9: Imports of natural gas in the European Union (Source: Eurostat)

Spain

In Spain, internal gas production summed only 24 mcm in 2014, while natural gas accounted for 20.8% (26.3 bcm) of primary energy supplied, and 17.2% (47.11 TWh) of total generated electricity. Almost all the natural gas is imported, being Algeria the main supplier, transferring 57.9% of the total gas. Around 20 bcm yearly are imported from Algeria through the Maghreb pipeline (12 bcm) and through the Mezgas pipeline (8 bcm), which crosses Morocco. The rest of the gas supply is more diversified, with around 10 countries exporting gas to Spain mainly by LNG. Norway, (11.5%), Qatar (8.6%) and Nigeria (7.8%) were the most important. Other exporters were Trinidad and Tobago, Peru and Portugal. (IEA Spain 2015 Review, 2015).

In 2015, 58% of the imports came through pipelines, while the other 42% was in form of LNG. Spain has interconnections with Portugal, France, Morocco and Algeria. Spain has six regasification plants located in Barcelona, Bilbao, Cartagena, Huelva, Mugardos and Sagunto which provide excessive regasification capacity (7.063 mcm/h, 61.87 bcm/year), one third of regasification capacity in the EU. (Sedigas, 2015) (IEA Spain 2015 Review, 2015).

However, due to limited capacity especially in the Spain to France pipelines (5.4 bcm/year) and in the Spain to Portugal pipelines (5.2 bcm/year), Spain does not use all of its regasification capacity. In 2013, only 20.6% (12.74 bcm) of its regasification capacity was used. (IEA Spain 2015 Review, 2015).

Demand has experienced a drastic decrease in the last decade, having reached its peak in 2013 with 349.83 bcm (IEA, 2015) and declining at a rate above 9% rate in 2013 and 2014. However in 2015 the tendency was again reversed and demand increased a 4.4%, having reached 315.14 GWh. (Sedigas, 2015)

3. RISK MANAGEMENT

3.1 Introduction

According to Collins Dictionary, risk is the possibility of incurring misfortune or loss. In every business, possible losses have to be estimated. The main risks a utility faces are:

Type of risk	Description
Operational risk	Risk or loss resulting from "inadequate or failed internal process, people, systems or from external events" (Bank for International Settlements, 2016). Some types of operational risks are system failures, for example cyberattacks or personnel errors.
Credit risk	Risk that a counterparty defaults on the payments agreed to pay their debt obligations. In the case of an energy utility, it will incur in credit risk when selling energy to another party. The higher the risk of default of a party, the higher the premium they will need to pay in order to borrow an amount of money.
Liquidity risk	Type of financial risk which refers to the situations where the assets cannot be sold in a short time of period in exchange for the value they are supposed to have.
Market risk	Possibility that the value of an assets decreases due to fluctuations in the markets. The risk is higher the higher the volatility of the market is.
Systemic risks	They refer to the probability of collapse on an entire interdependent system. Systemic risks are important in the banking sector every bank is exposed to immense credit risks from other financial counterparts.

Table 1: Types of risks

3.2 Market risk

The most relevant market risk of a utility are the prices of the commodities, which vary every day. A utility could face risks as many customers are signed with fixed-price contracts, or contracts which do not reflect the volatility of the spot market. The utility has several options to deal with these risks: accept them or look for a hedge.

A position where a participant is assuming the risk of the market is said to be an open position. However, utilities may try to eliminate part of the risk through derivative markets and OTC contracts, closing their position on their assets and reducing the risk of their portfolio. Another behavior which can reduce market prices is diversification. In the short term this is complicated for utilities as they have to deliver constantly the power or gas. In the long term, especially in the power market, a utility may diversify the technologies its portfolio of generating plants, reducing risk.

An agent can enter a derivative market with different interests. On one hand, there are agents who will try to hedge their risks, this is reducing the volatility of their portfolio. However, agents need to find a counterpart who agrees to assume these risks. In exchange for reducing risks, the counterpart will typically ask for a risk premium. The agents who enter open positions in markets intending to make larger profits are known as speculators. Finally, arbitrators try to find opportunities in markets, typically buying in a cheaper market and selling in another more expensive. This is very usual in energy business. For example, a natural gas agent with storage capacity may buy cheap gas in the spot market and sell it in a future market at a higher price, if there is a situation of contango (commodity prices are rising for the future months).

Although for energy companies hedging is vital, they can behave as speculators or arbitrators depending on the opportunities and their strategy.

Mark to market and VaR are indicators which can help assess the risk of a portfolio of assets:

- **Mark to market**
  Refers to the fair value of an asset based on its current price. In exchange markets, futures are usually marked-to-market daily, and the traders who suffer losses in their values have to deposit that loss to the exchange.

- **Value at risk (VaR)**

  Value at Risk is a statistical measure which quantifies the largest loss which can be suffered in a time period with a certain probability given normal conditions. VaR is measured with an α confidence level, usually between 95% and 99%.

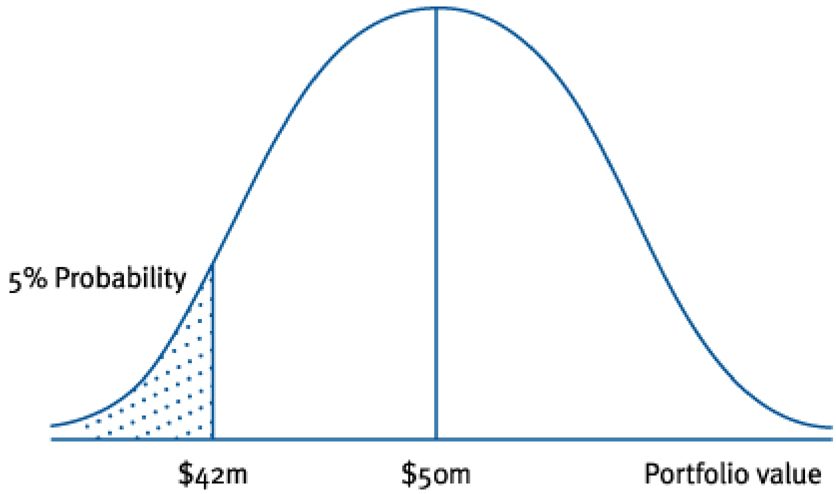


Figure 10: VaR

(Source: Kaplan Financial Knowledge Bank)

There are several methods to calculate VaR: (Farid, 2010)

- Historical Simulation: this method assumes that future returns will follow the same distribution as historical returns. Historical percentage changes in spot prices from one day to the following are studied. Then, the percentage loss in the α percentile is applied to the present spot price of the asset.
- Monte Carlo VaR: this method and historical simulation are much alike, but in Monte Carlo the distribution of the returns is determined by a probability distribution chosen by the analyst. The Monte Carlo tool developed can be used to perform the VaR calculation.
- Delta-Normal approach: this method, also known as variance-covariance, computes historical variances and covariances of all the risk factors. It implies all the asset returns follow a normal distribution. (Jorion, 2996) (Johnson, 2001)

Being Σ the covariances matrix of the returns of the portfolio, estimated from historical data, the variance of the portfolio is obtained by a linear combination of the risk factors.

$$\sigma_p = v^T \, \Sigma \, \mathrm{v}$$

Where v is the vector of coefficients which weighs the risk factors.

Then, the VaR can be obtained:

$$VaR(\alpha, t) = \sigma_p z_\alpha \sqrt{t}$$

Where z_α is the value of inverse normal distribution at an α probability level.

- **Conditional Value at risk (CVaR)**

CVaR estimates the expected loss evaluating only the $\alpha\%$ worst probabilities scenarios.

Delta hedging

The delta hedging strategy consists in obtaining a position where the value of the portfolio does not change with small changes in market prices. The delta of a portfolio is expressed as:

$$\Delta = \frac{\partial V}{\partial S(t)}$$

Where *V* is the value of the portfolio and *S* is the market value of an asset.

The delta position can be approximated by the expression: (Burguer, 2014)

$$\Delta \approx \frac{V[S + \Delta S] - V[S - \Delta S]}{2 \, \Delta S}$$

The delta hedging strategy consists in obtaining a position where the delta is neutral or zero, as the value of the portfolio will not be affected by small changes in the market price. A long position (buy) is attained when $\Delta > 0$. It means that the value of the portfolio will be higher if the commodity increases its price. A short position (sell), obtained when $\Delta < 0$, reduces the value of the portfolio if the price of the commodity rises.

For European options delta can be computed as follows:

$$\Delta_{\mathrm{C}} = e^{-r(T-t)} N(d_1)$$

$$\Delta_{\mathrm{P}} = e^{-r(T-t)} N(d_1 - 1)$$

3.3 Products traded in energy derivative markets

Forwards and futures: forward and futures contracts between two counterparts in order to buy or sell a specific amount of electricity or gas at a designated time.

Forward contracts are arranged in Over-The-Counter (OTC) markets bilaterally. Typically forward are settled at the maturity. Therefore one of the biggest inconvenient to set up forwards is be credit risk, as there can be no guarantee that the counterpart is trustworthy.

On the other hand, future contracts are traded in Organized Exchanges with common rules. In most exchanges, gains and losses are generally settled periodically, every day or week. One of the advantages of participating in an Organized Exchange is that there is higher liquidity and this allows the party to see at each moment the bid and the ask price for a product. They typically provide a bigger hedge against non-payments as they usually require the players to meet certain criteria in order to make trades. To participate in the exchange market agents are asked for a fee.

The buyer of the future is said to hold a long position, and will have a positive payoff in case the price of the underlying assets is higher at maturity than the agreed price. The seller of the future is said to hold a short position, and will have a negative payoff in case the price of the underlying assets is lower at maturity than the agreed price.

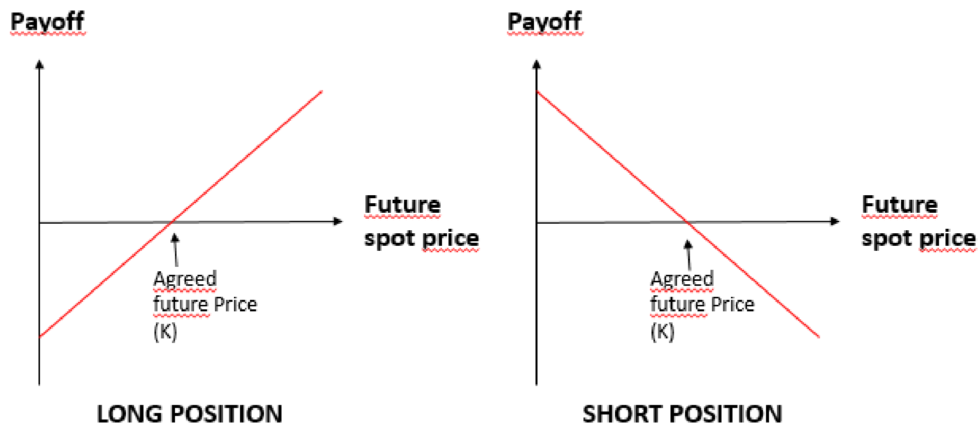


Figure 11. Payoff of a future

Being r the continuously compounding interest rate, T the maturity and t the time when the contract is set up, the non-arbitrage price of a future can be determined from the future spot price of the underlying asset:

$$F_{t,T} = S_t e^{r(T-t)}$$

However this formula has to be adjusted by including the cost of carry u or costs of storing the commodity, as well as the convenience yield c or benefits associated with holding the commodity, typically in situations of shortage of supply, resulting in:

$$F_{t,T} = S_t e^{(r+u-c)(T-t)}$$

When the convenience yield is lower than the sum of the cost of carry and the interest rate ($c < r + u$), future prices at maturity T are higher than spot prices at that time. This is the situation of a market in contango. This situation may happen when there are expectations of a demand increase or a supply shortage in the following months. There are incentives to use storage for those agents who are able to do it.

On the opposite, if the convenience yield is higher than the sum of the cost of carry and the interest rate ($c < r + u$), the market is said to be in backwardation. This is very typical in oil and gas markets when there is shortage of supply in the short term.

Figure 12: Representation of contango and backwardation situations.

(Source: Investopedia)

Swaps: a swap is an exchange of a variable price for a fixed price. It allows a party, for example the producer, to sell the commodity at a fixed price. Swaps are usually traded in OTC markets.

Options: an option is a financial derivative which gives the buyer the right, but not the obligation, to buy (call) or sell (put) an asset on a specific date or during a certain period of time at an agreed price *K* (exercise or strike price). The simplest options, known as vanilla options, can be traded in most derivative markets. More complex options are in most cases traded OTC.

Call options

An agent who buys a call option wants to hedge the risk against high prices, in exchange for giving a premium. The operation is more profitable for the buyer the more the spot price goes up at maturity. The breakeven point (zero payoff) is given when the spot price in the future equals the exercise price plus the premium.

An agent who sells a call option assumes an unlimited risk in event of high prices in exchange for receiving a premium. The agent would receive the maximum payoff if the prices do not increase over the exercise price, being it the premium of the option.

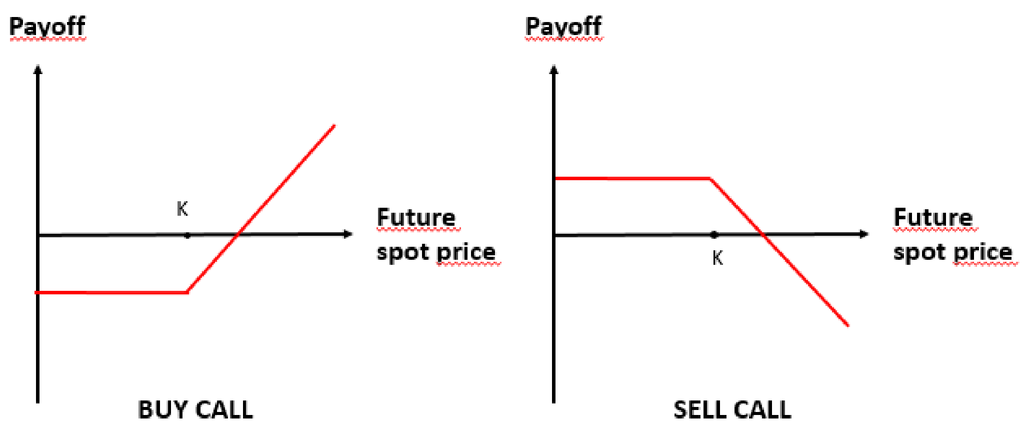


Figure 13: Representation of the payoff of a call option

Put options

An agent who buys a put option wants to hedge the risk against low prices, in exchange for giving a premium. The operation is more profitable for the buyer the more the spot price goes down at maturity. The breakeven point (zero payoff) is given when the spot price in the future equals the exercise price minus the premium.

An agent who sells a put option assumes an unlimited risk in event of high prices in exchange for receiving a premium. The agent would receive the maximum payoff if the prices do not decrease below the exercise price, being it the premium of the option.

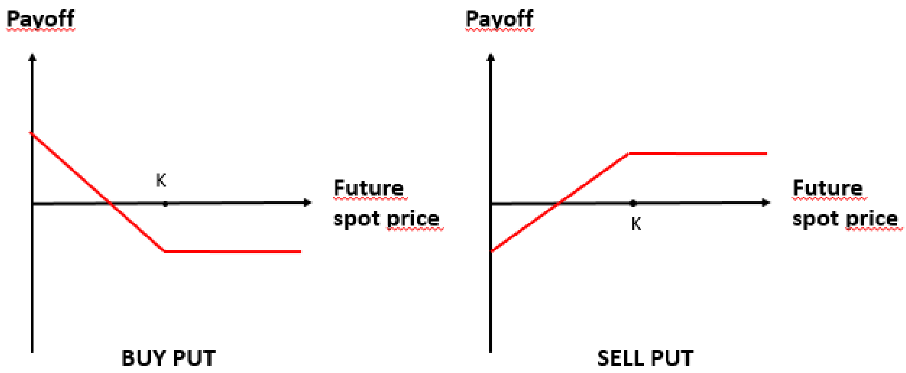


Figure 14: Representation of the payoff of a put option

Spreads

Options may be contracted in order to hedge for the difference in the price or spread between two products. This is the case for spread between countries or between different commodities. The spark spread refers to the difference in prices between electricity and gas for a unit of energy, typically MWh. The dark spread measures the difference between electricity and coal prices, while the quark spread is the difference between electricity and nuclear fuel prices. If the spreads include the costs of the CO2 emissions they are the referred also as “clean” spreads, while are denoted as “dirty” when they omit them.

Value of the option

The option value is the sum of its intrinsic value and its time value. For a call option, the intrinsic value is the difference between the price of the underlying spot price and the strike price, at the moment when the option is bought. Depending on this difference, options are classified as:

Type of option	Characteristic
Out-of-the-money option	$K > S_t$
At-the-money option	$K \approx S_t$
In-the-money option	$K < S_t$

Table 2: Types of option according to the relative price of the strike to the underlying asset

<u>Put-call parity</u>

One important relation between the call and the put prices is the put-call parity, which is given in an arbitrage-free market by the relation:

$$C - P = S(t) - Ke^{-r(T-t)} = e^{-r(T-t)}(F - K)$$

If an agent buys a call option and sells a put option, at $T = t$, if $S(T) > K$, the call option will be exercised and the will have a payoff of $S(T) - K$, while the put option will not be exercised. If $S(T) < K$ it will be the put option which will be exercised with a payoff of $S(T) - K$, while the call option will not be exercised. The added profit resulting from both options is equivalent to what a forward contract would have delivered. (Zhang, 2009)

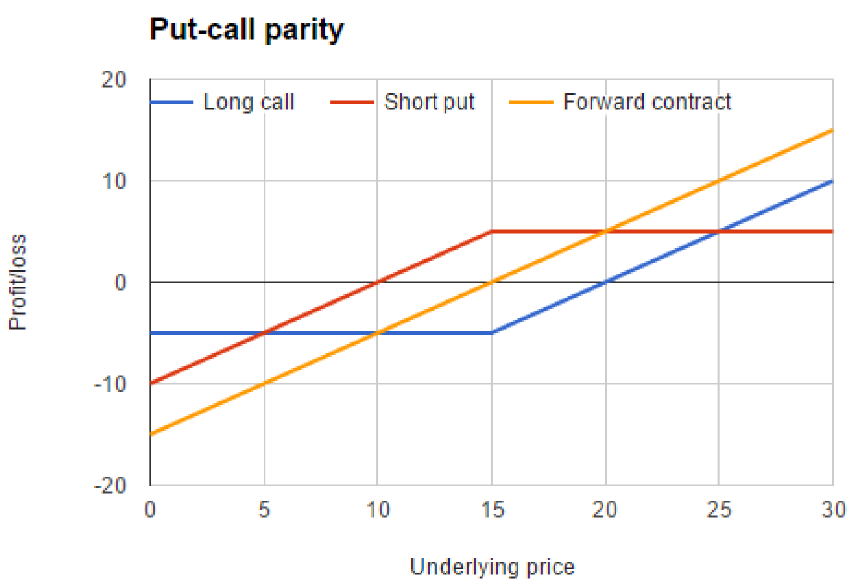


Figure 15: Representation of the put-call parity (Source: Investopedia)

Types of options

Depending on when they are exercised, there are three basic types of vanilla options:

- European options: can only be executed only at the expiration date of the option, which c
- American options: they can be executed at any time during a time window.
- Asian options: they are executed after delivery, on the average of the prices of the underlying asset during a time period.
- Bermudan options: can only be executed at certain dates.

3.4 Valuation of European Options

The most extended approach to valuate European options was proposed in 1973 by Fisher Black and Myron Scholes. It supposes the underlying asset follows a Geometric Brownian Motion (developed in section 3.5.1). The model makes several assumptions: there are no arbitrage opportunities, the riskless assets are supposed to earn a known and constant risk-free interest rate and markets are perfectly efficient. It also implies that there are no transaction costs to acquire the option, and that the option does not pay dividends. (Colín Betancourt, 2014) (Burger, 2014)

The payoffs at the maturity for the call *C* and put *P* options are:

$$C = \max(S_T - K, 0)$$

$$P = \max(K - S_T, 0)$$

By the risk-neutral world assumption, the option price at its time of purchase is the present value of the expectation of payoff.

$$C = \mathrm{e}^{-\mathrm{r}(\mathrm{T}-\mathrm{t})}\ \mathbb{E}_\mathrm{T}\ [\max(S_T - K, 0)]$$

$$P = \mathrm{e}^{-\mathrm{r}(\mathrm{T}-\mathrm{t})}\ \mathbb{E}_\mathrm{T}\ [\max(S_T - K, 0)]$$

The expectation of the payoff is defined by:

$$\mathbb{E}_\mathrm{T}\ [\max(S_T - K, 0)] = \int_0^\infty \max(S_T - K)$$

The Black-Scholes price of the European call option is given by:

$$C = S\,N(d_1) - K\,e^{-r(T-t)}N(d_2)$$

$$P = -S\,N(-d_1) + K\,e^{-r(T-t)}N(-d_2)$$

Where

$$d_1 = \frac{\ln\frac{S}{K} + \left(r - \frac{1}{2}\sigma^2\right)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = d1 - \sigma\sqrt{T-t}$$

3.5 Valuation of Asian Options

The payoff in the Asian options, contrary to the European, is not function of the price of the underlying option at a specific date, but to its average price during a certain time period.

The call option or cap will be exercised if the average price during the period is larger than the exercise price of the option: being the payoff:

$$C = \max[A(T) - K, 0]$$

Alternatively, the put option or floor will be exercised if the strike of the option is larger than the average price during the period.

$$P = \max[K - A(T), 0]$$

Where A_t is the arithmetic average price:

$$A(T) = \frac{1}{N}\sum_{i=1}^{N} S_{t_i}$$

This characteristic leads to the Asian option having typically lower volatilities than European or American options (Zhang). Another characteristic which favors Asian option is the fact that it reduces the incentives to price-makers to alter prices by exercising speculative actions, contrary to American options.

Asian options are path-dependent, thus the payoff depends on the path of spot prices during the option period. (Wiklund, 2012).

Many methods have been proposed to value Asian options. They can be classified in Monte Carlo techniques or approximating expressions:

	Advantages	Disadvantages
Monte Carlo	- Path dependent: suitable to compute the value of Asian options. - Adequate to value complex products, for which it is difficult to find approximating formulas. - Assumptions are easy to understand. - Flexible: Can be equipped with different spot price models.	- High computational time - High programming effort - Dependence on the spot price model utilized.
Approximating methods (Lévy, Vorst, etc.)	- Provide a suitable approximation of the payoff of an Asian option. - Low computational time - Lower programming effort (some exceptions)	- Rely heavily on assumptions which may be difficult to understand.

Table 3: Techniques to estimate the payoffs of Asian option

3.5.1 Monte Carlo methods

Monte Carlo methods are a range of computational algorithms which are used to solve stochastic problems. Being originally used by the casinos placed in the gambling resort which gives name to the technique to estimate their probable profits or losses in each game, these methods are currently being used in nearly any complex problem where randomness intervenes in fields as diverse as physics, biology, engineering or finance. It is very suitable for problems which cannot be solved with other techniques, for example some non-linear problems. (Murthy, 2003)

Monte Carlo simulation is one of the most common methods to estimate the fair price of Asian options. One of the biggest advantages of using this technique is that it provides a path-dependent payoff, as the Asian options themselves have. The main drawback of the technique is the great computational effort required to obtain accurate estimations.

The idea behind any Monte Carlo algorithm is to make a large number of independent random paths using a known distribution, and finally obtain the results by giving each sample the same probability and averaging the results.

When applied to an Asian option, Monte Carlo simulations relies greatly on the spot price model utilized. There are many different price generators to model the spot energy markets. In this thesis we will focus on three: Geometric Brownian Motion, Vasicek and ARIMA. A path of prices is obtained from the spot price model and a random number generator. Usually computer programs are equipped with pseudo-random number generators which simulate very well random numbers. Indeed, the reason why they are called pseudo-random numbers is because although they are obtained by deterministic algorithms, the obtained sequences of pseudo-random generators pass the randomness tests. (Biebighauser)

The payoff of an Asian option is calculated in one path by the already mentioned expressions:

$$\hat{\theta}_C = \max[A(T) - K, 0] \; ; \; \hat{\theta}_P = \max[K - A(T), 0]$$

The Monte Carlo method is more precise the more price paths are generated. The average payoff is computed from the payoffs of the individual paths.

Finally, the price of the option is calculated as the present value of the expected payoff, at the moment of purchase.

$$\text{Price} = \text{PV}(\hat{\theta}) = \mathrm{e}^{-\mathrm{rT}}\, \hat{\theta}$$

Where *t* is the time to maturity

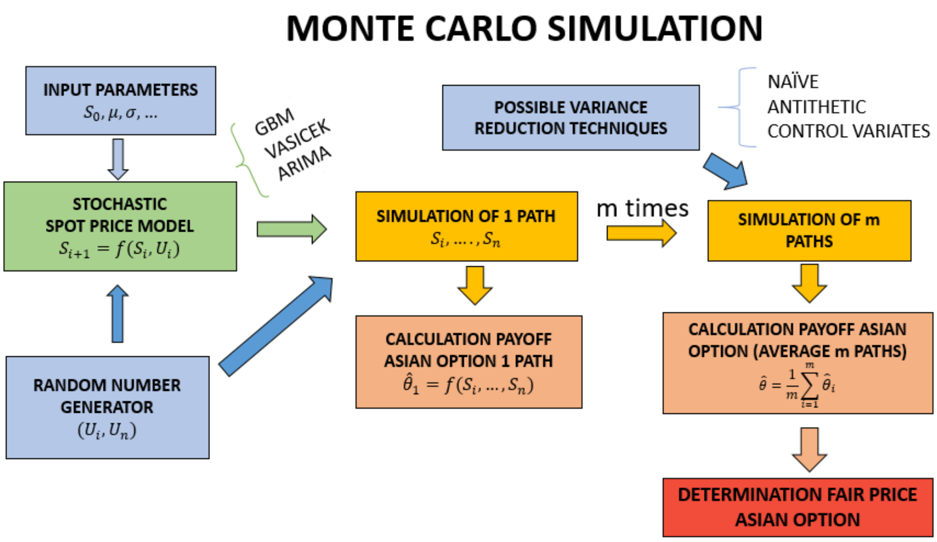


Figure 16: Flowchart of the Monte Carlo simulation methods

3.5.3.1. Monte Carlo techniques (variance reduction)

Naïve Monte Carlo (Zhang, 2009)

The Naïve Monte Carlo method is the simplest technique to perform a Monte Carlo simulation. By this approach, no variance reduction techniques are applied to accelerate the model. If a variable θ is function of independent random observations X:

$$\theta = \mathbb{E}[f(X)]$$

When valuating Asian options, the searched variable θ is the payoff of the option and X is the spot price S.

For every path, the calculation of the payoff θ is performed. If a large number of paths are simulated, then the estimator $\hat{\theta}$ of the payoff can be obtained, as the average of the payoffs of each path.

$$\hat{\theta} = \frac{1}{n}\sum_{i=1}^{n} \hat{\theta}_i$$

The estimator $\hat{\theta}$ is unbiased, as its mathematical esperance is the actual parameter θ.

$$\mathbb{E}[\hat{\theta}] = \mathbb{E}\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right] = \frac{1}{n}\sum_{i=1}^{n} \mathbb{E}[x_i] = \frac{1}{n} n\hat{\theta} = \theta$$

The expression of the unbiased estimator of the variance of θ is:

$$s = Var(\hat{\theta}) = \frac{1}{n-1}\sum_{i=1}^{n} (x_i - \hat{\theta})^2$$

By the central limit theorem, if a large number of simulations (n) is made, $\theta - \hat{\theta}$ tends to a normal distribution. Its confidence interval at an α confidence level is given by:

$$\left[\hat{\theta} - z_\alpha \frac{s}{\sqrt{n}}, \hat{\theta} + z_\alpha \frac{s}{n}\right]$$

The concern when using Monte Carlo simulation (especially the Naïve method) to price option is that accurate estimates can very time consuming to obtain. In order to reduce the computational time, several variance reduction techniques have been proposed in recent years, such as Antithetic variables, control variates or stratified. (Wiklund, 2012)

Antithetic technique

The Antithetic MC is a variance-reduction method which consists in taking for every path generated by the random numbers $(U_1, U_2, \dots, U_m)$ a path given by the random numbers $(1 - U_1, 1 - U_2, \dots, 1 - U_m)$.

The estimated payoff $\hat{\theta}$ will be therefore: (Wiklund, 2012)

$$\hat{\theta} = \frac{f(U_1, U_2, \dots, U_\mathrm{m}) + f(1 - U_1, 1 - U_2, \dots, 1 - U_\mathrm{m})}{2} = \frac{\hat{\theta}_1 + \hat{\theta}_2}{2}$$

The aim of this technique is to reduce the variance of the estimate, which is interesting as it allows to reduce the confidence interval. Being $\hat{\theta}_1$ the estimator for the randomly generated paths and $\hat{\theta}_2$ the estimator for the antithetic paths, the variance of the estimate $\hat{\theta}$ (payoff) will be:

$$Var\left(\hat{\theta}\right) = \frac{Var\left(\hat{\theta}_1\right) + Var\left(\hat{\theta}_2\right) + 2\, Cov(\hat{\theta}_1, \hat{\theta}_2)}{2}$$

The method is sustained by the hypothesis that $Cov\left(\hat{\theta}_1, \hat{\theta}_2\right) \leq 0$, as $Var\left(\hat{\theta}_1\right) \approx Var\left(\hat{\theta}_2\right)$. Therefore

$$Var\left(\hat{\theta}\right) \leq \frac{Var\left(\hat{\theta}_1\right) + Var\left(\hat{\theta}_2\right)}{2}$$

3.5.3.2 Spot price generators

The central element of the Monte Carlo simulation method is the spot price generator. Several spot price models common in option valuation are reviewed: Geometric Brownian Motion, Vasicek and ARIMA. Tools to make estimations of the parameters of the first three models have been developed, as well as to generate spot prices from them.

Geometric Brownian Motion

One of the simplest stochastic processes used to model spot energy prices is the Geometric Brownian Motion. In this process, the logarithm of the variable follows a Brownian motion (or Wiener Process), existent in Nature in collision between fast-moving atoms and molecules in liquids and gas.

The stochastic process S_t follows a Geometric Brownian Motion if it satisfies the following expression:

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

- μ is commonly referred as drift and denotes the continuously compounded expected return.
- σ is the volatility of the process.
- W_t is the Wiener process. It is characterized for having independent Gaussian increments with W has Gaussian increments with mean 0 and variance 1 $(dW_t \sim N(0,1))$ and continuous paths.

  The Wiener process is commonly characterized in discretized processes as $dW_t = \epsilon\sqrt{t}$, where ϵ is a random normal variable.

The impact of the drift of the GBM is assessed in the figure below:

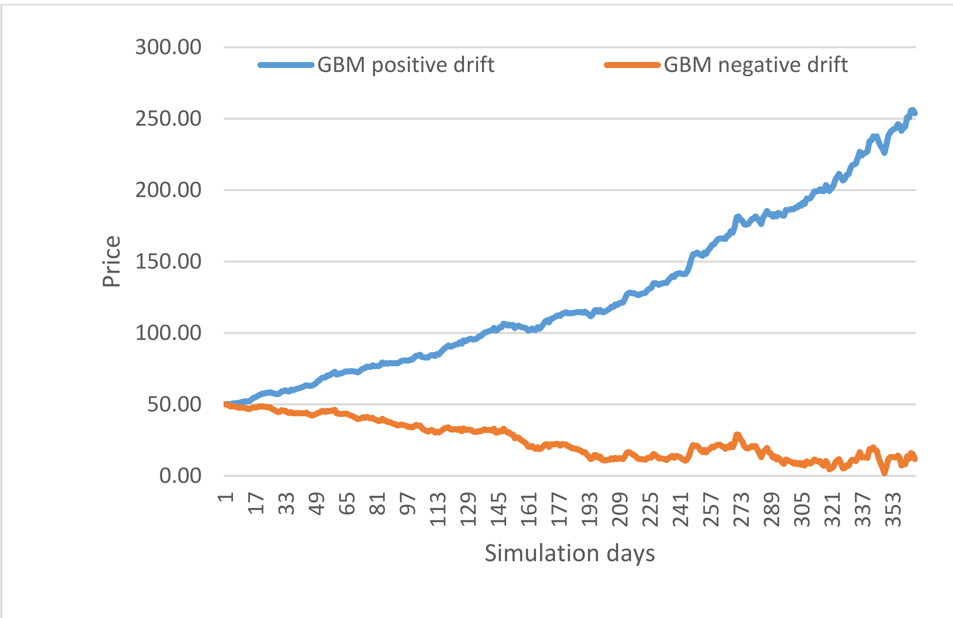


Figure 17: Representation of GBM with positive and negative drifts

Obtention of a closed form

If a stochastic variable X follows an Îto process: (Wiklund, 2012)

$$dX = a(X,t)dt + b(X,t)dz$$

A function $f(X,t)$ follows the process

$$df(X,t) = \left(\frac{\partial f}{\partial X}a(X,t) + \frac{\partial f}{dt} + \frac{1}{2}\frac{\partial^2 f}{dX^2}b^2(X,t)\right)dt + \frac{\partial f}{\partial X}b\ dz$$

Applying this to the Geometric Brownian Motion process for the spot price, with $f = \ln(S_t)$

$$dln(S_t) = \left(\frac{1}{S_t}\mu\, S_t\, dt + \frac{1}{2}\left(-\frac{1}{S^2}\right)\sigma^2\, S_t^2\right)dt + \frac{1}{S_t}\sigma S dz = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma\ dz$$

Discretizing, $dt = \Delta t$ is applied:

$$\Delta ln(S_t) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\,\epsilon\,\sqrt{\Delta t}$$

This results in:

$$ln(S_{t+\Delta t}) - \ln(S_t) = \ln\left(\frac{S_{t+\Delta t}}{S_t}\right) = \left(\mu - \frac{\sigma^2}{2}\right)\Delta t + \sigma\,\epsilon\,\sqrt{\Delta t}$$

Which results in the closed-form expression:

$$S_t = S_0 \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\,\epsilon\,\sqrt{\Delta t}\right]$$

The main problems of the Geometric Brownian Motion (Johnson and Barz 1999, Steele 2010) are that it does not allow to represent mean-reversive processes characteristic in energy markets or seasonal changes. There can also be price spikes as well as too much dependence between price and volatility.

If the mean of the spot price is considered to be constant, then μ can be considered to be 0 in the long-term. This is a common assumption in energy markets. The volatility σ is commonly estimated as the standard deviation of the logarithm of the spot price returns. (Burger, 2014)

$$x_i = \frac{S_i - S_{i-1}}{S_i}$$

$$r_i = \ln(S_i) - \ln(S_{i-1})$$

$$\hat{\sigma} = \frac{\sqrt{\frac{1}{N-1}\sum_1^n (r_i - \bar{r})^2}}{\sqrt{t_i - t_{i-1}}}$$

$$\hat{\mu} = \frac{\bar{r}}{t_i - t_{i-1}} + \frac{\hat{\sigma}^2}{2}$$

Ornstein-Uhlenbeck processes

The Ornstein-Uhlenbeck process is also a physical process which describes the one-dimensional movement of a Brownian particle under friction. They can be more suitable to model power and gas spot prices as they consider mean reversion. There is a drift towards the long-term mean, more intense the higher the deviation to the mean is.

- **Vasicek:** $dS_t = \lambda(\mu - S_t)\, dt + \sigma\, dW_t$
- **Cox-Ingersoll-Ross:** $dS_t = \lambda(\mu - S_t)\, dt + \sigma\sqrt{S_t}\, dW_t$

The Vasicek approach assumes a constant volatility while the Cox-Ingersoll-Ross approach considers the volatility dependent to the spot price.

Vasicek model

The Vasicek model is described by the expression

$$dS_t = \lambda(\mu - S_t)\, dt + \sigma\, dW_t$$

Where $\lambda > 0$, $\mu > 0$ and $\sigma > 0$ is the volatility.

λ is the mean reversion rate, μ is the long-term mean and $\sigma > 0$ is the volatility.

There are two approaches for parameter estimation: the Least Squares method and the maximum likelihood. (van den Berg, 2011)

- Least Squares method: it is based on the assumption that there is a linear regression model can be obtained between consecutive observations S_{i+1} and S_i:

$$S_{i+1} = a\, S_i + b + \epsilon$$

The parameters of the models can be easily obtained: (van den Berg, 2011)

$$a = \frac{n \sum_{i=1}^{n} S_i\, S_{i-1} - \sum_{i=1}^{n} S_{i-1} \sum_{i=1}^{n} S_i}{n \sum_{i=1}^{N} S_{i-1}^2 - (\sum_{i=1}^{N} S_i)^2}$$

$$b = \frac{\sum_{i=1}^{n} S_i - a \sum_{i=1}^{n} S_{i-1}}{n}$$

$$sd(\epsilon) = \sqrt{\frac{\sum_{i=1}^{n} S_i^2 - (\sum_{i=1}^{n} S_i)^2 - a \sum_{i=1}^{n} S_{i-1} \sum_{i=1}^{n} S_i}{n(n-2)}}$$

In MS Excel a and b can be obtained with the slope and intersection commands, respectively. For a set of data contained from cells A2 to A100:

a=SLOPE(A3:A100;A2:A99)

b=INTERSECTION(A3:A100;A2:A99)

The parameters of the Vasicek model can be obtained: (van den Berg, 2011)

$$\lambda = -\frac{\ln(a)}{\Delta t}$$

$$\mu = \frac{b}{1-a}$$

$$\sigma = sd(\epsilon) \sqrt{\frac{-2\ln(a)}{\Delta t(1-a^2)}}$$

- Maximum likelihood: in this model, the parameters are obtained by maximizing the likelihood function of the set of observations $(S_0, S_1, \ldots, S_n)$: (van den Berg, 2011)

$$\mathcal{L}(\mu,\lambda,\sigma) = \sum_{i=1}^{n} \ln f(S_i, S_{i-1}; \mu, \lambda, \sigma\,)$$

$$= -\frac{n}{2}\ln(2\pi) - n\ln(\hat{\sigma}) - \frac{1}{2\hat{\sigma}^2}\sum_{i=1}^{n}\left[S_i - S_{i-1}e^{-\lambda\,\Delta t} - \mu\left(1 - e^{-\lambda\,\Delta t}\right)\right]^2$$

The maximum of the function is obtained by making the partial derivatives equal to zero:

$$\frac{\delta\mathcal{L}(\mu,\lambda,\sigma)}{\delta\mu} = \frac{1}{\hat{\sigma}^2}\sum_{i=1}^{n}\left[S_i - S_{i-1}e^{-\lambda\,\Delta t} - \mu\left(1 - e^{-\lambda\,\Delta t}\right)\right]^2 = 0$$

$$\frac{\delta\mathcal{L}(\mu,\lambda,\sigma)}{\delta\lambda} = -\frac{\Delta t e^{-\lambda\,\Delta t}}{\hat{\sigma}^2}\sum_{i=1}^{n}(S_i - \mu)(S_{i-1} - \mu) - e^{-\lambda\,\Delta t}(S_{i-1} - \mu)^2 = 0$$

$$\frac{\delta\mathcal{L}(\mu,\lambda,\sigma)}{\delta\hat{\sigma}} = \frac{n}{\hat{\sigma}} - \frac{1}{\hat{\sigma}^3}\sum_{i=1}^{n}[(S_i - \mu - e^{-\lambda\,\Delta t}\,(S_{i-1} - \mu)]^2 = 0$$

Obtaining:

$$\hat{\mu} = \frac{\sum_{i=1}^{n} S_i - S_{i-1}e^{-\hat{\lambda}\,\Delta t}}{n(1 - e^{-\lambda\,\Delta t})}$$

$$\lambda = -\frac{1}{\Delta t}\ln\left(\frac{\sum_{i=1}^{n}(S_i - \hat{\mu})(S_{i-1} - \hat{\mu})}{\sum_{i=1}^{n}(S_i - \hat{\mu})^2}\right)$$

$$\hat{\sigma}^2 = \frac{1}{n}\sum_{i=1}^{n}[(S_i - \hat{\mu} - e^{-\lambda\,\Delta t}\,(S_{i-1} - \hat{\mu})]^2$$

Cox-Ingersoll-Ross

The Cox-Ingersoll-Ross (CIR) model is described by the expression

$$dS_t = \lambda(\mu - S_t) + \sigma\sqrt{S_t}\,dW_t$$

λ is the mean reversion rate, μ is the long-term mean and $\sigma > 0$ is the volatility.

The CIR process is one of few cases, among the diffusion processes, where the transition density has a closed form expression. (Kladívko) (Chou, 2006)

$$p(S_{t+\Delta t}|S_t;\mu,\lambda,\sigma) = c\,e^{-u-v}\left(\frac{v}{u}\right)^{\frac{q}{2}} I_q\left(2\sqrt{uv}\right)$$

Where

$$c = \frac{2\lambda}{\sigma^2(1 - e^{-\lambda\Delta t})}$$

$$u = cS_t e^{-\lambda\Delta t}$$

$$v = cS_{t+\Delta t}$$

$$q = \frac{2\lambda\mu}{\sigma^2} - 1$$

And $I_q\left(2\sqrt{uv}\right)$ is the modified Bessel function of the first kind and order q.

By the Ordinary Least Squares approach suggested by Kladívko, the estimators of the CIR of model are:

$$\hat{\lambda} = \frac{N^2 - 2N + 1 + \sum_{i=1}^{N-1} S_{t+\Delta t} \sum_{i=1}^{N-1} \frac{1}{S_t} - \sum_{i=1}^{N-1} S_t \sum_{i=1}^{N-1} \frac{1}{S_t} - (N-1)\sum_{i=1}^{N-1} \frac{S_{t+\Delta t}}{S_t}}{\left(N^2 - 2N + 1 - \sum_{i=1}^{N-1} S_t \sum_{i=1}^{N-1} \frac{1}{S_t}\right)\Delta t}$$

$$\hat{\mu} = \frac{(N-1)\sum_{i=1}^{N-1} S_{t+\Delta t} - \sum_{i=1}^{N-1} \frac{S_{t+\Delta t}}{S_t} \sum_{i=1}^{N-1} S_t}{N^2 - 2N + 1 + \sum_{i=1}^{N-1} S_{t+\Delta t} \sum_{i=1}^{N-1} \frac{1}{S_t} - \sum_{i=1}^{N-1} S_t \sum_{i=1}^{N-1} \frac{1}{S_t} - (N-1)\sum_{i=1}^{N-1} \frac{S_{t+\Delta t}}{S_t}}$$

$\hat{\sigma}$ is found as the standard deviation of residuals.

ARIMA

The ARIMA (Autoregressive Integrated Moving Average) model is a widely used method to approximate and forecast time series.

An ARIMA (p,d,q) process is described by the expression:

$$\left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1 - L)^d X_t = \left(1 - \sum_{i=1}^{q} \phi_i L^i\right)\varepsilon_t$$

Where L is the lag operator:

$$Ly_t = y_{t-1}$$

The I for integrated indicates the number of differences applied to the original time series. A first order difference is expressed as:

$$y_t' = y_t - y_{t-1}$$

The AR(p) term indicates that the output depends on a linear combination of its previous *p* values. The MA(q) term, indicates that the output depend on the linearly on current and *q* past values of a white-noise variable.

3.5.2 Vorst model

The (Vorst, 1992) approximation model assumes that spot prices follow a GBM of drift μ and volatility σ. With these parameters as well as the number of days of the averaging period of the Asian option n, the duration of the averaging period T, and the first price S_0, Vorst provides an estimation for the payoff of the option.

The approximation takes advantage of the fact that the geometric average $G(T)$ is always lesser than the arithmetic average $A(T)$, providing a lower bound for the pricing of the option. Vorst also manages to obtain an upper bound for the option referred to the geometric option price, and provides an approximation of the average price by computing the price of the Asian geometric option price on an option with a an adjusted strike K'. (Burger, 2014)

The geometric average $G(T)$:

$$G(T) = \sqrt[n]{\prod_{i=1}^{n} S_{t_i}}$$

The adjusted strike K' of the option is given by:

$$K' = K - (\mathbb{E}[\mathrm{A(T)}] - \mathbb{E}[\mathrm{G(T)}])$$

Supposing S_t follows a GBM, the geometric mean is log-normally distributed, as it has independent increments: (Burger, 2014)

$$ln\ G = \frac{1}{n}\sum_{i=1}^{n} \ln S_{t_i} = \ln S_0 + \sum_{i=1}^{n}\left(\ln S_{t_i} - \ln S_{t_{i-1}}\right) = \ln S_0 + \sum_{i=1}^{n} ln\frac{S_{t_i}}{S_0}$$

Under the assumption that the differences of the instants $t_0, t_1, \ldots, t_n$ are uniform, Nielsen obtains a closed form for the mean and the variance of the logarithm of the geometric mean. (Nielsen, 2011)

$$\mu_G = \ln S_0 + \left(r - \frac{1}{2}\sigma^2\right)\frac{T+h}{2}$$

$$Var(G) = \sigma^2(h + (T-h)\frac{2n-1}{6n}$$

Being $h = \frac{T}{m}$

Vorst solution to price the Asian Geometric option is given by:

$$C_G = e^{-rT} \left[e^{\mu_G + \frac{1}{2} Var(G)} N(d_1) - K\, N(d_2) \right]$$

Where

$$d_1 = \frac{\mu_G - \ln K + Var(G)}{\sqrt{Var\ (G)}}$$

$$d_2 = x - \sqrt{Var\ (G)}$$

By substituting K by K', the Vorst approximation for the arithmetic option is obtained.

Upper and lower bounds

Vorst obtains a lower and an upper bound for the pricing of the option. Given that the arithmetic average is larger than the geometric average, the arithmetic option price will always be larger than or equal to the geometric option price. (Nielsen, 2011)

The arithmetic option price is given by:

$$C = \exp(-rT)\, \mathbb{E}[\max(\mathrm{A} - \mathrm{K}, 0)]$$

The geometric option price, or lower bound of the arithmetic option price, is given by:

$$\underline{C} = C_G = \exp(-rT)\, \mathbb{E}[\max(\mathrm{G} - \mathrm{K}, 0)]$$

The lower bound can be obtained from the inequality (Nielsen, 2011):

$$\max(\mathrm{A} - \mathrm{K}, 0) = \max(G - K, G - A) + A - G \leq \max(G - K, 0) + A - G$$

And leads to the upper bound:

$$\overline{C} = \exp(-rT)\, \mathbb{E}[\max(G - K, 0) + A - G] = \underline{C} + e^{-rt}\, (\mathbb{E}[\mathrm{A}] - \mathbb{E}[\mathrm{G}])$$

Therefore the price of the arithmetic option will be bounded: $\underline{C} \leq C \leq \overline{C}$

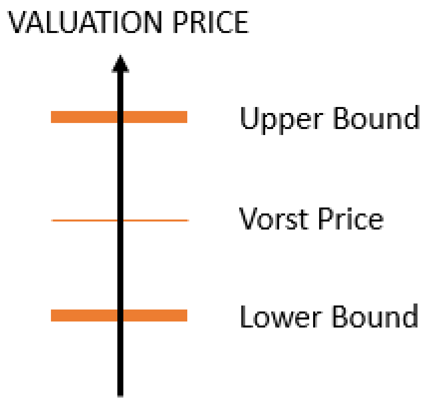


Figure 18: Vorst valuation price of an Asian option

Vorst's suggestion is a price located between both bounds. However any other price respecting this criteria could also be proposed. As it has been mentioned, the fair value proposed by Vorst is:

$$\mathrm{C} = e^{-rT}\left[e^{\mu_G+\frac{1}{2}Var(G)}\, N(d_1) - K'\, N(d_2)\right]$$

With

$$d_1 = \frac{\mu_G - \ln K' + Var(G)}{\sqrt{Var\,(G)}}$$

$$d_2 = x - \sqrt{Var\,(G)}$$

$$K' = K - (\mathbb{E}[\mathrm{A(T)}] - \mathbb{E}[\mathrm{G(T)}])$$

Application

Vorst option valuation expression is straightforward once the spot prices are characterized as a GBM with drift μ and volatility σ (annualized) with iterations every h years. Generally h =1/365 as average daily prices are computed. The only other inputs needed is the period of the option.

To assess the value of the put, the put-call parity is employed:

$$C - P = e^{-r(T-t)}(A(t) - K)$$

4. METHODOLOGY

4.1 Software developed

4.1.1 Selection of the support and implications

The selection of VBA/Excel as the support of the application is explained by the need of the utility of having an application which can be simply used by non-experts, and the visibility of its results.

Nevertheless, the choice of MS Excel has complicated greatly the development of the tool, as this software is not the optimal to work with numerical vectors. In fact, to compute easy operations such as the mean, standard deviations or an α percentile of an array, it is necessary to either print all the numbers in the Excel worksheet and apply the predefined functions, or manufacture all the functions in the VBA code, a task which can be really arduous.

The choice of the author was to rely as much as possible on the functions implemented in Excel, for which it was necessary to print all the numbers in the Excel worksheets. This, besides complicating the results calculation due to having to search for the desired data between the rows and columns, had severe implications in computational time and in the number of price paths that the model could generate. Each path is placed in a column and Excel v. 7.0 sheets have a maximum of 16384 columns.

As having this number of price paths is not enough to carry out precise calculations, the first approach to solve this was to implement an Antithetic Monte Carlo, in order to reduce the variance of the payoffs and be able to reduce the confidence intervals. As this proved to be insufficient, a generalization of the tool had to be made to several sheets, so that the price paths are allocated in a different sheet once the previous one is completed, and then the results take into account all the sheets. With this approach, computations of hundreds of thousands of simulations can be made, in a quite acceptable time.

The price generators needed for the Monte Carlo model, GBM and Vasicek, were also developed with VBA/Excel. The sole exception were the ARIMA models, which were produced using *R* software, however their results are easily integrated in the Excel package.

4.1.2 Inputs

A Monte Carlo evaluation tool has been implemented in the platform Excel/VBA. The user can enter several inputs.

- Start date.
- Simulation days. The final date is obtained from this number.
- Number of trials.
- Monte Carlo technique: A choice can be made between GBM or Vasicek model.

Input Data Monte Carlo	
Today	01/10/2015
Sim. Days	93
number trials	7500
Spot price generator	Geometric Brownian
Final date	01/01/2016
Simulation method	Antithetic Monte Carlo

Figure 19: Input data for the Monte Carlo tool

The parameters of the spot price model are also inputs in the software: Drift (μ), volatility (σ) for both models, long-term mean (μ) for the Vasicek model, and reversion speed (λ) for the Vasicek model. There are parallel applications in order to estimate the parameters of the GBM and the Vasicek model, the latter with either Least Squares or Maximum Likelihood techniques.

Input spot price generator	
Drift (GBM)	0.05
Volatility (GBM, Vasicek)	0.6798
Ref price - Mean (Vasicek)	18.67
Lambda (Vasicek)	205.94

Figure 20: Input data for the spot price generator integrated in the Monte Carlo tool

The valuation of the Asian options can be performed at the strike K that the user prefers.

Valuation Asian options	
K	50
Valuation period (days)	Month

Figure 21: Input data for the Asian option valuation

The main advantage of this software is that it allows to compute daily, weekly, monthly and yearly average prices, from where caps, floors and futures can be valuated. This allows the utility to valuate contracts in the long term.

Weekday	Date	Spot price generator PATH1		
lu.	**01/01/2018**	17.77933228	18.01038277	
ma.	**02/01/2018**	17.79397603	18.22432143	
mi.	**03/01/2018**	17.83041679	17.92303892	
ju.	**04/01/2018**	17.72621348	18.23251344	
vi.	**05/01/2018**	17.62923542	18.16526722	
sá.	**06/01/2018**	17.76806615	18.17613676	
do.	**07/01/2018**	17.36316385	18.31444594	
lu.	**08/01/2018**	17.49148949	18.00703344	
ma.	**09/01/2018**	17.6550945	18.05018491	
mi.	**10/01/2018**	17.75617393	18.03436245	
ju.	**11/01/2018**	17.44353496	17.93365617	
vi.	**12/01/2018**	17.40296713	17.99144791	
sá.	**13/01/2018**	17.60211755	18.0620661	
do.	**14/01/2018**	17.75068074	17.94203454	
lu.	**15/01/2018**	17.68988426	17.83155297	
				Tot. Avg Path 1
	Average Path 1	17.64548977	18.05989633	**17.85269305**
	Asian option payoff Path 1	1.645489771	2.059896331	**1.852693051**
	Week 1 Avg	17.69862914	18.14944378	**17.92403646**
	Week 2 Avg	17.58600833	18.00296936	**17.79448884**
	Week 1 Asian Payoff	1.698629144	2.149443782	**1.924036463**
	Week 2 Asian Payoff	1.586008327	2.00296936	**1.794488844**

	Spot price generator PATH2		
	17.57929608	17.353776	
	17.5629042	17.148177	
	17.5250895	17.434524	
	17.62617877	17.136716	
	17.72119786	17.19827	
	17.58080656	17.186102	
	17.98881271	17.054445	
	17.85488168	17.343694	
	17.68748684	17.300335	
	17.5848712	17.313616	
	17.89808178	17.408933	
	17.93783788	17.351111	
	17.73294567	17.281378	
	17.58260395	17.395084	
	17.64109835	17.500943	
			Tot. Avg Path 2
Average Path 2	17.70027287	17.293807	**17.497**
Asian option payoff Path	1.700272868	1.2938068	**1.49704**
Week 1 Avg	17.65489795	17.216001	**17.4354**
Week 2 Avg	17.75410129	17.342022	**17.5481**
Week 1 Asian Payoff	1.654897953	1.2160013	**1.43545**
Week 2 Asian Payoff	1.754101286	1.3420215	**1.54806**

		SINGLE PATH		
	Asian option payoff Whole period	0.822744885	1.029948166	**0.926346525**
	Week 1 Asian Payoff	1.676763548	1.682722564	**1.679743056**
	Week 2 Asian Payoff	1.670054807	1.672495434	**1.67127512**
Std payoff	**0.313561745**			
LBound ConfInt	0.619056015			
UBound ConfInt	1.233637036			

Figure 22: Results of the Monte Carlo simulator (Antithetic version)

With the results from all the simulated paths, risk metrics such as VaR and CVaR can be computed. To assess the results of the model in option valuation, other valuation tools have been implemented such as Black-Scholes and Vorst approximation.

4.1.3 Option valuation

The objective of this section is to evaluate the price of Asian options in real situations in the Spanish market, by means of Monte Carlo simulations relying on different spot price generators: GBM, Vasicek and ARIMA models are utilized, and their results are compared with other the Vorst option valuation methods for Asian options such.

The aim is to provide a valuation tool for periods where future market behavior is considered to have resemblances with the periods where the models are estimated. This excludes from this study periods when prices do not show relatively normal behavior, both in values and volatility, where expected prices could be better predicting by understanding the fundamentals models of the market rather than by reproducing historical behavior. Due to the non-standard behavior of the market prices during the years 2016, and the beginning of 2017, year 2015 is selected as the benchmark for the option valuation.

For the valuation of options, the following approach steps are followed:

1. For each spot price model, its parameters are estimated during a time period, the training set.
2. A Monte Carlo simulation is performed simulating the Asian option payoffs for different strike prices. These payoffs are compared to the actual payoffs during that period to observe how well models behave with respect to reality. The payoffs of the Asian option are important as they determine the fair price of buying/selling that Asian option during that period.
3. A Monte Carlo simulation is performed on a future time window determining the fair value of the Asian option according to the spot price model used before its start. The aim is to simulate what would be the price to pay for an Asian option in the determined period.

Although the developed approach is mainly based on historical behavior, it is known that the price of any derivative depends on future expectations. To address this, the future price of the valuation period before its beginning is considered, and is used as the start price in the GBM, as the start price and long-term mean in the Vasicek model, and to correct the estimates afterwards in the ARIMA model.

It is remarked that, in order to obtain good results, there has to be certain similarity in the behavior of the prices between the training period and the valuation of period, as the models are primarily based on historical performance of the spot market. This is an analysis that traders must make when using historical prices to estimate the prices in the future: they need to understand what historical period is better to assess future market behavior.

5. RESULTS

5.1 Comparison Antithetic – Naïve Monte Carlo performance

A same MC experiment has been designed to assess the performance of the two methods implemented. The objective is to obtain the price of the Asian option of the next month, the day before its start. The chosen spot price generator is GBM and the values of the parameters are set as follows: valuation time T=30 days, strike price K=25 €/MWh, start price $S_0 = 23\ €/MWh$, $\sigma = 20\%$. The simulation is performed for a total N number of paths:

	NAÏVE MC		**ANTITHETIC MC**	
N	**Average [Confidence interval 95%] (€/MWh)**	**Std deviation payoff (€/MWh)**	**Average [Confidence interval 95%] (€/MWh)**	**Std deviation payoff (€/MWh)**
100	3.06 [2.92, 3.21]	0.72	3.00 [2.85, 3.14]	0.73
1000	2.96 [2.91, 3.01]	0.78	3.00 [2.95, 3.04]	0.79
10000	3.00 [2.99, 3.02]	0.77	3.00 [2.99, 3.01]	0.77

Table 4: Comparison of average and standard deviation of the payoff of an option with Naïve and Antithetic MC methods

It is observed that, contrary to what was believed (see section 3.5.1), the standard deviation of the payoffs does not decrease when using the antithetic technique. However, an improvement in the computational time was observed, therefore the Antithetic Monte Carlo was the model further developed, and used in the rest of the document.

Computational time can be one of the biggest difficulties when performing Monte Carlo Simulation. In VBA there are four commands which can speed up the simulations. They have to be added at the beginning and at the end of the code.

```
Application.Display.Alerts = False
Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual
```

```
Application.EnableEvents = False
‘’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’
CODE
‘’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’’
Application.Display.Alerts = True
Application.ScreenUpdating = True
Application.Calculation = xlCalculationAutomatic
Application.EnableEvents = True
```

The author suffered for a long time the consequences of not including these lines in the code. The improvement in simulation time that these lines of code provide is dramatic. Although computational time depends on the computer utilized as well as on the level of optimization of the code, a comparison of the results obtained adding or removing these lines when performing the previous simulation is shown.

	NAÏVE MC		**ANTITHETIC MC**	
N	**Without code accelerator**	**With code accelerator**	**Without code accelerator**	**With code accelerator**
100	143.45	1.70	40.54	1.5
1000	1514	13.89	423.20	9.96
10000	15984	598	4560	100.62

Table 5: Comparison in Computational time (seconds) in Naïve and Antithetic versions

5.2 Suitability of spot price generators

The objective of this section is to assess if spot price generators developed (GBM, Vasicek and ARIMA) are suitable to simulate spot power and gas prices in a market as well as power and gas spreads between two countries.

5.2.1 Power spot

The power spot prices in Spain, France and Germany during January 1, 2011 and March 31, 2017 are shown below:

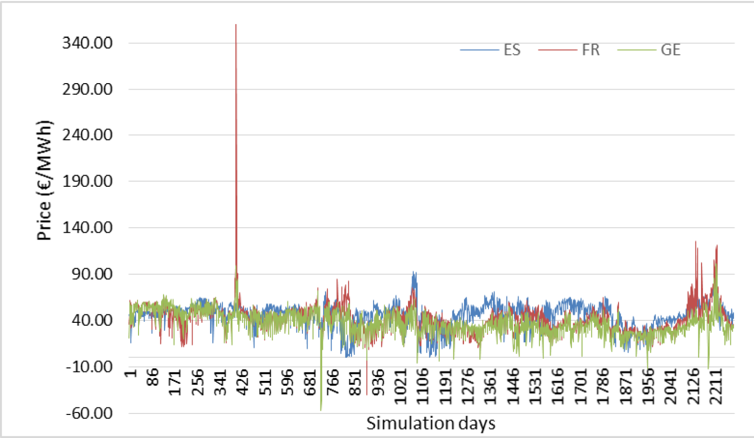


Figure 23: Average daily power spot prices in Spain, Germany and France

As outliers causes the charts to be tiny, in the figure below only values between -20€/MWh and 100 €/MWh are shown:

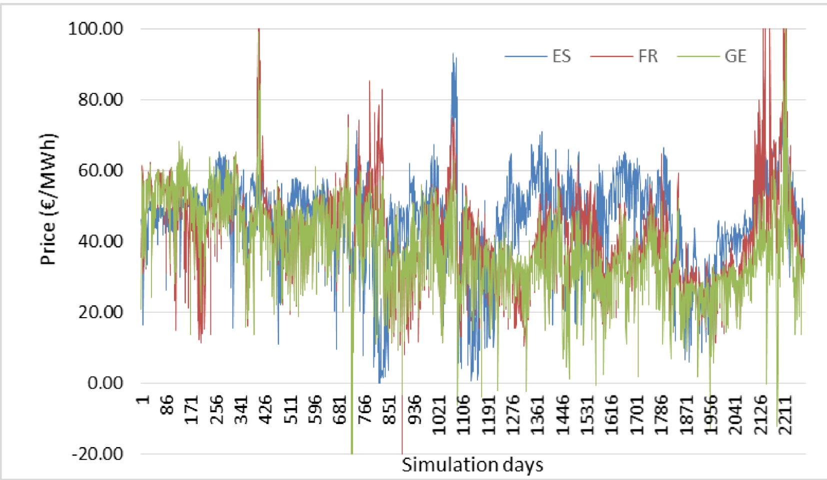


Figure 24: Average daily power spot prices in Spain, Germany and France (without outliers)

The 20-day moving average volatilities of power spot prices in Spain, France and Germany are analyzed, from the start of 2011 to the end of 2017Q1.

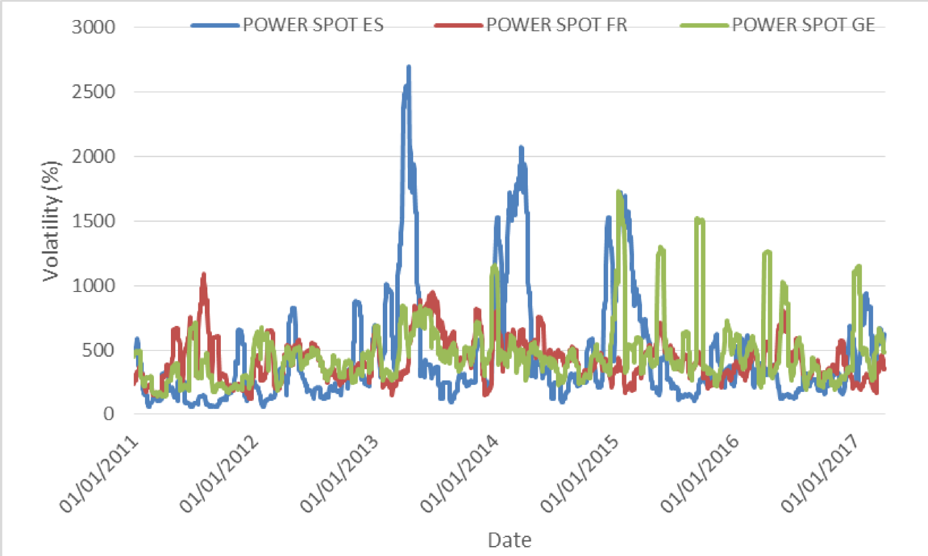


Figure 25: 20-day moving-average volatility in power spot prices in Spain, Germany and France

It is observed that there are very high volatilities in the spot price in all three countries, with averages over 350% and peaks over 1000% in all cases.

Parameter		ES	FR	GE
20-day annual (daily) volatilities (%)	**Min**	57% (3.59%)	121.4% (7.65%)	137.1% (8.6%)
	Max	2693% (169.6%)	1091.7% (68.7%)	1726.2% (108.7%)
	Average	381% (24%)	422.6% (26.6%)	457.5% (28.8%)
Average price (€/MWh)		45.97	42.01	37.63

Table 6: 20-day moving average volatilities and average prices and in Spain, France and German power spot markets

5.2.1.1 GBM

A Geometric Brownian Motion is calibrated with the parameters of the Spanish spot market. As prices are seen to reverse to the mean, drift is assumed to be zero ($\mu = 0$), start price $S_0 = 45.97\ €/MWh$ and annual volatility $\sigma = 381\%$, the average volatility of the Spanish power spot market between 01/01/2011 and 1/9/2017.

5 paths of a 5-year GBM simulation with these parameters is made:

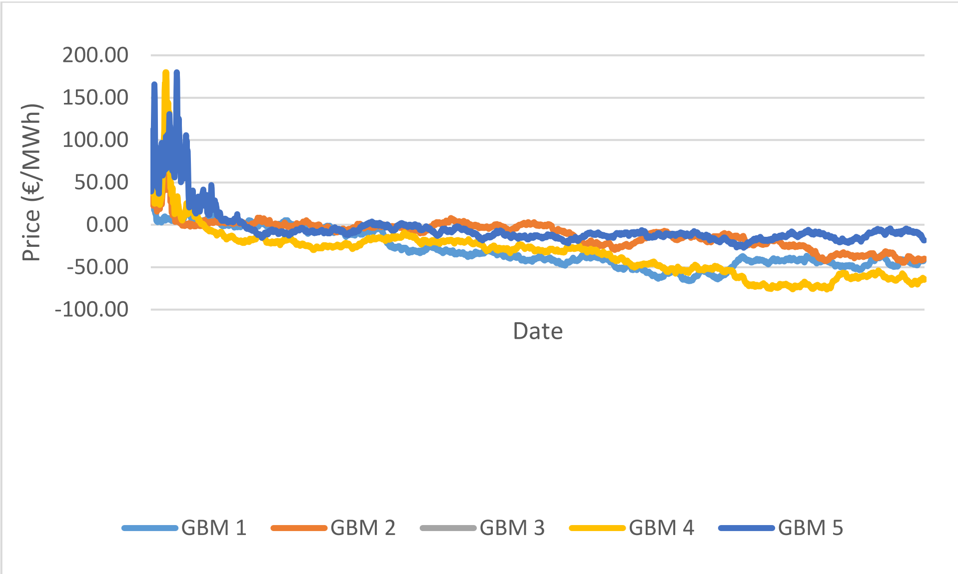


Figure 26: Simulation of 5 paths of a GBM calibrated with the parameters of the Spanish spot market

It is observed that the results obtained are disastrous, even if there is a cap of 180€/MWh. There can be seen severe price spikes and finally, all the simulations converge to zero. The reason is that in the GBM equation, in the increment between prices, the stochastic term is dependent on the previous price.

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

For high prices, volatility is extremely high, therefore at each iteration, the price can either contrinue growing towards very high values or drop dramatically. On the other hand, for low prices, volatility is very small, and at the following iteration the spot price will be very similar to the previous one, getting trapped at low prices. To avoid this, a floor on the prices has been set at 4€/MWh. 5 simulation paths are shown:

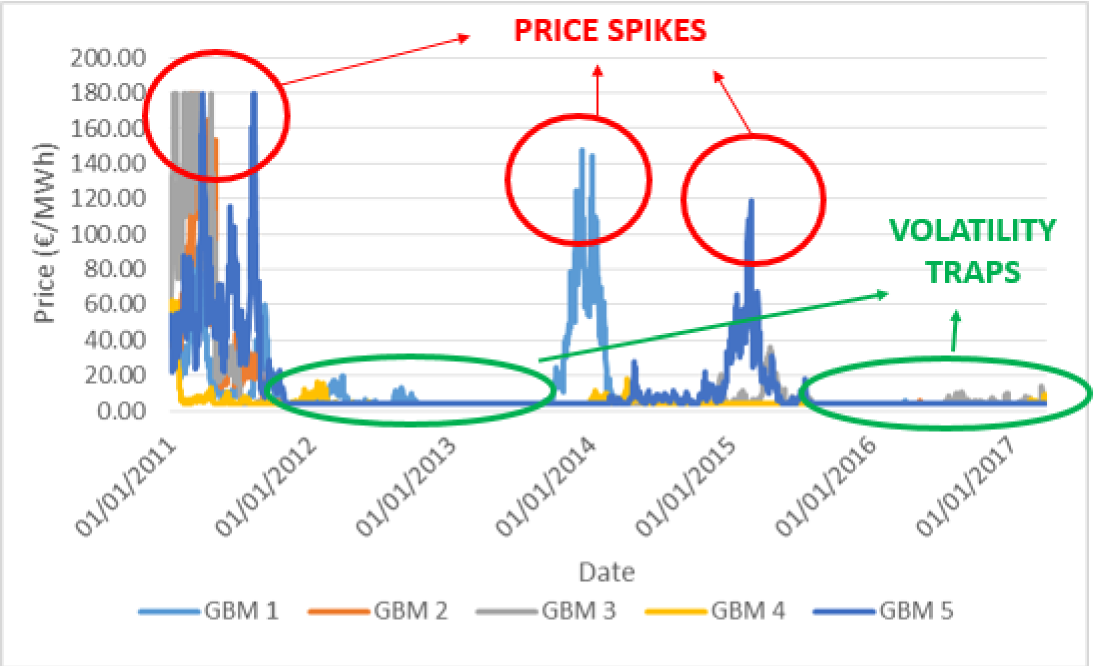


Figure 27: Simulation of 5 paths of a GBM calibrated with the parameters of the Spanish spot market after imposing a floor of 4€/MWh

After the corrections made, the model continues having serious flaws. There are severe price spikes (three series over 200 €/MWh at some point), as well as volatility traps in low prices. The model is not representative of the Spanish power spot market. For France and Germany power spot prices, the model should behave even worse, as their average price volatility is even higher than for the Spanish case.

5.2.1.2 Vasicek

The Vasicek model is trained with the average daily Spanish prices from the start of 2011 until the end of 2017Q1. The maximum likelihood parameters are:

- Long term mean $\hat{\mu} = 45.88$
- Vasicek Volatility $\hat{\sigma} = 166.05$
- Mean reversion tem $\hat{\lambda} = 76.43$

The simulation is programmed to start from $S_0 = \mu$.

The results of one simulation path of the Vasicek model are confronted with the actual prices during the 5 years and a quarter period.

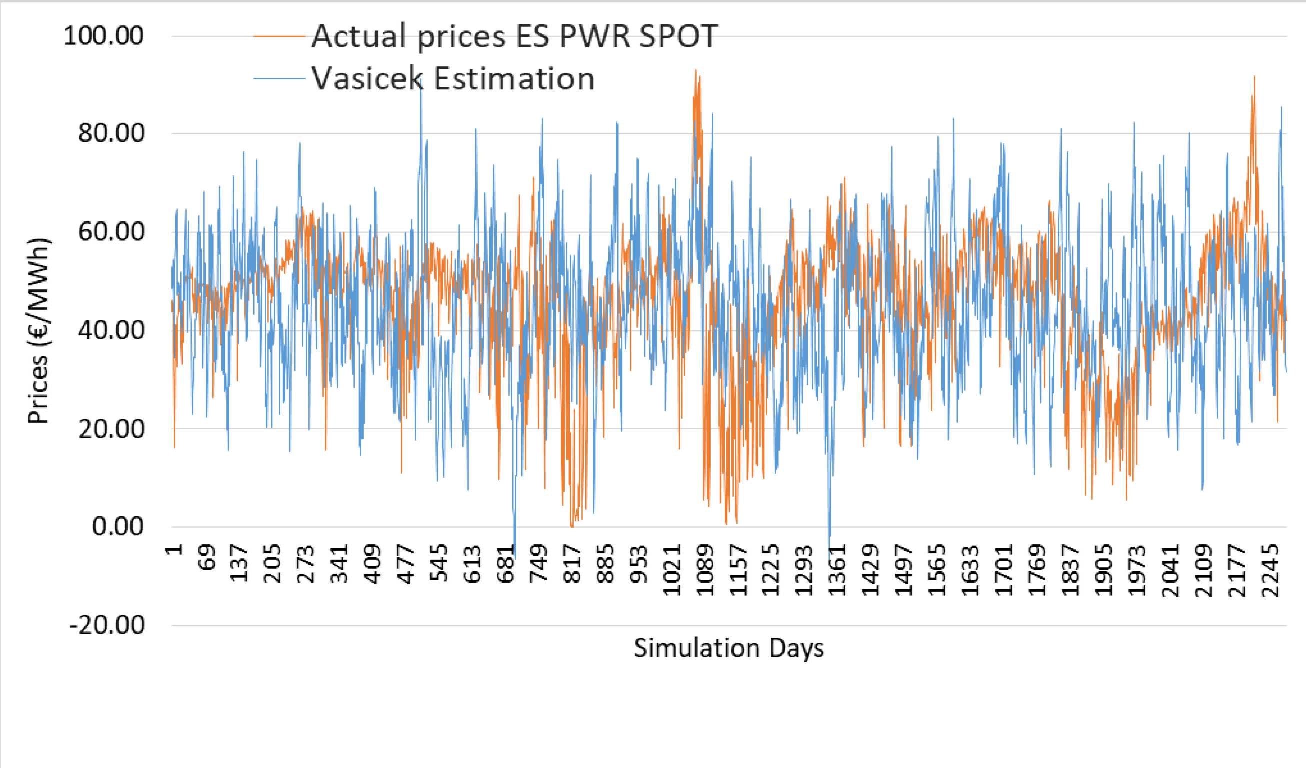


Figure 28: Comparison of 5 paths of Spain daily power spot prices to Vasicek estimation

The Vasicek model is observed to deliver a much better representation of the spot daily average prices behavior. Its aim is not to predict prices every day, but to offer an acceptable approximation to value derivatives during a period. It is seen that, the volatility of the curve is very similar to the real spot prices, thus it can be a good value to assess the temporary value of Asian options.

5.2.2 Gas spot

The gas spot daily prices in MIBGAS, TRS and TTF between the beginning of 2014 until the end of March 2017 are shown below. The MIBGAS started operations at the end of 2015, however it was not until March 14 when there was a Day Ahead clearing was performed daily.

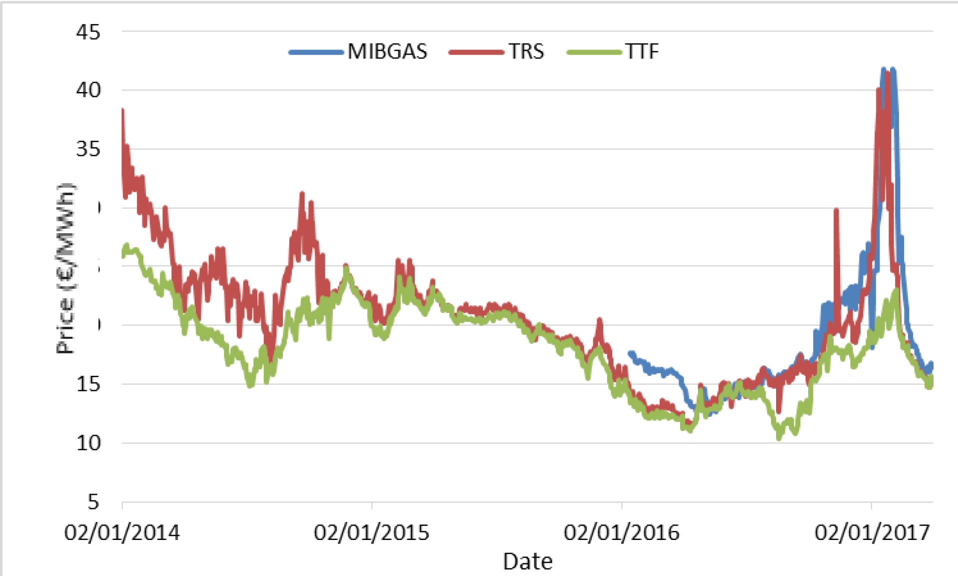


Figure 29: Comparison of 5 paths of Spain power spot prices to Vasicek estimation

Parameter		**MIBGAS (Jan 16-Mar17)**	**TRS (Jan 14–Mar 17)**	**TTF (Jan 14–Mar 17)**
20-day anual (daily) volatilities (%)	**Min**	14.3% (0.9%)	16.8% (1.1%)	10.3% (0.65%)
	Max	179.9% (11.3%)	215.2% (13.6%)	87.5% (5.6%)
	Average	53.9% (3.4%)	64.4% (4.1%)	41.2% (2.6%)
Average price (€/MWh)		45.97	20.47	37.63

Table 7: Volatilities and average prices and in Spain (MIBGAS), South France (TRS) and Netherlands (TTF) gas spot markets

5.2.2.1. GBM

A Geometric Brownian Motion model is calibrated with the TTF spot market prices. In this case, as it appears to be a downward tendency in TTF prices, but there is a rebound at the end. The estimators for volatility σ and drift μ from the data in the whole period are:

- Volatility $\hat{\sigma} = 44.16\%$
- Drift $\hat{\mu} = \frac{\bar{r}}{t_i - t_{i-1}} + \frac{\hat{\sigma}^2}{2} = -0.139$

The start price S_0 is taken at the start price the first day of the period: $S_0 = 26.30\ €/MWh$

Five simulation paths are shown in the image, along with actual TTF prices (in green):

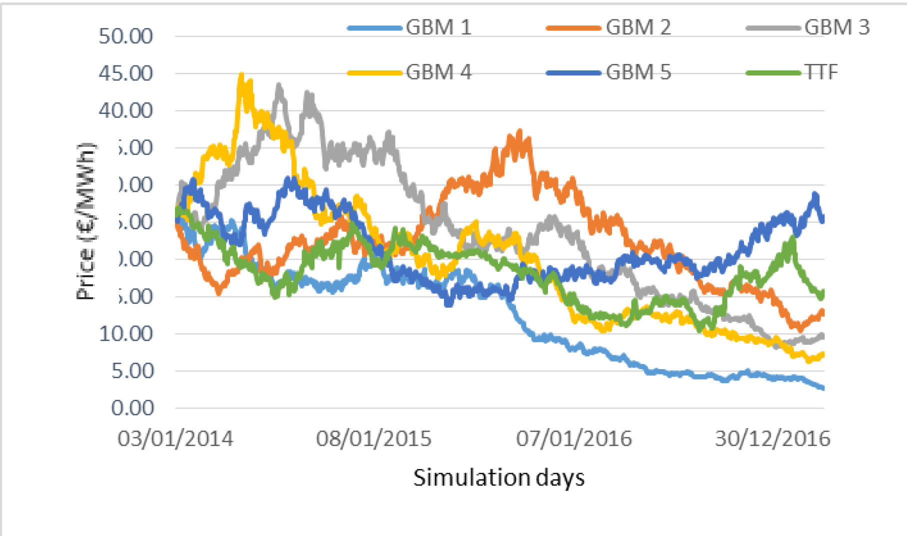


Figure 30: Simulation of GBM (5 paths) calibrated with the TTF prices

It is observed that, while for power spot prices the GBM had much faults, for gas spot prices it is a much more precise model. This is due to the fact that volatilities in daily spot gas prices are much lower than for average daily power prices. Therefore in the GBM model there are not big increments between one price and the following, and this allows to avoid both spikes and lower prices where the GBM model gets trapped.

5.2.2.2. Vasicek

The calibration of the Vasicek model with the TTF prices for the same period yields:

- Long-term mean $\hat{\mu} = 13.69$
- Vasicek volatility $\hat{\sigma} = 13.91$
- Mean reversion parameter $\hat{\lambda} = 2.61$

In this case the start price $S_0 = 26.30\ €/MMBtu$ on January 2, 2014 is far from the long-term mean, so it does not make sense to start from $\hat{\mu}$.

5 paths of Vasicek simulations are compared with the TTF reference:

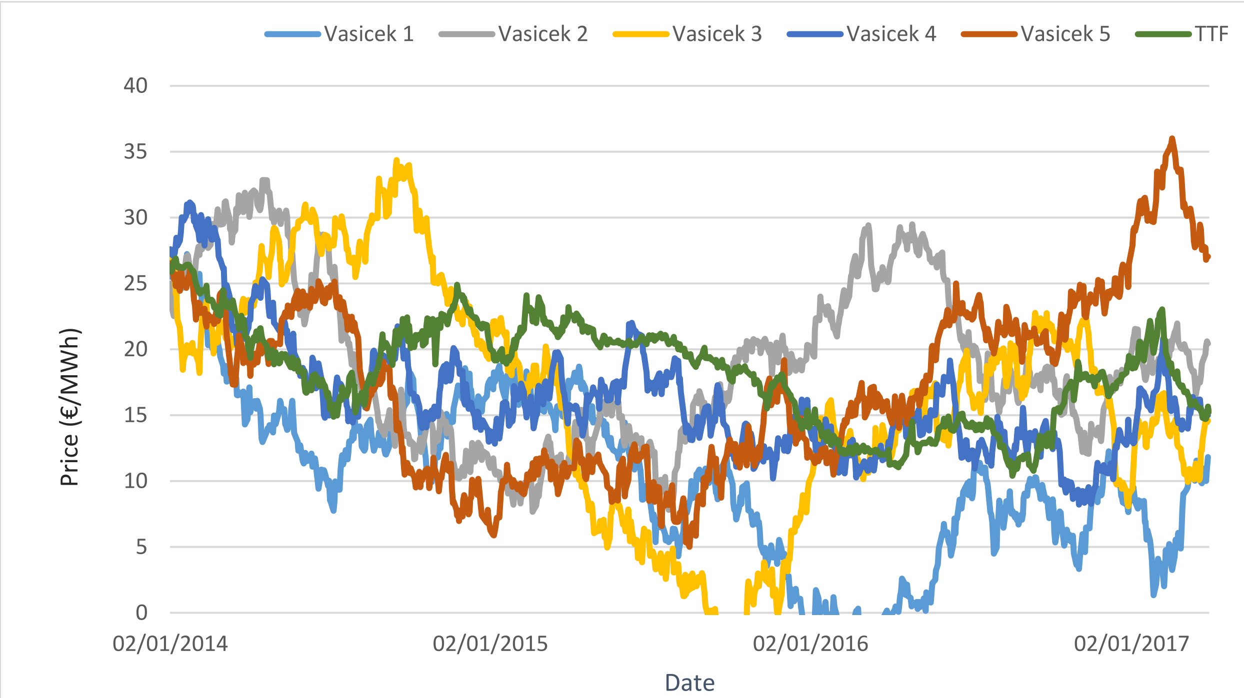


Figure 31: Simulation of Vasicek model (5 paths) calibrated with the TTF prices

The Vasicek simulation does not take into account the drift or tendency. It is seen that the spread between the simulations and the TTF price is much bigger than the spread between the GBM and the TTF. In the Vasicek model, the Vasicek "volatility" σ is typically larger than its counterpart in the GBM, but the reversion term allows to adjust to the mean values. However, in this example λ is very small, causing high differences between the Vasicek simulation and the TTF prices.

5.3 Valuation of options in Spanish electricity market

The aim is to make valuation of Asian options for year 2015 as in real conditions. An in-sample test will be made for 2014Q4. Although prices in 2014 were lower than in 2015 due to high renewable production, during the last quarter prices showed very similar behavior to 2015 prices, therefore it is used as the training set to valuate options in the beginning of 2015.

5.3.1 In-Sample Testing

In-sample testing for Q4 2014

Monthly caps are valuated for the last quarter of 2014, with the valuation models (GBM MC, Vasicek MC) trained within that period. The only exception is the ARIMA model, for which the training period has been set to include three months prior, which has been observed to be a more robust choice.

The models are fitted and their payoffs of the monthly caps during 2014Q4 are computed. In the case of the Monte Carlo models (GBM and Vasicek spot price generators), the payoff is calculated as the average of 7,500x2 antithetic paths. For the ARIMA model, the payoff is computed for the fitted model. The monthly payoffs are averaged for different strike prices (*K*) and compared to the actual results of real monthly caps during that period.

The value of the option is the expectation of profits, discounted three months at an interest rate r. The risk-free interest rate *r* will be assumed to be zero, therefore the value of the option will be the same as the expectation of profits.

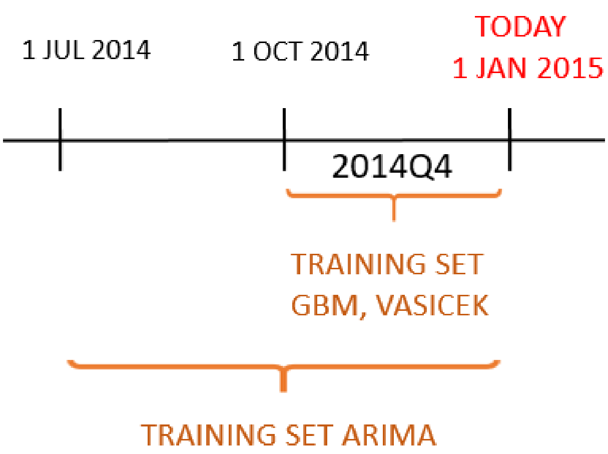


Figure 32: Representation of the training sets of the GBM, the Vasicek and the ARIMA price generators

The average price of the period was 49.82 €/MWh, while volatility was 379% for 2014Q4. Therefore, in the GBM the parameters are set to:

- The assumption drift $\mu = 0$ is made,
- $S_0 = 49.82\, €/MWh$
- $\sigma = 379\%$

The parameters for the Vasicek model are estimated during the same period (2014Q4), obtaining:

- $\hat{\sigma} = 238$
- $\hat{\lambda} = 205.94$
- Long-term mean $\mu = 49.82\, €/MWh$

The ARIMA model is trained for the last previous six months to the date of valuation (2014Q3 & Q4). The best fit was achieved for ARIMA(1,0,1)(1,0,0)[7]. It is observed that the coefficients are significant, and the ACF and PCF plots do not suggest significant correlation.

```
             Estimate Std. Error z value  Pr(>|z|)
ar1          0.448481   0.092733  4.8363 1.323e-06 ***
ma1          0.330681   0.095167  3.4747 0.0005114 ***
sar1         0.446894   0.066422  6.7281 1.719e-11 ***
intercept  931.671731  60.596270 15.3751 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

Figure 33: Significance of coefficients in the ARIMA model

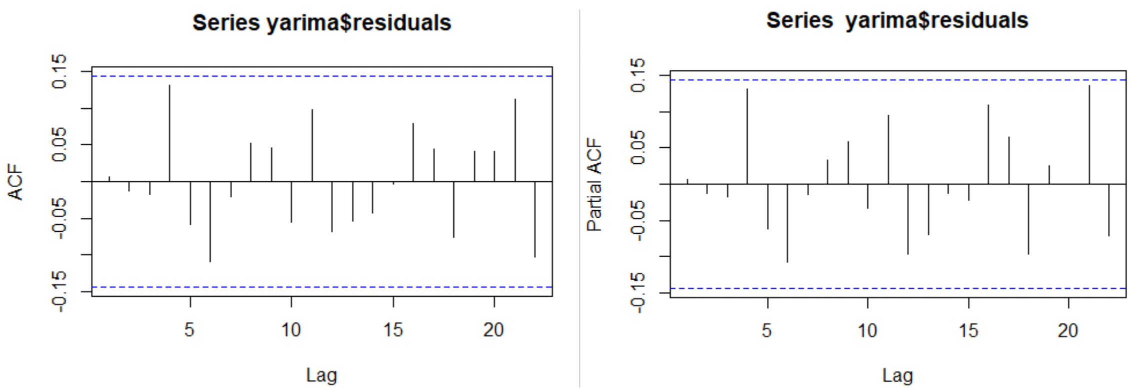
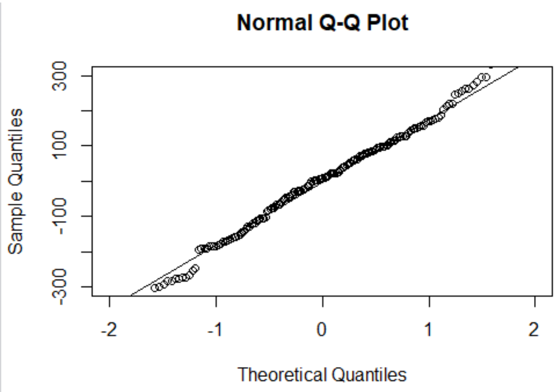


Figure 34: ACF and PCF of the residuals of the ARIMA model

The QQ plot and the Ljung-Box test indicate normality and independence between the residuals, respectively:




```
        Box-Ljung test

data:  yarima$residuals
X-squared = 11.715, df = 14, p-value = 0.6291
```

Figure 35: QQplot of the residuals and Ljung-Box test of the ARIMA model

The actual average monthly prices for the 2014Q4 are shown, together with the average prices and payoffs per month obtained by each method.

Month	Parameter	Actual	GBM-MC	Vasicek-MC	ARIMA
OCT	**Average Price**	55.10	49.51	49.79	54.68
	Payoff (K=48)	7.10	13.37	2.59	6.68
NOV	**Average Price**	46.80	49.70	49.84	48.87
	Payoff (K=48)	0	23.73	2.71	0.87
DEC	**Average Price**	47.47	49.51	49.78	48.86
	Average Payoff (K=48)	0	31.08	2.60	0.86
AVERAGE OCT-DEC	**Average Monthly Price**	49.92	49.57	49.80	50.81
	Average Payoff (K=48)	0	31.08	2.60	2.80

Table 8: In-sample computation of payoffs of monthly caps during 2014Q4

It is noted that in the Monte Carlo based models, the payoff is not the difference between the average monthly price and the strike price (*K*). This would only be the case if all the price paths had average price over *K*. However, as there are paths with average monthly price below *K*, the averaging of the path payoffs does not reflect any more the average price of the month.

The real payoff of the Asian option at different strikes is compared to the average monthly payoffs estimated with the three different methods. Nevertheless, in the GBM only the results of the first month are considered, as the model diverges quickly and the estimations of the payoffs in the latter months are totally unrealistic in this case. The results are shown below:

K	**44**	**46**	**48**	**49**	**50**	**52**	**54**	**56**
OCT	11.10	9.10	7.10	6.10	5.10	3.10	1.10	0
NOV	2.80	0.80	0	0	0	0	0	0
DEC	3.47	1.47	0	0	0	0	0	0
AVG (OCT-NOV)	5.79	3.79	2.37	2.03	1.70	1.03	0.37	0
ANTITHETIC – VASICEK (15000 sim)	5.93	4.15	2.63	2.01	1.47	0.71	0.28	0.09
ANTITHETIC – GBM (15000 sim)	14.95	14.14	13.37	13.00	12.65	11.97	11.34	10.75
ARIMA	6.80	4.80	2.80	1.71	1.89	0.89	0.23	0

Table 9: Average payoffs during the training period of the 3 developed models, compared with the actual payoffs

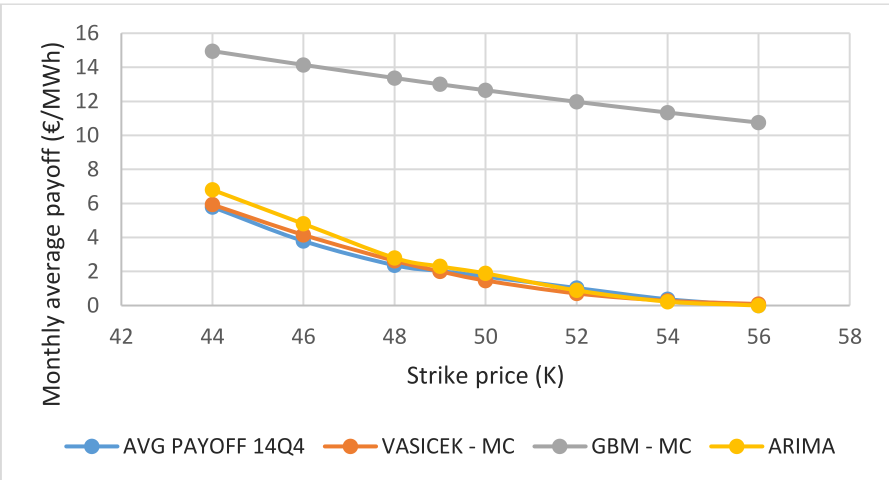


Figure 36: Average payoffs during the training period of the 3 developed models, compared with the actual payoffs

In this example, it is observed that both the valuation with both Monte Carlo Vasicek and the ARIMA fitted values shows very similar payoffs to the actual payoffs in the period.

5.3.2 Out-of-Sample Testing

In this section, monthly caps and floors will be evaluated for all the different quarters in 2015. The valuation will be done the day before starting the quarter. The models are trained with the prices of the quarter before, in the case of the Monte Carlo GBM and Vasicek models, and with the data of the last six months, in the case of the Monte Carlo ARIMA model. For 2015Q1, the training sets for each model are shown below:

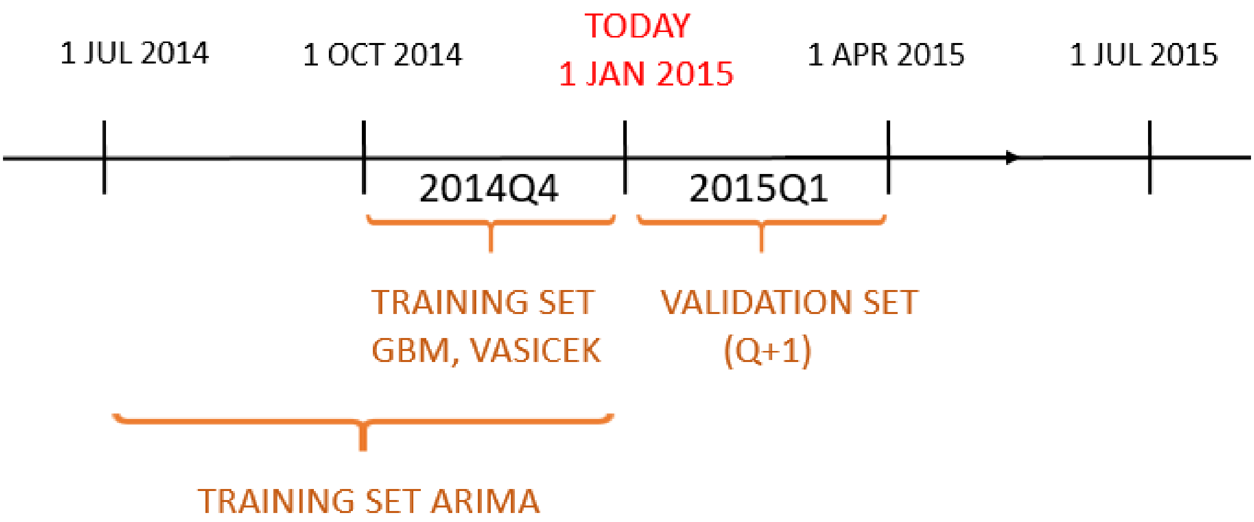


Figure 37: Representation of the training and validation sets of the GBM, the Vasicek and the ARIMA price generators

In order to make a valuation of the options, it is important not only to look at the historical data, but to correct the values according to future expectations. For this reason, an adjustment has been made in the models, comparing the average prices during the training period the Q+1 future the day before the valuation quarter will be considered as the price reference.

For the GBM and Vasicek spot price models 7,500x2 antithetic Monte Carlo paths are simulated in the valuation period, and the average payoff of the Asian option is computed. For the GBM and Vasicek, the Q+1 future the day before the start of the valuation period is set as the start price. For the Vasicek model it is also fixed equal to the long-term mean μ..

For the ARIMA model, the parameter estimation is performed in the six months before the start of the valuation period. 1,000 random Monte Carlo simulations are performed for the valuation period, and the average payoff of the Asian option is computed.

A corrected ARIMA valuation is performed. This approach takes the random ARIMA forecasts, and adds to each path the difference between the Q+1 price in the following period and the average price of the training period. This approach is performed to eliminate the gap between historical behavior and future expectations, which are key when assessing option prices.

The results will be compared with Vorst estimation and with the actual payoffs. The risk-free interest rate r is set to 0.0001, to avoid problems with the Vorst estimation, where r cannot be set to zero

Estimation of the price of a monthly cap for 2015Q1

The three models have been trained in Q4 2014, therefore its parameters, except the average are estimated in this period, and have been obtained in the previous section. S_0 and μ are fixed to 44.13 $€/MWh$, the price of Q+1 on December 31, 2014.

The ARIMA model employed is the (1,0,1)(1,0,0)[7] discussed in the previous section. The normality and independence of its residuals is analyzed in Annex I. Its simulation forecasts (percentiles 60 and 95%) are compared to the actual spot prices (in red) and can be seen below:

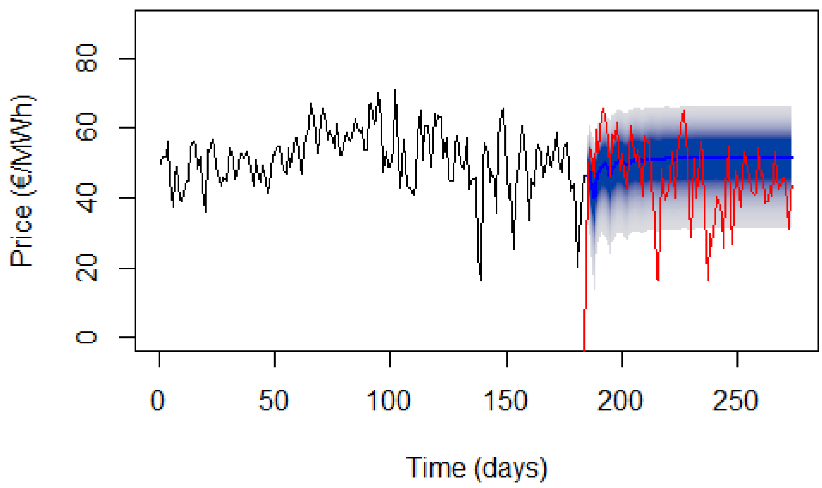


Figure 38: Representation of the forecasts of daily prices in date 12/31/2014 (in blue, 60% and 95% percentiles shown) of the ARIMA model, and the actual average daily prices in the following quarter (red)

Five random ARIMA paths are shown below for the forecasted period:

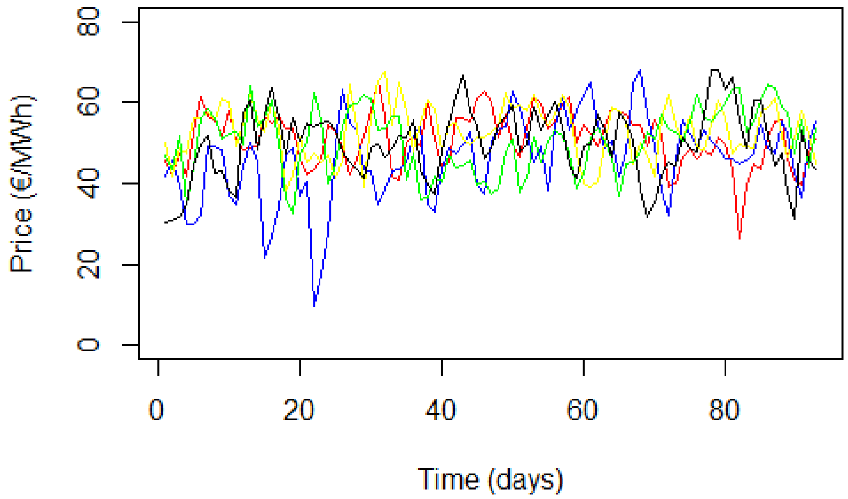


Figure 39: Representation of 5 random paths of the ARIMA forecast, during 2015Q1

A comparison is made between the real actual payoffs of the Asian period and the fair price of the Asian options according to the Monte Carlo estimations, after averaging the payoffs of the three months:

K	38	40	42	43	44	45	46	48
JAN	13.60	11.60	9.60	8.60	7.60	6.60	5.60	3.60
FEB	4.57	2.57	0.57	0	0	0	0	0
MAR	5.11	3.11	1.11	0.11	0	0	0	0
AVG (JAN-MAR)	7.76	5.76	3.76	2.90	2.53	2.19	1.87	1.20
ANTITHETIC – VASICEK (15000 sim)	6.22	3.97	2.82	2.17	1.61	1.14	0.81	0.34
VORST	12.72	11.81	10.97	10.57	10.18	9.82	9.47	8.80
ANTITHETIC – GBM (15000 sim)	14.04	12.87	12.20	11.97	11.68	11.15	10.54	9.92
ARIMA	9.66	7.77	5.98	5.13	4.34	3.62	2.95	1.84
ARIMA CORRECTED	6.35	4.67	3.22	2.59	2.05	1.58	1.18	0.61

Table 10: Out-of-sample estimation of the average payoffs of monthly caps during 2015Q1, during the training period, and comparison with the actual payoffs

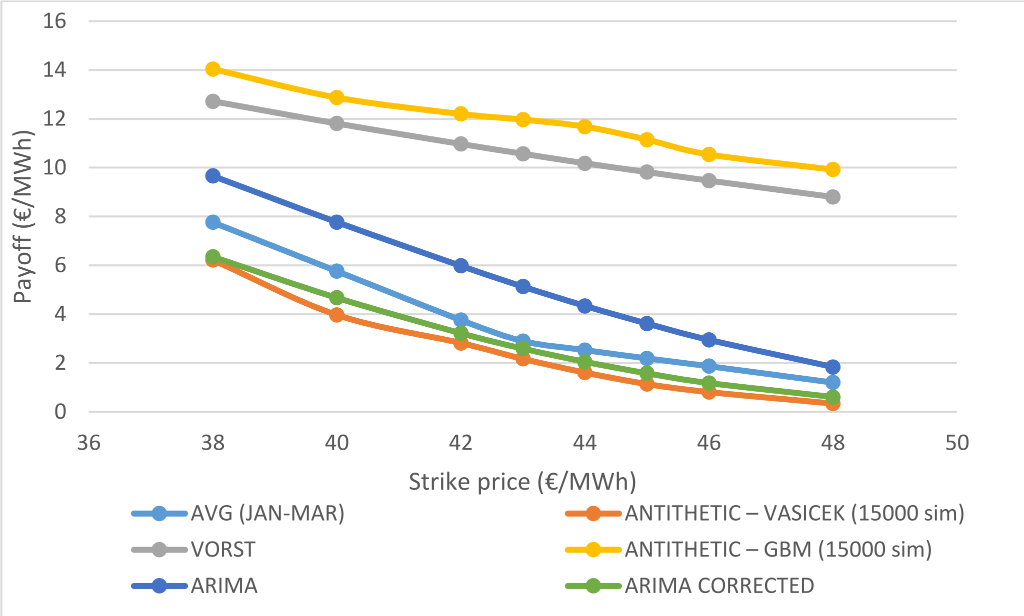


Figure 40: Comparison of the forecasted payoffs for monthly caps in 2015Q1 for different methods, and the actual payoffs which would have delivered these caps during 2015Q1.

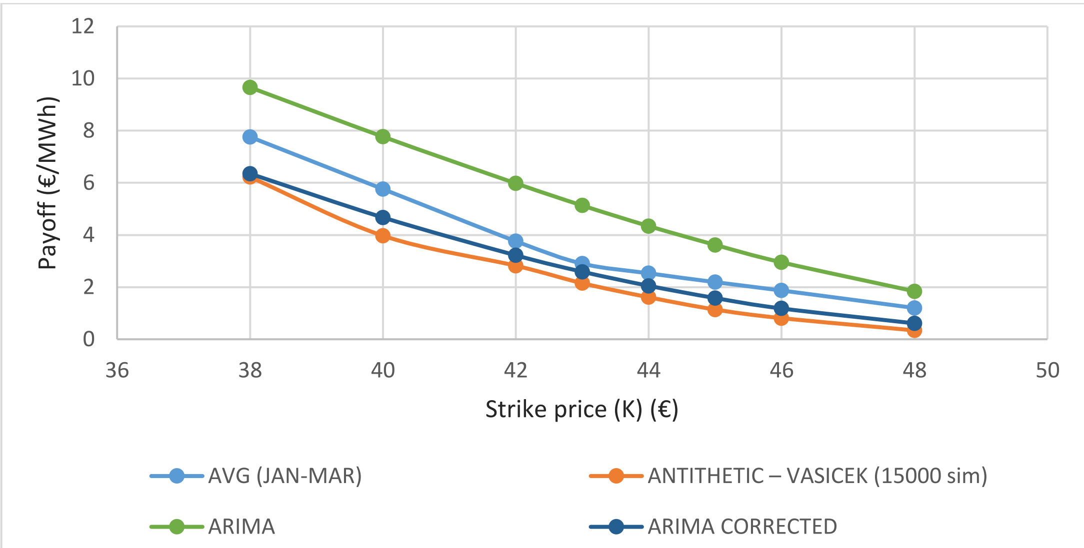


Figure 41: Comparison of the forecasted payoffs for monthly caps for 2015Q1 for the most reliable methods and the actual payoffs which would have delivered these caps during 2015Q1.

In this case, the corrected MC-ARIMA method is the one which would have predicted prices better with the MC-Vasicek being the next one. It is reminded that the appearance of errors in an estimation does not imply that the estimation is poor, as prices can change. As a matter of fact, the average price in the Vasicek model was 44.14 €/MWh, almost similar to its long-term mean, while finally the actual average price in the period was 45.76 €/MWh meaning that there was an increase in the prices with respect to the expectations. It is seen that Vorst estimation and the MC-GBM do not provide a nice approximation in this context of high volatilities. This is coherent with the (Nielsen, 2011) finding that the Vorst model fails in context of high volatilities, as it is the case.

The utility may face risk high market prices in case it decides not to hedge them average monthly price in 95% percentile is 50.14 €/MWh (a 13.6% increase in market prices).

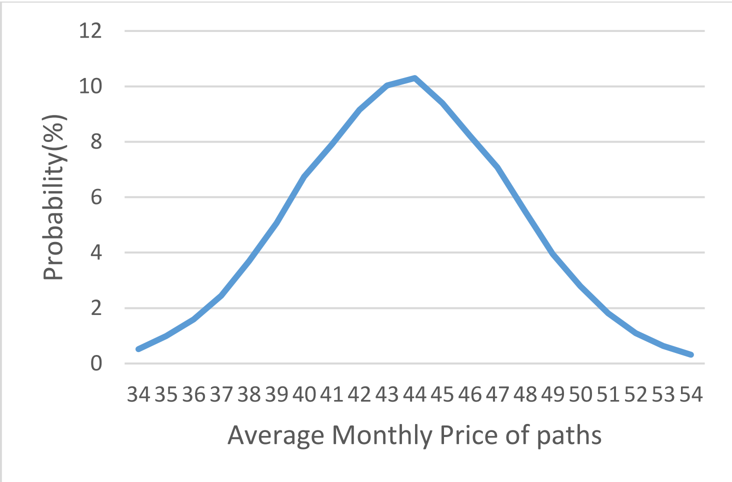


Figure 42: Percentage of paths with a certain average monthly price (rounded to the closest unit)

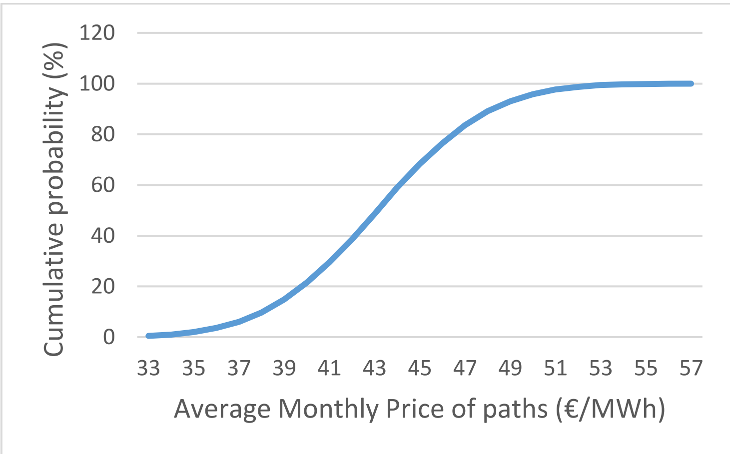


Figure 43: Cumulative percentage of paths with a certain average monthly price (rounded to the closest unit)

An approximation delta of a monthly option of strike $K = 43\ €/MWh$ is computed for the MC-Vasicek method at the current reference price $S = 44.13\ €/MWh$.

$$\Delta \approx \frac{V[S+\Delta S] - V[S-\Delta S]}{2\ \Delta S} = \frac{V[44.13+0.5] - V[44.13-0.5]}{2\ \ 0.5} = 2.53 - 1.91 = 0.62$$

Estimation of the price of a monthly cap for 2015Q2

The Q+1 on March 31, 2015 was 44.29 €/MWh. The estimation of the parameters during 2015Q1 yields $\hat{\sigma} = 4.19$ (annually) for GBM and $\hat{\sigma} = 219.68$ (annually), $\hat{\lambda} = 189.02$ for Vasicek.

The ARIMA model trained during periods 2014Q4 and 2015Q1 is again (1,0,1)(1,0,0)[7]. It is shown to reflect insignificance of coefficients, independence and normality of residuals (Annex I).

A forecast of 2015Q2 delivered by ARIMA model is shown (percentiles 60% and 95%) compared to the actual spot prices which were obtained for the period:

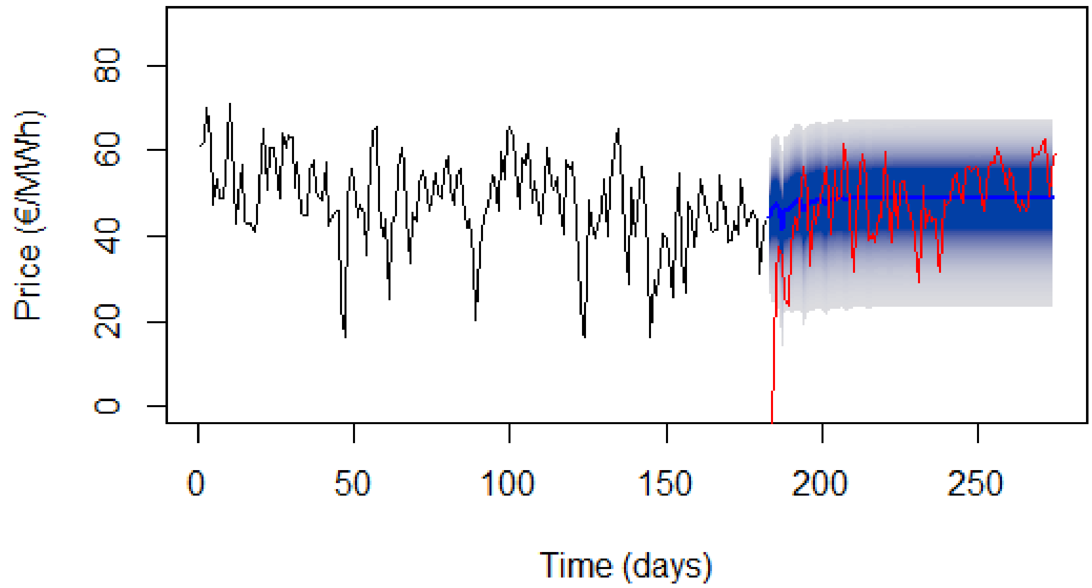


Figure 44: Representation of the forecasts of daily prices in date 3/31/2014 (in blue, 60% and 95% percentiles shown) of the ARIMA model, and the actual average daily prices in the following quarter (red)

K	38	40	42	43	44	45	46	48
APR	7.34	5.34	3.34	2.34	1.34	0.34	0	0
MAY	7.12	5.12	3.12	2.12	1.12	0.12	0	0
JUN	16.73	14.73	12.73	11.73	10.73	9.73	8.73	6.73
AVG (APR-JUNE)	10.40	8.40	6.40	5.40	4.40	3.40	2.91	2.24
ANTITHETIC – VASICEK (15000 sim)	6.38	4.57	2.97	2.30	1.71	1.26	0.85	0.35
VORST	13.51	12.63	11.82	11.44	11.07	10.71	10.37	9.71
ANTITHETIC – GBM (15000 sim)	15.04	14.21	13.46	13.09	12.70	12.37	12.03	11.46
ARIMA	9.66	7.78	5.98	5.14	4.35	3.62	2.95	1.84
ARIMA CORRECTED	6.41	4.73	3.28	2.67	2.12	1.65	1.26	0.68

Table 11: Out-of-sample estimation of the average payoffs of monthly caps during 2015Q2, during the training period, and comparison with the actual payoffs

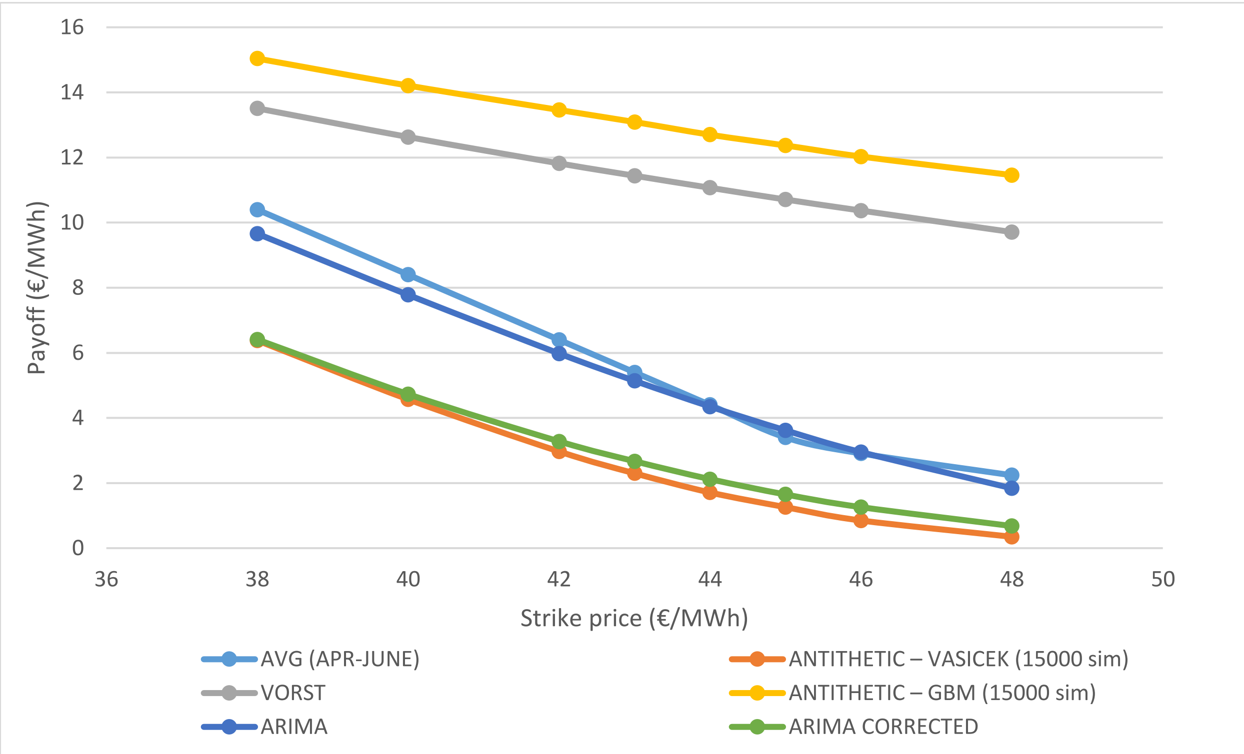


Figure 45: Comparison of the forecasted payoffs for monthly caps for 2015Q2 for different methods, and the actual payoffs which would have delivered these caps during 2015Q2.

In this case, an increase in the prices in June (54.73€/MWh) during 2015Q2 with respect to the expectations (44.29 €/MWh) implies that all the valuation models' payoff expectation were short. The corrected ARIMA and the MC-Vasicek estimate nearly the same payoffs. Vorst estimation and MC-GBM do not provide accurate estimations, due to the high volatilities of power spot markets.

Estimation of the price of a monthly floor for Q3 2015

The Q+1 on June 30, 2015 was 53.52 €/MWh. The estimation of the parameters during 2015Q1 yields $\hat{\sigma} = 2.87$ (annually) for GBM and $\hat{\sigma} = 229.29$ (annually), $\hat{\lambda} = 247.57$ for Vasicek.

The ARIMA model trained during periods 2015Q1 and 2015Q2 is the model ARIMA (1,0,2)(0,1,1)[7].

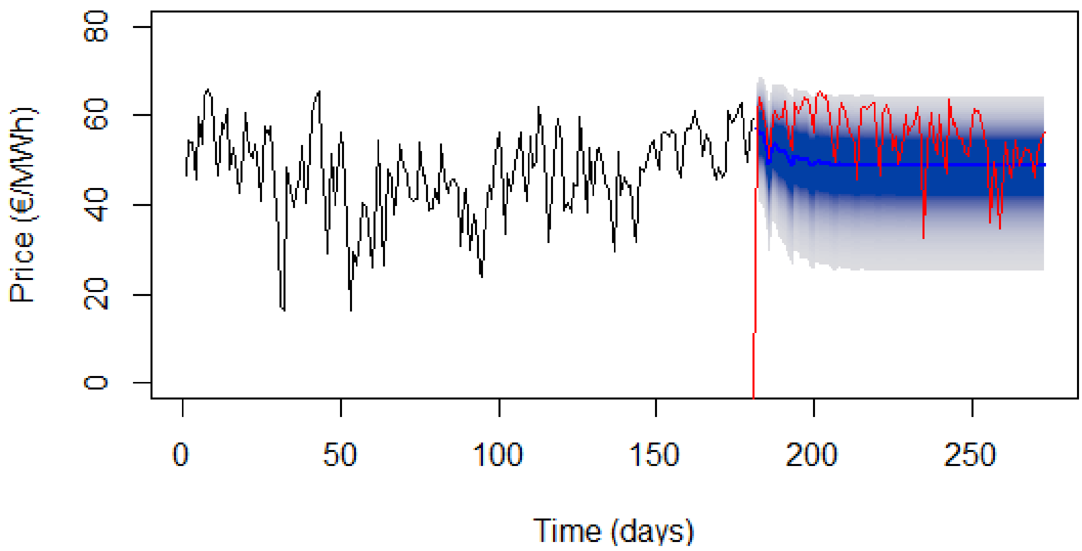


Figure 46: Representation of the forecasts of daily prices in date 6/30/2014 (in blue, 60% and 95% percentiles shown) of the ARIMA model, and the actual average daily prices in the following quarter (red)

K	48	50	52	53	54	55	56	58
AVG (JUL-SEP)	0	0	0.041	0.374	0.707	1.041	1.512	2.845
ANTITHETIC – VASICEK (15000 sim)	0.06	0.21	0.65	1.03	1.52	2.13	2.86	4.58
VORST	6.74	7.79	8.90	9.48	10.07	10.69	11.31	12.04
ANTITHETIC – GBM (15000 sim)	7.18	8.25	9.39	9.98	10.6	11.22	11.85	12.61
ARIMA	1.86	2.96	4.39	5.22	6.1	7.02	7.98	9.95
ARIMA CORRECTED	0.31	0.56	0.98	1.3	1.69	2.15	2.71	4.08

Table 12: Out-of-sample estimation of the average payoffs of monthly floors during 2015Q3 and comparison with the actual payoffs

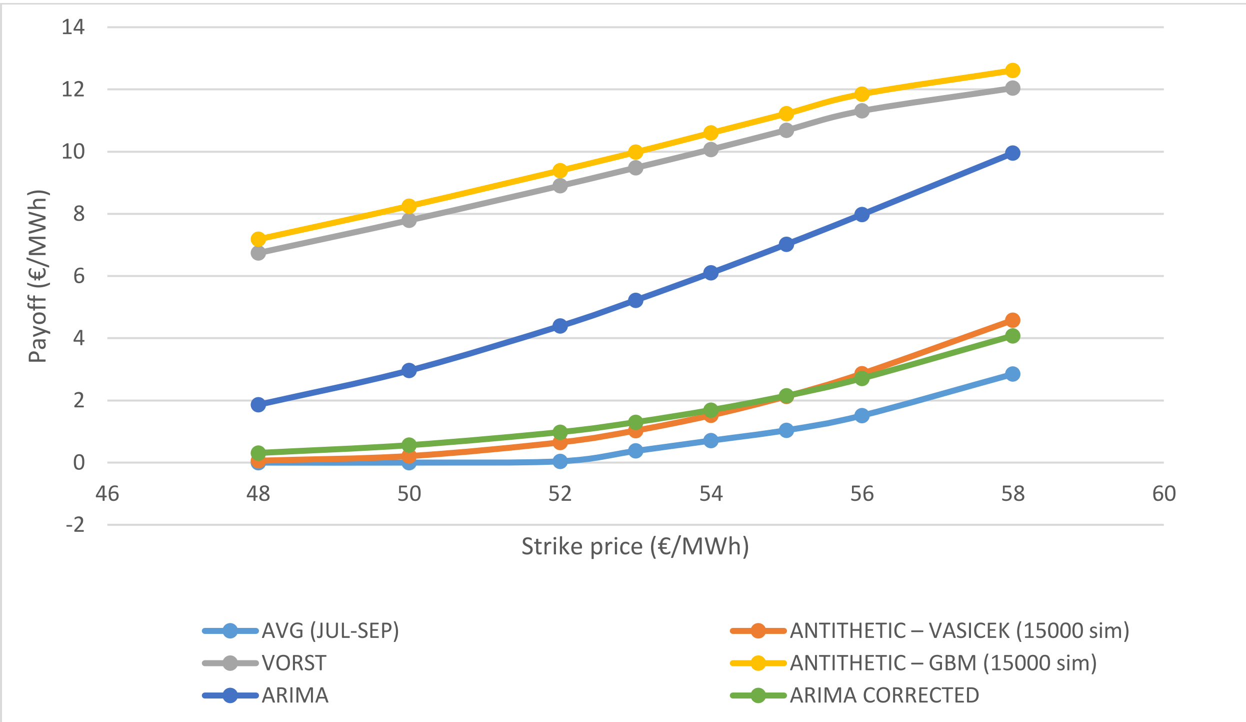


Figure 47: Comparison of the forecasted payoffs for monthly caps for 2015Q3 for different methods, and the actual payoffs which would have delivered these caps during 2015Q3.

In this case the prices drop a little, therefore being payoff estimations a bit overvalued. There are still similarities between the estimation of the corrected ARIMA and the MC-Vasicek models, providing once again to be a fairly good estimation.

Estimation of the price of a monthly floor for Q4 2015

The Q+1 on September 30, 2015 was 47.85 €/MWh. The estimation of the parameters during 2015Q13 yields $\hat{\sigma} = 2.27$ (annually) for GBM and $\hat{\sigma} = 219.91$ (annually), $\hat{\lambda} = 285.73$ for Vasicek.

The ARIMA model trained during periods 2015Q2 and 2015Q3 is again an ARIMA (1,0,2)(0,1,1)[7].

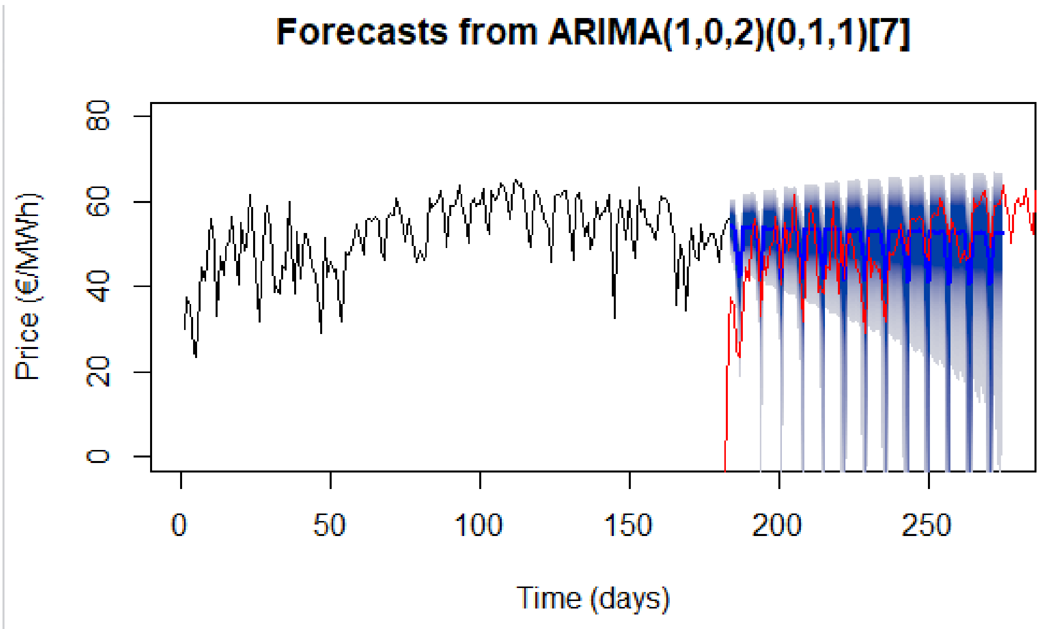


Figure 48: Representation of the forecasts of daily prices in date 9/30/2014 (in blue, 60% and 95% percentiles shown) of the ARIMA model, and the actual average daily prices in the following quarter (red)

K	42	44	46	47	48	49	51	53
AVG (OCT-DEC)	0	0	0	0	0	0	0.37	1.76
ANTITHETIC – VASICEK (15000 sim)	0.011	0.08	0.37	0.67	1.12	1.72	3.30	5.17
VORST	3.91	4.83	5.84	6.38	6.95	7.54	8.78	10.11
ANTITHETIC – GBM (15000 sim)	4.30	5.18	6.22	6.88	7.46	8.06	9.33	10.53
ARIMA	3.10	3.61	4.24	4.61	5.01	5.44	6.47	7.72
ARIMA CORRECTED	4.31	5.09	6.04	6.59	7.20	7.85	9.33	10.99

Table 13: Out-of-sample estimation of the average payoffs of monthly floors during 2015Q4 and comparison with the actual payoffs

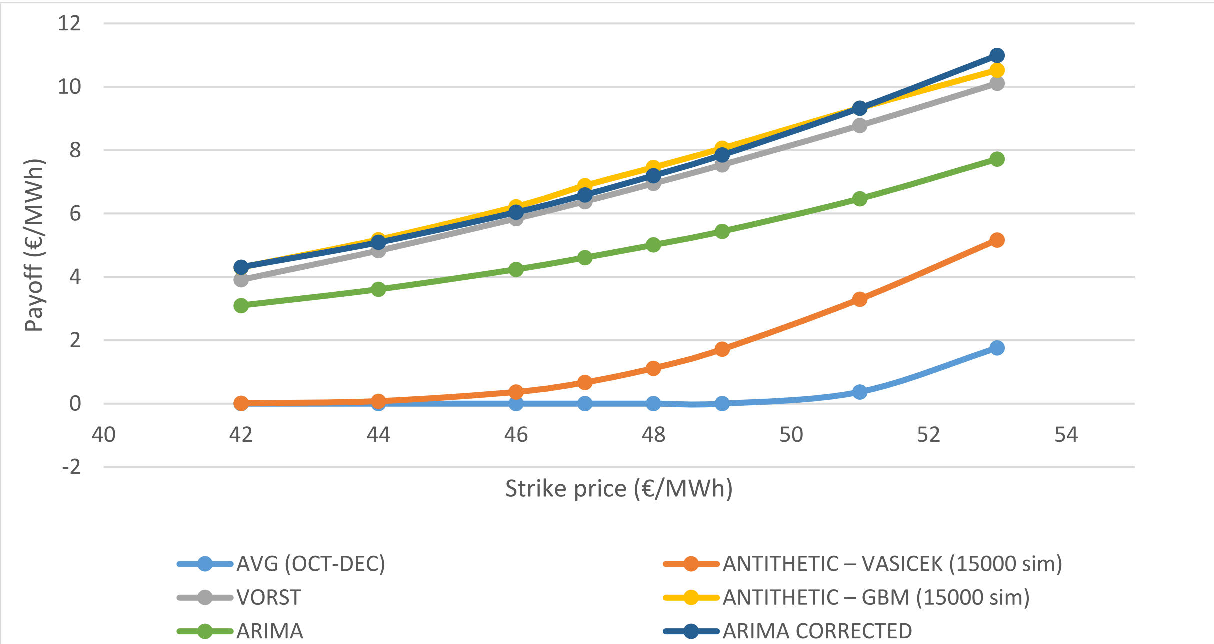


Figure 49: Comparison of the forecasted payoffs for monthly caps for 2015Q4 for different methods, and the actual payoffs which would have delivered these caps during 2015Q4.

The prices drop in the last quarter of 2015. The ARIMA model does not provide a good estimation of the results. In fact, this was predictable studying the forecasts of the model (Figure 48), which did not have enough quality. MC-Vasicek do not estimate as weel the payoffs because of the fall in the prices.

5.4 Risk assessment

In this section it is shown how to perform calculations of the risk indicators, according to historical tools and the Monte Carlo simulations.

5.4.1 Computation of MTM

On February 7, the price of the MAR17 future is 48 €/MWh. Therefore, the Mark-to-Market of the future contract is:

$$MTM = 50{,}000\ [48 - 49.20\ ] = -60{,}000\ €$$

5.4.2 Computation of VaR

VaR Case 1: Computation of VaR for the following day

Supposing 50,000 MWh of a Spanish power MAR17 future are bought on February 6, 2017, at a price of 49.20 €/MWh.

To estimate the highest loss (with a 5% probability) on February 8, three approaches can be made:

Historical simulation

The evolution of prices of the Spanish power MAR17 product is:

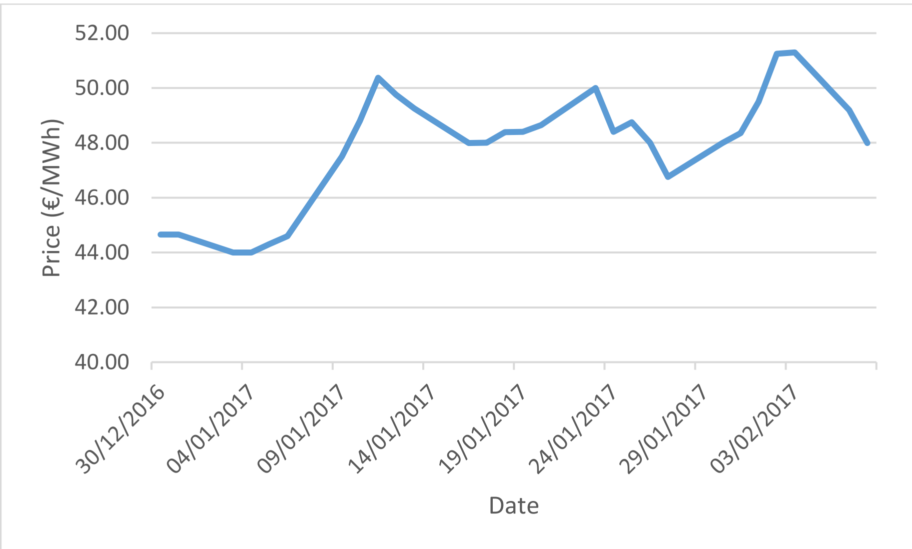


Figure 50: Representation of the evolution of Spanish power MAR17

The daily log returns of the previous 20 days are computed:

Date	MAR 18 (€/MWh)	LOG RETURN	ANNUALIZED LOG RETURN
10/01/2017	48.80		
11/01/2017	50.38	3.18%	50.42%
12/01/2017	49.75	-1.25%	-19.82%
13/01/2017	49.25	-1.01%	-16.03%
16/01/2017	48.00	-2.58%	-40.94%
17/01/2017	48.00	0.01%	0.17%
18/01/2017	48.39	0.82%	12.94%
19/01/2017	48.40	0.01%	0.08%
20/01/2017	48.65	0.52%	8.28%
23/01/2017	50.00	2.74%	43.44%
24/01/2017	48.40	-3.26%	-51.70%
25/01/2017	48.75	0.72%	11.50%
26/01/2017	48.00	-1.55%	-24.59%
27/01/2017	46.75	-2.64%	-41.88%
30/01/2017	48.00	2.63%	41.76%
31/01/2017	48.35	0.74%	11.78%
01/02/2017	49.49	2.33%	37.02%
02/02/2017	51.24	3.47%	55.15%
03/02/2017	51.30	0.10%	1.60%
06/02/2017	49.20	-4.18%	-66.34%
07/02/2017	48.00	-2.47%	-39.23%

The 5% percentile of returns (the worst return in 20 days) is -4.18%. From this figure the VaR is deduced:

$$VaR(per\ MWh) = 0.0418 * \ 48 = 2.01\ €/MWh$$

This is a total $VaR = 2.01 * 50{,}000 = 100{,}500\ €$

$$Price(8\ Feb, 5\%) = 48 - 2.01 = 45.89\ €$$

Estimation directly from the distribution

This method can only be made to account for one iteration.

According to estimation data, volatility $\sigma_{day} = 2.26\%$ (daily), annually. If the prices follow a GBM, the change in the prices would be:

$Change\ in\ prices = \varepsilon\ \sigma\ S_{t-1}\ \sqrt{t}$

With $t = 1$ as volatility is daily. ε follows a standard normal distribution. The value of inverse standard normal distribution at 5% probability is $\varepsilon(0.05) = -1.64$

$$VaR(per\ MWh, 5\%, 1\ day) = 1.64 * 0.02266 * 48 = 1.78\ €/MWh$$

This is a total $VaR(\ 5\%, 1\ day) = 1.78 * 50{,}000 = 89{,}000\ €$

$$Price(8\ Feb, 5\%) = 48 - 1.78 = 46.22\ €$$

Estimation from Monte Carlo method

This method allows to compute the VaR for longer periods of time.

15,000 simulations of a GBM are made, with $F_0 = F(7\ Feb) = 48€$ and $\sigma_{annual} = \sigma_{day} = \sqrt{365}\ 2.26 = 43.1\%$.

Only the price of the following day is computed:

The average price of the 15,000 simulations for February 8 is 48.00 €, as it is logical.

The price in the 5% percentile is 46.22 €, therefore $VaR\ \ (per\ MWh, 5\%, 1\ day) = 48 - 46.22 = 1.78\ €/MWh$, logically coinciding with the estimation based on the distribution.

This is a total $VaR\ (5\%, 1\ day) = 1.78 * 50{,}000 = 89{,}000\ €$

CVaR (5%) is obtained rapidly in the Monte Carlo simulation tool from the average of prices below the 5% percentile:

$$Avg\ prices\ (Percentile < 5\%) = 45.79\ €/MWh$$

$$CVaR\ (per\ MWh, 5\%, 5\ days) = 48 - 45.79 = 2.21\ €/MWh$$

$$CVaR\ (\ 5\%, 5\ days) = 50{,}000 * 2.21 = 110{,}500\ €$$

VaR Case 2: Computation of VaR for 5 days after

Estimation from Monte Carlo method

One of the best characteristics of the Monte Carlo simulation is that it allows to easily calculate VaR for dates further in the future.

If the VaR in a 5-day period time is wanted, only a Monte Carlo simulation has to be implemented for this period. The future prices are modelled as a GBM of known drift and volatility. For the previous case, VaR on the MAR17 estimated at February 7, 2017, the same volatility $\sigma_{annual} = 0.431$ is considered (with zero drift).

15,000 simulations are made. The average price obtained is 48 €/MWh. The price in the 5% percentile is 44.06 €/MWh, therefore VaR:

$$VaR(per\ MWh, 5\ days) = 48 - 44.06 = 3.95\ €$$

$$VaR(\ 5\ days) = 3.95 * 50{,}000 = \ 197{,}500\ €$$

This is a logical result as prices there can be higher change in the prices in longer periods.

The CVar calculation yields:

$$Avg\ prices\ (Percentile < 5\%) = 43.134\ €/MWh$$

$$CVaR\ (per\ MWh, 5\%, 5\ days) = 48 - 43.13 = 4.87\ €/MWh$$

$$CVaR\ (\ 5\%, 5\ days) = 50{,}000 * 4.87\ \ = 243{,}500\ €$$

6. CONCLUSIONS

In this work a flexible Monte Carlo simulation tool has been implemented in VBA/Excel, which allows to calculate easily the value of future Asian options, including caps and floors of periods of different range. An Antithetic and a naïve MC methods have been implemented, with the antithetic consuming less time.

A number of different spot price models has been compared to the power and gas daily spot market prices. It has been observed that the GBM is a very flawed model to simulate electricity prices due to high volatilities, while Vasicek model is more robust and provides a better approximation for the valuation of the options. An ARIMA price model has also been implemented to value options. It is a methodology which is not common in literature, but it has proven to provide fairly accurate results to option valuation, providing that conditions do not change abruptly. It has been shown that Vasicek and ARIMA together can provide a sensible estimation of the fair value of an Asian option.

In the case of power spot markets, these two models overperform Vorst estimation for Asian options common GBM approaches, due to mainly high volatility in the prices.

Regarding gas prices, GBM behaves better thanks to lower volatilities, and to the fact that it allows to estimate and capture drift. For this reason Vorst estimation is expected to work well. Vasicek model is not as robust, as gas prices are not as mean-reverting as power prices, producing bigger errors.

The model allows to predict the probabilities of having certain prices, and calculate VaR and CVar for different time periods.

In the following steps, more complex spot price models can be integrated, for example Cox-Ingersoll-Ross, 2-factor models as Schwartz or ARMA-GARCH models.

7. REFERENCES

Bank for International Settlements, "Principles for the Sound Management of Operational Risk", June 2011, Basel.

BP, "BP Statistical Review of World Energy 2016", 65^{th} edition, 2016.

C. Chou, H. Lin, "Some Properties of CIR Processes", Stochastic Analysis and Applications, 24:4, 901-912, DOI: 10.1080/07362990600753643, 2006.

C. Johnson, "Value at risk: teoría y aplicaciones", Estudios de Economía. Vol. 28 - Nº 2, Diciembre 2001. Págs. 217-247

CORES, "Balance de producción y consumo de gas natural en España. Año 2015", 2016.

Dan Biebighauser, "Testing Random Number Generators", University of Minnesota - Twin Cities, REU Summer 2000

E. Colín Betancourt, "Modeling, Pricing and Hedging Derivatives on Natural Gas: an Analysis of the Influence of the Underlying Physical Market", Madrid, 2014.

E. Levy, "Pricing European average rate currency options", Journal of International Money and Finance, 11, 474-491, 1992.

E. Wiklund, "Asian Option Price and Volatility", Stockholm, 2012.

Economic Consulting Associates (ECA), "European Electricity Forward Markets and Hedging Products – State of Play and Elements for Monitoring", Multiple Framework Contract ACER/OP/DIR/08/2013/LOT 2/RFS 05, 2015.

Economic Consulting Associates, "European Electricity Forward Markets and Hedging Products – State of Play and Elements for Monitoring. Final Report." September 2015, Multiple Framework Contract ACER/OP/DIR/08/2013/LOT 2/RFS 05

European Commission, "Liquefied Natural Gas and gas storage will boost EU's energy security". Fact Sheet. 16 February 2016, Brussels.

G. Wynn, P. Coghe, "Europe's Coal-Fired Power Plants: Rough Times Ahead. Analysis of the Impact of a New Round of Pollution Controls", IEEFA, May 2017

H. Zhang, "Pricing Asian Options using Monte Carlo Methods", Uppsala University, U.U.D.M. Project Report 2009:7, 2009.

IGU (International Gas Union), "2016 World LNG Report. LNG 18 Conference & Exhibition Edition".

International Energy Agency (IEA), "Energy Policies of IEA Countries. Spain 2015 Review", July 2015, France.

International Energy Agency (IEA), "Key World Energy Statistics 2016", 2016, Paris.

J. Farid, "Calculating Value at Risk (VaR): VaR Methods", 2010.

J. Teusch, "Shale Gas and the EU Internal Gas Market: Beyond the hype and hysteria", CEPS Working Document No.369, September 2012.

K. Kladívko, "Maximum Likelihood Estimation of the Cox-Ingersoll-Ross Process: The Matlab Implementation".

K. P. N. Murthy, "An Introduction to Monte Carlo Simulations in Statistical Physics", Indira Gandhi Centre for Atomic Research, 2003.

L. Nielsen, "Pricing Asian Options", 2011.

M. Burger, B. Graebre, G. Shindlmayr, "Managing Energy Risk. A Practical Guide for Risk Management in Power, Gas and Other Energy Markets", 2^{nd} edition, Ed. Wiley, 2014.

M. Cartwright, "Trade in Ancient Greece", 18 January 2012.
http://www.ancient.eu/article/115/

MIBEL Regulatory Council, "Description of the Operation of the MIBEL", November 2009.

Nord Pool web page. http://www.nordpoolspot.com/

Oxford Dictionary O. https://en.oxforddictionaries.com/definition/commodity.

P. Jorion, "Methods to estimate VaR", 1996. https://merage.uci.edu/~jorion/oc/case4.html

PCR (Price Coupling of Regions), "PCR Project. Main features", August 2016.

R. Hyndman, "Thoughts of the Ljung-Box test", Hyndsight blog, January 2014.

R. Millán, "Los mercados de futuro de la electricidad", Endesa distribución eléctrica, 1996, Sevilla.

Realmarkits, "A Brief History of Derivatives"
http://www.realmarkits.com/derivatives/3.0history.php

Red Eléctrica Española, “El Sistema Eléctrico Español 2016”, 2017.

REE, “El Sistema Eléctrico Español 2013”, July 2014.

Sedigas (Asociación Española del Gas), “El Gas en España. Informe anual 2015”.

T. van den Berg, “Calibrating the Ornstein-Uhlenbeck (Vasicek) method”, 2011

T. Vorst, “Prices and Hedge Ratios of Average Exchange Rate Options”, International Review of Financial Analysis, 1(3): 179-193, 1992.

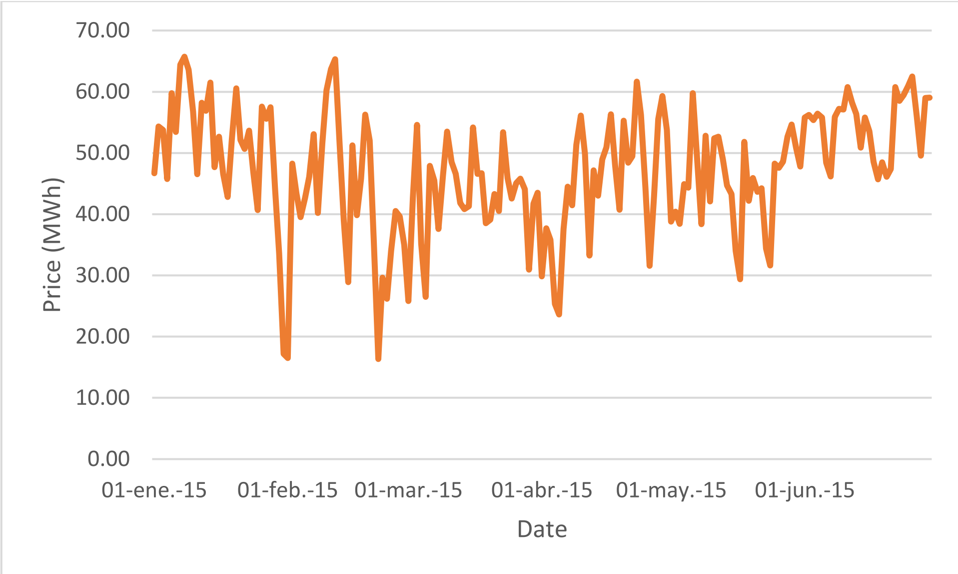
The University of Texas at Austin, “The History and Evolution of the U.S. Electricity Industry”, https://energy.utexas.edu/files/2016/09/UTAustin_FCe_History_2016.pdf

US Energy Information Administration (US EIA), “Annual Energy Outlook 2017 with projections to 2050”, January 5, 2017.

8. ANNEXES

8.1 Annex I: Fitting an ARIMA model (Practical Case)

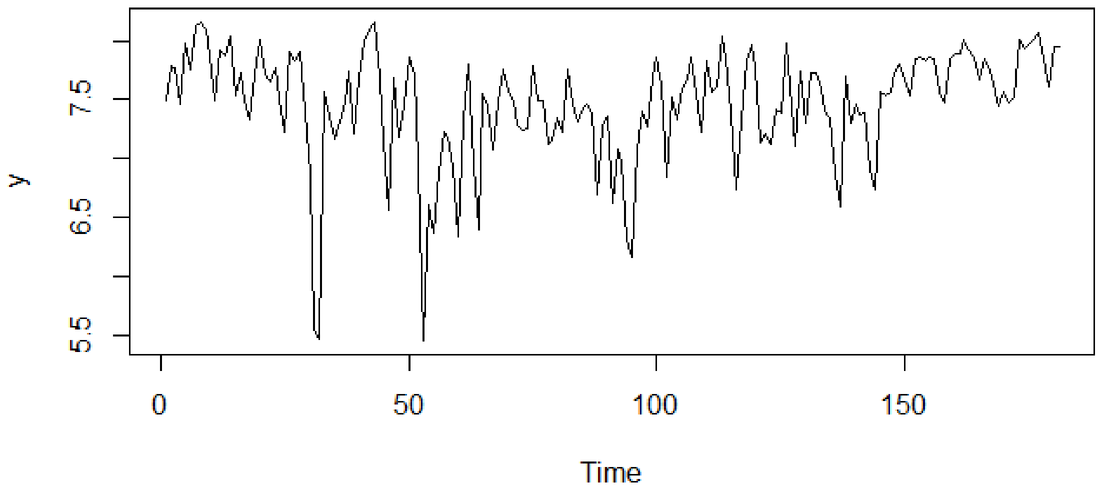
An ARIMA model in *R* has been fitted to the daily power spot prices of Spain during the first semester of 2015.



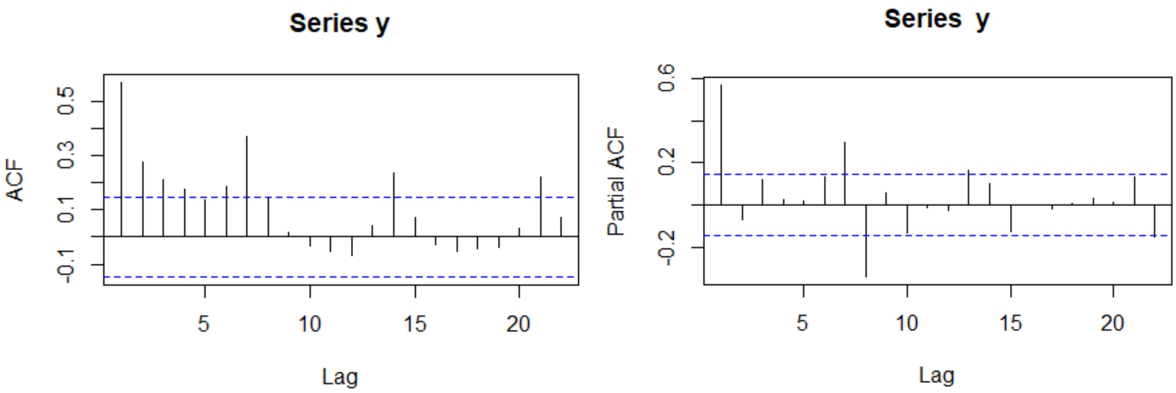
In order to adjust volatility. In order to stabilize variance, a Box-Cox transformation is made. *R* software delivers $\alpha = 1.95$ as the Box-Cox parameter by the log likelihood maximization method, applying thus:

$$y = 1.95 \ln(x)$$

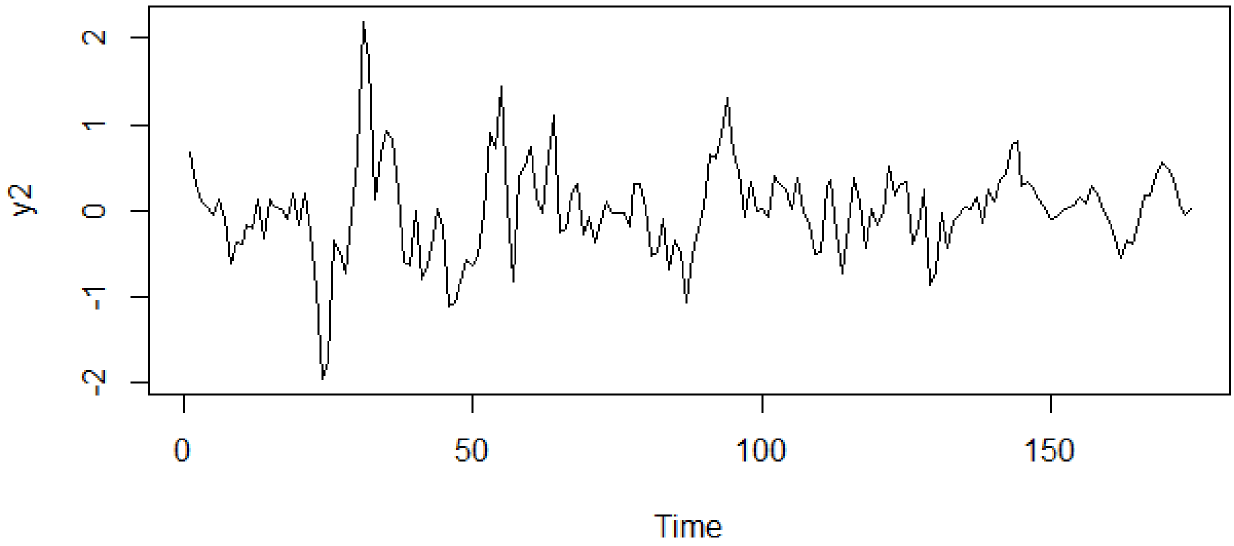
Where x is the original spot price series and y is the Box-Cox transformation. The transformed trime series is shown below:



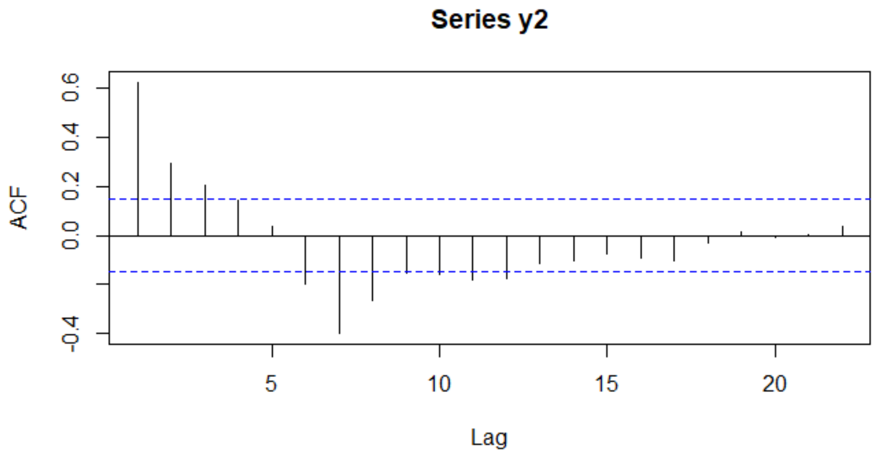
Still after having applied the Box-Cox transformation, there are still some concerns regarding volatility between days 30 and 50. Next, measures are taken to stabilize mean. The ACF and PCF are then analyzed to obtain an idea of the possible ARIMA model.

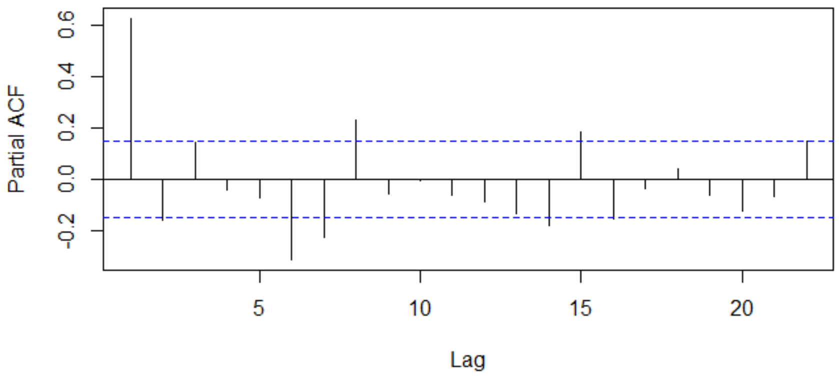


The declining trend in the lags multiple of 7 in the ACF suggests a weekly AR(1) seasonality. To solve it, seasonal differentiation is applied. The resulting time series is:



It is seen that after applying seasonal differences the mean has been stabilized. ACF and PCF of time series y2 are then studied again.





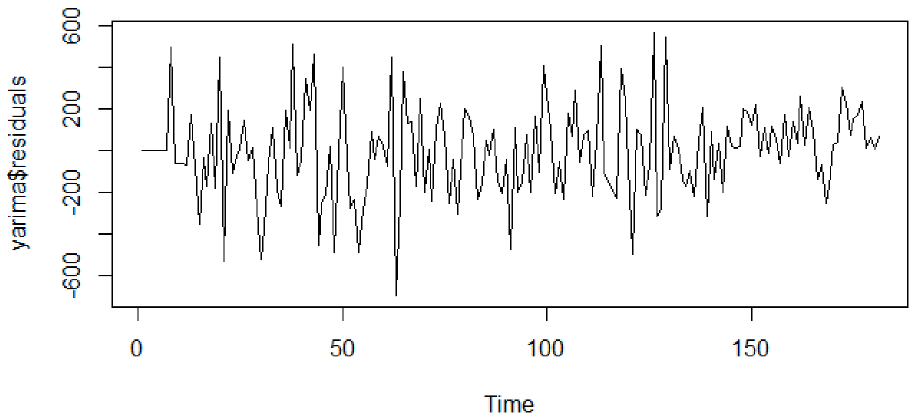
The downward trend of the first lags in the ACF suggests a weekly seasonal differencing process could explain some of the series. After applying this ARIMA model, the p-value of the coefficient (0) suggests it I an explanatory variable. By inspection, studying constantly the form, ACF and PACF of the residuals of the different ARIMA method trials, an ARIMA

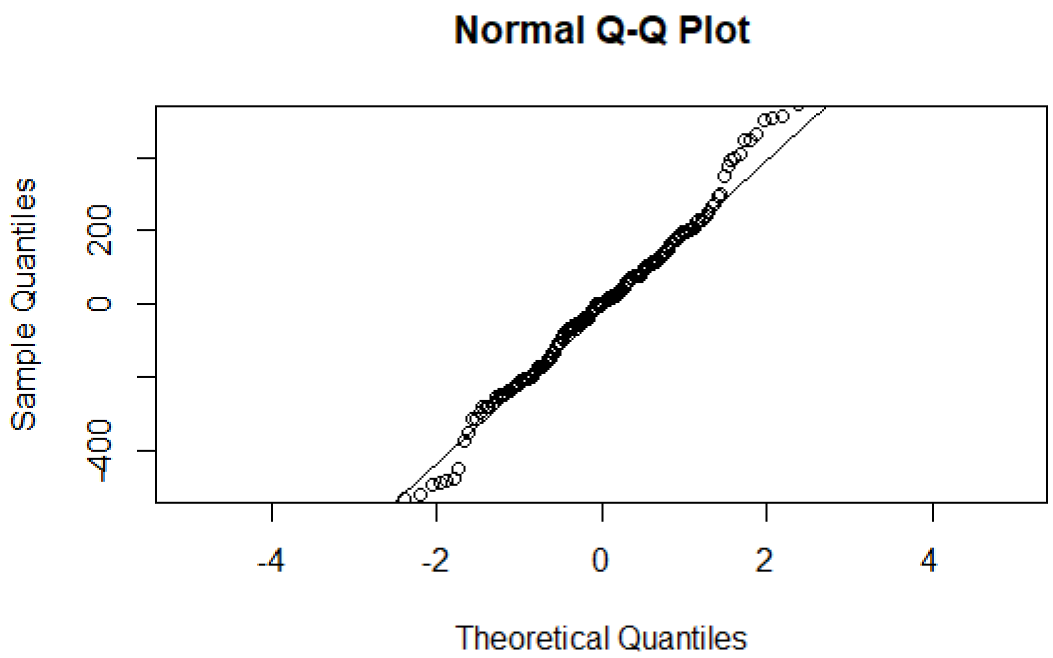
(1,0,2)(0,1,1)[7] applied on the Box-Cox transformation of the original time series is believed to be the best fit for the six-period month.

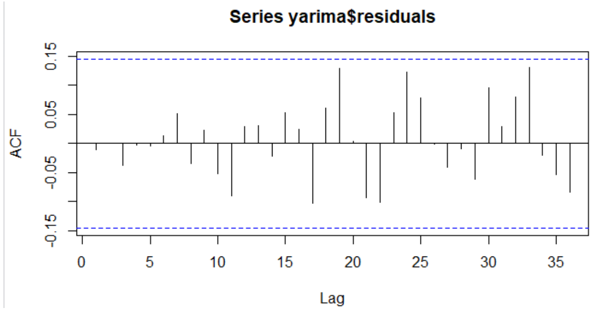
The significance of the coefficients is calculated:

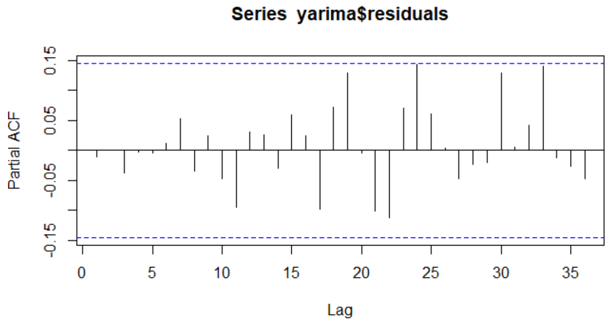
```
      Estimate  Std. Error  z value  Pr(>|z|)
ar1   0.894755   0.067434  13.2685 < 2.2e-16 ***
ma1  -0.189061   0.106723  -1.7715  0.076476 .
ma2  -0.260081   0.095613  -2.7201  0.006525 **
sma1 -0.999989   0.063213 -15.8192 < 2.2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

The residuals are normally distributed and independent according to their curve, the QQ plot, the ACF and the PCF.

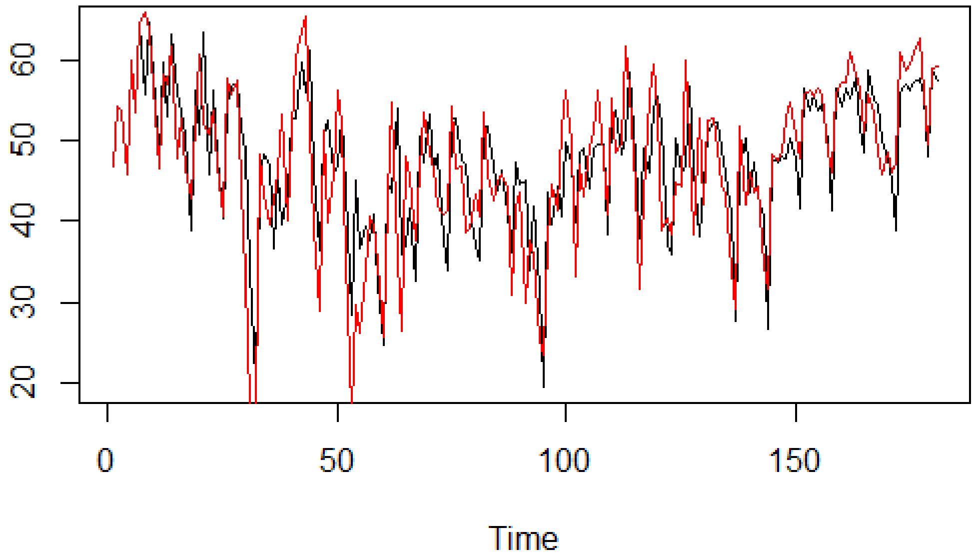








The fitted model is shown in black in the next figure, while the real prices are shown in red in the figure below:



Finally, the Ljung-Box test is applied. (Hyndman, 2014) suggests $h = 2\ m$ degrees of freedom, where m is the period of seasonality. Therefore 14 degrees of freedom are selected. The results of the Ljung-Box tests are:

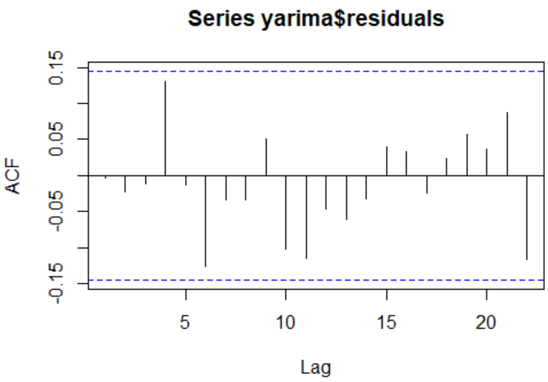
```
        Box-Ljung test

data:  yarima$residuals
X-squared = 3.7087, df = 14, p-value = 0.997
```

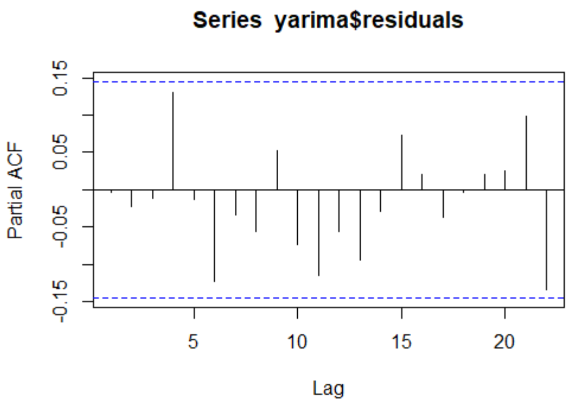
As the p-value is much larger than 0.05, the null hypothesis which states that the residuals are independently distributed cannot be rejected.

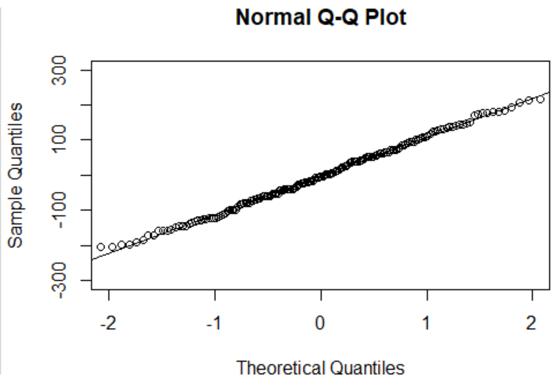
ARIMA model used to forecast of Asian monthly call option for 2015Q2

Calibration for period 2014Q1 & 2014Q2

```
z test of coefficients:

           Estimate Std. Error z value  Pr(>|z|)    
ar1        0.463746   0.097026  4.7796 1.757e-06 ***
ma1        0.289561   0.103633  2.7941  0.005204 ** 
sar1       0.432346   0.067907  6.3668 1.930e-10 ***
intercept 437.390853  32.909070 13.2909 < 2.2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```





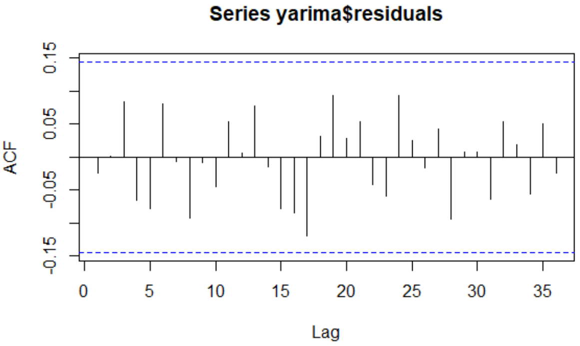


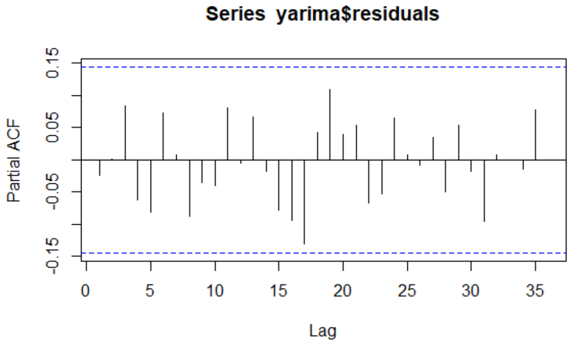

```
        Box-Ljung test

data:  yarima$residuals
X-squared = 13.267, df = 14, p-value = 0.5056
```

ARIMA model used to forecast of Asian monthly call option for 2015Q4

The ARIMA model trained during periods 2015Q2 and 2015Q3 an ARIMA (1,0,2)(0,1,1)[7].

```
        Estimate Std. Error z value  Pr(>|z|)
ar1    0.993098   0.019348 51.3271 < 2.2e-16 ***
ma1   -0.259737   0.072727 -3.5714 0.0003551 ***
ma2   -0.425327   0.075600 -5.6260 1.844e-08 ***
sma1  -0.999846   0.124636 -8.0221 1.039e-15 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```






```
        Box-Ljung test

data:  yarima$residuals
X-squared = 8.5323, df = 14, p-value = 0.8598
```

8.2 Annex II: Code

```
Sub Macro2()

Application.DisplayAlerts = False
Application.ScreenUpdating = False
Application.Calculation = xlCalculationManual
Application.EnableEvents = False


''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''



'Timer
    Dim PauseTime, Start, Finish, TotalTime
    Start = Timer    ' Set start time

    Call Eraser

    'Variable declaration
    Dim days_sim As Integer
    Dim nb_trials As Long
    Dim nb_trials_last As Integer
    Dim n_sheets As Integer
    Dim ref_spot_price As Double
    Dim annual_vol As Double
    Dim drift As Double
    Dim tenor As Double
    Dim acc_parameter_OU As Double
    Dim today_date As Date
    Dim select_val_period As String
    Dim nsim As Long

    Dim i As Integer
    Dim j As Integer
    Dim k As Integer
    Dim p As Integer

    ' Get inputs
    days_sim = Worksheets("MonteCarlo1").Range("B3").Value
    nb_trials = Worksheets("MonteCarlo1").Range("B4").Value
    annual_vol = Worksheets("MonteCarlo1").Range("B10").Value
    acc_parameter_OU = Worksheets("MonteCarlo1").Range("B12").Value
    tenor = Worksheets("GBM SPG").Range("B7").Value
    today_date = Worksheets("MonteCarlo1").Range("B2").Value
    drift = Worksheets("GBM SPG").Range("B9").Value

    n_sheets = nb_trials / 7500
    nb_trials_last = nb_trials Mod 7500
    If (nb_trials_last = 0) Then
        nb_trials_last = 7500
    End If
    nsim = CLng(15000 * (CLng(n_sheets) - 1) + CLng(nb_trials_last) * 2)

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''
''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''


    'Generate spot price for nb_trials
    Dim sim_MC_mode As String
    Dim select_model_PG As String
    Dim prev_price As Double
    Dim prev_price1 As Double
    Dim prev_price2 As Double
    ReDim PriceGen1(days_sim, 7500, n_sheets) As Double
```

```
ReDim PriceGen2(days_sim, 7500, n_sheets) As Double
ReDim PriceGen(days_sim, 7500, n_sheets) As Double
Dim TotAvgPrice As Double
Dim rand As Double
Dim path_numb As Integer

sim_MC_mode = Worksheets("MonteCarlo1").Range("B8").Value
select_model_PG = Worksheets("MonteCarlo1").Range("B6").Value

Select Case sim_MC_mode
  Case "Naive Monte Carlo"
    Call WriteDates(today_date, days_sim, n_sheets)        'Write dates for the period chosen
    Call WriteTitles(days_sim, nb_trials)       'Write titles
    Call WriteTrialsHeader(nb_trials)          'Write trials
    Select Case select_model_PG
      Case "Geometric Brownian"
        For j = 1 To nb_trials
          ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
          prev_price = ref_spot_price

          For i = 1 To days_sim
            rand = Rnd()
            PriceGen(i, j) = PriceGeneratorGBM(prev_price, annual_vol, tenor, drift, rand)
            prev_price = PriceGen(i, j)

            Worksheets("MonteCarlo1").Cells(i + 7, j + 4).Value = PriceGen(i, j)
          Next i
        Next j

      Case "OU Vasicek Mean Reversion"

      For j = 1 To nb_trials
        ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
        prev_price = ref_spot_price

        For i = 1 To days_sim
          rand = Rnd()
          PriceGen(i, j) = PriceGeneratorOU(ref_spot_price, prev_price, acc_parameter_OU, annual_vol, tenor, rand)
          prev_price = PriceGen(i, j)
          Worksheets("MonteCarlo1").Cells(i + 7, j + 4).Value = PriceGen(i, j)
        Next i
      Next j
    End Select

  path_numb = 1
  Call CalculateAsian(PriceGen(), nb_trials, days_sim, today_date, path_numb, n_sheets, nb_trials_last, nsim)
  'Call CalculateVaR(PriceGen(), nb_trials, days_sim
  Case "Antithetic Monte Carlo"

    If (n_sheets >= 2) Then
    ReDim sheet_exists(1 To n_sheets) As Double

      For j = 2 To n_sheets
        For i = 1 To Worksheets.Count
          If Worksheets(i).Name = "MonteCarlo" & j Then
            sheet_exists(j) = True
          End If
        Next i

        If Not sheet_exists(j) Then
        Worksheets.Add.Name = "MonteCarlo" & j
        End If
      Next j
    End If

    Call WriteDates(today_date, days_sim, n_sheets)        'Write dates for the period chosen
    Call WriteAntitheticTitles(days_sim, nb_trials_last, n_sheets)
```

```
Call WriteAntitheticTrialsHeader(nb_trials_last, n_sheets)           'Write trials
Select Case select_model_PG
  Case "Geometric Brownian"

    If n_sheets >= 2 Then
      For k = 1 To n_sheets - 1
        For j = 1 To 7500
        ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
        prev_price1 = ref_spot_price
        prev_price2 = ref_spot_price

          For i = 1 To days_sim
            rand = Rnd()
            PriceGen1(i, j, k) = PriceGeneratorGBM(prev_price1, annual_vol, tenor, drift, rand)
            prev_price1 = PriceGen1(i, j, k)
            rand = 1 - rand
            PriceGen2(i, j, k) = PriceGeneratorGBM(prev_price2, annual_vol, tenor, drift, rand)
            prev_price2 = PriceGen2(i, j, k)
            Worksheets("MonteCarlo" & k).Cells(i + 7, j + 4).Value = PriceGen1(i, j, k)
            Worksheets("MonteCarlo" & k).Cells(i + 7, j + 7 + 7500).Value = PriceGen2(i, j, k)
          Next i
        Next j
      Next k

      For j = 1 To nb_trials_last
        ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
        prev_price1 = ref_spot_price
        prev_price2 = ref_spot_price

        For i = 1 To days_sim
          rand = Rnd()
          PriceGen1(i, j, k) = PriceGeneratorGBM(prev_price1, annual_vol, tenor, drift, rand)
          prev_price1 = PriceGen1(i, j, k)
          rand = 1 - rand
          PriceGen2(i, j, k) = PriceGeneratorGBM(prev_price2, annual_vol, tenor, drift, rand)
          prev_price2 = PriceGen2(i, j, k)
          Worksheets("MonteCarlo" & k).Cells(i + 7, j + 4).Value = PriceGen1(i, j, n_sheets)
          Worksheets("MonteCarlo" & k).Cells(i + 7, j + 7 + nb_trials_last).Value = PriceGen2(i, j, n_sheets)
        Next i
      Next j

    ElseIf n_sheets = 1 Then
      k = 1
      For j = 1 To nb_trials_last
        ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
        prev_price1 = ref_spot_price
        prev_price2 = ref_spot_price

        For i = 1 To days_sim
          rand = Rnd()
          PriceGen1(i, j, k) = PriceGeneratorGBM(prev_price1, annual_vol, tenor, drift, rand)
          prev_price1 = PriceGen1(i, j, k)
          rand = 1 - rand
          PriceGen2(i, j, k) = PriceGeneratorGBM(prev_price2, annual_vol, tenor, drift, rand)
          prev_price2 = PriceGen2(i, j, k)
          Worksheets("MonteCarlo" & 1).Cells(i + 7, j + 4).Value = PriceGen1(i, j, n_sheets)
          Worksheets("MonteCarlo" & 1).Cells(i + 7, j + 7 + nb_trials_last).Value = PriceGen2(i, j, n_sheets)
        Next i
      Next j
    End If

  Case "OU Vasicek Mean Reversion"
    If n_sheets >= 2 Then
      For k = 1 To n_sheets - 1
        For j = 1 To 7500
          ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
          prev_price1 = ref_spot_price
```

```
                    prev_price2 = ref_spot_price

                    For i = 1 To days_sim
                       rand = Rnd()
                       PriceGen1(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price1, acc_parameter_OU, annual_vol,
tenor, rand)
                       prev_price1 = PriceGen1(i, j, k)
                       rand = 1 - rand
                       PriceGen2(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price2, acc_parameter_OU, annual_vol,
tenor, rand)
                       prev_price2 = PriceGen2(i, j, k)
                       Worksheets("MonteCarlo1").Cells(i + 7, j + 4).Value = PriceGen1(i, j, k)
                       Worksheets("MonteCarlo1").Cells(i + 7, j + 7 + 7500).Value = PriceGen2(i, j, k)
                    Next i
                 Next j
              Next k

              For j = 1 To nb_trials_last
                 ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
                 prev_price1 = ref_spot_price
                 prev_price2 = ref_spot_price
                 For i = 1 To days_sim
                    rand = Rnd()
                    PriceGen1(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price1, acc_parameter_OU, annual_vol,
tenor, rand)
                    prev_price1 = PriceGen1(i, j, k)
                    rand = 1 - rand
                    PriceGen2(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price2, acc_parameter_OU, annual_vol,
tenor, rand)
                    prev_price2 = PriceGen2(i, j, k)
                    Worksheets("MonteCarlo" & k).Cells(i + 7, j + 4).Value = PriceGen1(i, j, k)
                    Worksheets("MonteCarlo" & k).Cells(i + 7, j + 7 + nb_trials_last).Value = PriceGen2(i, j, k)
                 Next i
              Next j

           ElseIf n_sheets = 1 Then
              k = 1
              For j = 1 To nb_trials_last
                 ref_spot_price = Worksheets("MonteCarlo1").Range("B11").Value
                 prev_price1 = ref_spot_price
                 prev_price2 = ref_spot_price
                 For i = 1 To days_sim
                    rand = Rnd()
                    PriceGen1(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price1, acc_parameter_OU, annual_vol,
tenor, rand)
                    prev_price1 = PriceGen1(i, j, k)
                    rand = 1 - rand
                    PriceGen2(i, j, k) = PriceGeneratorOU(ref_spot_price, prev_price2, acc_parameter_OU, annual_vol,
tenor, rand)
                    prev_price2 = PriceGen2(i, j, k)
                    Worksheets("MonteCarlo" & k).Cells(i + 7, j + 4).Value = PriceGen1(i, j, k)
                    Worksheets("MonteCarlo" & k).Cells(i + 7, j + 7 + nb_trials_last).Value = PriceGen2(i, j, k)
                 Next i
              Next j
           End If


     End Select

     path_numb = 1
     Call CalculateAsian(PriceGen1(), nb_trials, days_sim, today_date, path_numb, n_sheets, nb_trials_last, nsim)
     path_numb = 2
     Call CalculateAsian(PriceGen2(), nb_trials, days_sim, today_date, path_numb, n_sheets, nb_trials_last, nsim)
     Call CalculateSTD

  End Select
```

```
    Application.DisplayAlerts = True
    Application.ScreenUpdating = True
    Application.Calculation = xlCalculationAutomatic
    Application.EnableEvents = True

    Finish = Timer                                    ' Set end time.
    TotalTime = Finish - Start                        ' Calculate total time.
    Worksheets("MonteCarlo1").Range("D1").Value = TotalTime
End Sub

''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''''

    'Calculate average spot price during the period & Asian option payoff valuation
Sub CalculateAsian(ByRef PriceGen() As Double, nb_trials As Long, days_sim As Integer, today_date As Date, path_numb
As Integer, n_sheets As Integer, nb_trials_last As Integer, nsim As Long)

    Dim final_date As Date

    Dim ini_square As Range
    Dim rolling_rng As Range
    Dim period_rng As Range
    Dim ini_row_period As Integer
    Dim final_row_period As Integer

    Dim tot_avg_period As Double
    Dim asian_payoff_period As Double

    Dim finalStdPayoff As Double

    Dim K_asian As Double
    Dim pos_payoff As Double
    Dim select_val_period As String
    Dim valuation_period As Double
    Dim count_weeks As Integer
    Dim count_months As Integer
    Dim count_years As Integer

    Dim count_periods As Integer

    Dim start_month As Integer
    Dim final_month As Integer
    Dim month_oneday As Integer
    Dim month_nextday As Integer

    Dim year_oneday As Integer
    Dim year_nextday As Integer

    ReDim AvgPrice(1 To 7500) As Double
    ReDim AsianPayoff(1 To 7500) As Double
    ReDim DeviationAsianPayoff(1 To 7500) As Double

    Dim finalAvgPayoff As Double

    Dim summary_average As Double
    Dim summary_sum_squares As Double
    Dim summaryStdev As Double


    'Get inputs

    final_date = today_date + days_sim - 1
    Worksheets("MonteCarlo1").Range("B7").Value = final_date

    K_asian = Worksheets("MonteCarlo1").Range("B44").Value
    select_val_period = Worksheets("MonteCarlo1").Range("B45").Value
```

```
    count_weeks = CountWeeks(days_sim)                                   'count weeks
    count_years = CountYears(today_date, final_date)                      'count years
    count_months = CountMonths(today_date, final_date, count_years)    'this function gets arguments from count_years!

    count_periods = CountPeriods(select_val_period, days_sim, count_weeks, count_months, _
                count_years)                                         'count periods

    'Call WriteGreeksHeaders(days_sim, nb_trials, count_periods)

    ReDim AvgPricePeriod(1 To (count_periods), 1 To nb_trials) As Double
    ReDim TotAvgPricePeriod(1 To (count_periods)) As Double

    ReDim AsianPayoffPeriod(1 To (count_periods), 1 To nb_trials) As Double
    ReDim AvgAsianPayoffPeriod(1 To (count_periods)) As Double

    'Call delta
    ReDim CDAsianPayoffPeriod(1 To (count_periods), 1 To nb_trials) As Double
    ReDim CDAvgAsianPayoffPeriod(1 To (count_periods)) As Double

    'Put delta
    ReDim PDAsianPayoffPeriod(1 To (count_periods), 1 To nb_trials) As Double
    ReDim PDAvgAsianPayoffPeriod(1 To (count_periods)) As Double

    Set ini_square = Range("D8")
    Set rolling_rng = Range("D7")
    rolling_rng.Select


    k = 1   'Initialization

    Select Case select_val_period
       Case "Day"

          If n_sheets >= 2 Then
             For p = 1 To n_sheets - 1
                Worksheets("MonteCarlo" & p).Activate
                For k = 1 To count_periods
                   Worksheets("MonteCarlo" & p).Cells((days_sim + k + 18 + count_periods), 4 + (path_numb - 1) * (7500 +
3)).Value = "Day " & k & " Asian Payoff"
                   For j = 1 To 7500
                      AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
                      CDAsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
                      Cells(days_sim + count_periods + k + 18, j + 4 + (path_numb - 1) * (7500 + 3)).Value =
AsianPayoffPeriod(k, j)
                   Next j

                   AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * (7500 + 3)), Cells(days_sim + k + 18 + count_periods, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
                   With Cells(days_sim + k + 18 + count_periods, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                        .Value = AvgAsianPayoffPeriod(k)
                        .Font.Bold = True
                   End With
                Next k
             Next p

             Worksheets("MonteCarlo" & p).Activate
             For k = 1 To count_periods
                Worksheets("MonteCarlo" & p).Cells((days_sim + k + 18 + count_periods), 4 + (path_numb - 1) * (nb_trials_last
+ 3)).Value = "Day " & k & " Asian Payoff"
                For j = 1 To nb_trials_last
                   AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
                   CDAsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
                   Cells(days_sim + count_periods + k + 18, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)
                Next j
```

```
            AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5 +
(path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, nb_trials_last + 4 + (path_numb - 1) *
(nb_trials_last + 3))))
            With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                .Value = AvgAsianPayoffPeriod(k)
                .Font.Bold = True
            End With
          Next k

        ElseIf n_sheets = 1 Then
          p = 1
          Worksheets("MonteCarlo" & p).Activate
          For k = 1 To count_periods
            Worksheets("MonteCarlo" & p).Cells((days_sim + k + 18 + count_periods), 4 + (path_numb - 1) * (nb_trials_last
+ 3)).Value = "Day " & k & " Asian Payoff"
            For j = 1 To nb_trials_last
              AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
              CDAsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, PriceGen(k, j, p) - K_asian)
              Cells(days_sim + count_periods + k + 18, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)
            Next j

            AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5 +
(path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, nb_trials_last + 4 + (path_numb - 1) *
(nb_trials_last + 3))))
            With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                .Value = AvgAsianPayoffPeriod(k)
                .Font.Bold = True
            End With
          Next k
        End If

      Case "Week"

        If n_sheets >= 2 Then
          For p = 1 To n_sheets - 1
            Worksheets("MonteCarlo" & p).Activate
            k = 1
            Set ini_square = Range("D8")
            Set rolling_rng = Range("D7")
            rolling_rng.Select

            For i = 1 To days_sim - 1
              Set rolling_rng = rolling_rng.Offset(1, 0)

              If (StrComp(Cells(i + 7 + 1, 3).Value, "lu.", vbTextCompare) = 0) Then
                Range("D" & (days_sim + k + 16)).Value = "Week " & k & " Avg"
                Range("D" & (days_sim + k + 18 + count_periods)).Value = "Week " & k & " Asian Payoff"
                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To 7500
                  AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb -
1) * (7500 + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (7500 + 3))))
                  Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (7500 + 3)).Value = AvgPricePeriod(k, j)
                  TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb -
1) * (7500 + 3)), Cells(days_sim + k + 16, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
                  With Cells(days_sim + k + 16, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                    .Value = TotAvgPricePeriod(k)
                    .Font.Bold = True
                  End With
                  AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                  Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (7500 + 3)).Value =
AsianPayoffPeriod(k, j)

                Next j
```

```
                    AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods,
5 + (path_numb - 1) * _
                        (7500 + 3)), Cells(days_sim + k + 18 + count_periods, (7500 + 4) + (path_numb - 1) * (7500 + 3))))

                    With Cells(days_sim + k + 18 + count_periods, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                        .Value = AvgAsianPayoffPeriod(k)
                        .Font.Bold = True
                    End With

'                   Set period_rng = Range(ini_square, rolling_rng)
'                   period_rng.Select

                    Set ini_square = rolling_rng.Offset(1, 0)
                    k = k + 1
                End If

            Next i
        Next p
        Worksheets("MonteCarlo" & p).Activate
        k = 1
        Set ini_square = Range("D8")
        Set rolling_rng = Range("D7")

        For i = 1 To days_sim - 1
            Set rolling_rng = rolling_rng.Offset(1, 0)

            If (StrComp(Cells(i + 7 + 1, 3).Value, "lu.", vbTextCompare) = 0) Then
                Range("D" & (days_sim + k + 16)).Value = "Week " & k & " Avg"
                Range("D" & (days_sim + k + 18 + count_periods)).Value = "Week " & k & " Asian Payoff"
                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To nb_trials_last
                    AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
                    TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1)
* (nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                        .Value = TotAvgPricePeriod(k)
                        .Font.Bold = True
                    End With
                    AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                    Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)

                Next j

                AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * _
                    (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

                With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                    .Value = AvgAsianPayoffPeriod(k)
                    .Font.Bold = True
                End With

'               Set period_rng = Range(ini_square, rolling_rng)
'               period_rng.Select

                Set ini_square = rolling_rng.Offset(1, 0)
                k = k + 1
            End If
        Next i

    ElseIf n_sheets = 1 Then
```

```
            For i = 1 To days_sim - 1
               Set rolling_rng = rolling_rng.Offset(1, 0)

               If (StrComp(Cells(i + 7 + 1, 3).Value, "lu.", vbTextCompare) = 0) Then
                  Range("D" & (days_sim + k + 16)).Value = "Week " & k & " Avg"
                  Range("D" & (days_sim + k + 18 + count_periods)).Value = "Week " & k & " Asian Payoff"
                  ini_row_period = ini_square.Row
                  final_row_period = rolling_rng.Row

                  For j = 1 To nb_trials_last
                     AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                     Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
                     TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1)
* (nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                     With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                        .Value = TotAvgPricePeriod(k)
                        .Font.Bold = True
                     End With
                     AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                     Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)

                  Next j

                  AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * _
                        (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

                  With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                     .Value = AvgAsianPayoffPeriod(k)
                     .Font.Bold = True
                  End With

'                 Set period_rng = Range(ini_square, rolling_rng)
'                 period_rng.Select

                  Set ini_square = rolling_rng.Offset(1, 0)
                  k = k + 1
               End If
            Next i
         End If


   Case "Month"
      If n_sheets >= 2 Then
         For p = 1 To n_sheets - 1
            Worksheets("MonteCarlo" & p).Activate
            k = 1
            Set ini_square = Range("D8")
            Set rolling_rng = Range("D7")

            For i = 1 To days_sim - 1
               Set rolling_rng = rolling_rng.Offset(1, 0)
               month_oneday = Month(Cells(i + 7, "D").Value)
               month_nextday = Month(Cells(i + 8, "D").Value)

               If month_oneday <> month_nextday Then
                  With Range("D" & (days_sim + k + 16))
                     .Value = "Month " & k & " Avg"
                     .Font.Bold = True
                  End With

                  With Range("D" & (days_sim + k + 18 + count_periods))
                     .Value = "Month " & k & " Asian Payoff"
                     .Font.Bold = True
```

```
            End With

            ini_row_period = ini_square.Row
            final_row_period = rolling_rng.Row

            For j = 1 To 7500
              AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(7500 + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (7500 + 3))))
              Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (7500 + 3)).Value = AvgPricePeriod(k, j)
              TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1)
* (7500 + 3)), Cells(days_sim + k + 16, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
              With Cells(days_sim + k + 16, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                .Value = TotAvgPricePeriod(k)
                .Font.Bold = True
              End With
              AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
              Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (7500 + 3)).Value =
AsianPayoffPeriod(k, j)

            Next j

            AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * _
                (7500 + 3)), Cells(days_sim + k + 18 + count_periods, (7500 + 4) + (path_numb - 1) * (7500 + 3))))

            With Cells(days_sim + k + 18 + count_periods, 7500 + 5 + (path_numb - 1) * (7500 + 3))
              .Value = AvgAsianPayoffPeriod(k)
              .Font.Bold = True
            End With

            Set period_rng = Range(ini_square, rolling_rng)
            period_rng.Select

            Set ini_square = rolling_rng.Offset(1, 0)
            k = k + 1
          End If
        Next i
      Next p

      Worksheets("MonteCarlo" & p).Activate
      k = 1
      Set ini_square = Range("D8")
      Set rolling_rng = Range("D7")
      For i = 1 To days_sim - 1
        Set rolling_rng = rolling_rng.Offset(1, 0)
        month_oneday = Month(Cells(i + 7, "D").Value)
        month_nextday = Month(Cells(i + 8, "D").Value)

        If month_oneday <> month_nextday Then
          With Range("D" & (days_sim + k + 16))
            .Value = "Month " & k & " Avg"
            .Font.Bold = True
          End With

          With Range("D" & (days_sim + k + 18 + count_periods))
            .Value = "Month " & k & " Asian Payoff"
            .Font.Bold = True
          End With

          ini_row_period = ini_square.Row
          final_row_period = rolling_rng.Row

          For j = 1 To nb_trials_last
            AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
            Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
```

```
                TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                    .Value = TotAvgPricePeriod(k)
                    .Font.Bold = True
                End With
                AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)

            Next j

            AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5 +
(path_numb - 1) * _
                (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

            With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                .Value = AvgAsianPayoffPeriod(k)
                .Font.Bold = True
            End With

            Set period_rng = Range(ini_square, rolling_rng)
            period_rng.Select

            Set ini_square = rolling_rng.Offset(1, 0)
            k = k + 1
        End If
    Next i
ElseIf n_sheets = 1 Then
        Worksheets("MonteCarlo1" & p).Activate
        k = 1
        Set ini_square = Range("D8")
        Set rolling_rng = Range("D7")
        For i = 1 To days_sim - 1
            Set rolling_rng = rolling_rng.Offset(1, 0)
            month_oneday = Month(Cells(i + 7, "D").Value)
            month_nextday = Month(Cells(i + 8, "D").Value)

            If month_oneday <> month_nextday Then
                With Range("D" & (days_sim + k + 16))
                    .Value = "Month " & k & " Avg"
                    .Font.Bold = True
                End With

                With Range("D" & (days_sim + k + 18 + count_periods))
                    .Value = "Month " & k & " Asian Payoff"
                    .Font.Bold = True
                End With

                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To nb_trials_last
                    AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
                    TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1)
* (nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                        .Value = TotAvgPricePeriod(k)
                        .Font.Bold = True
                    End With
                    AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                    Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)
```

```
                Next j

                AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * _
                    (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

                With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                  .Value = AvgAsianPayoffPeriod(k)
                  .Font.Bold = True
                End With

                Set period_rng = Range(ini_square, rolling_rng)
                period_rng.Select

                Set ini_square = rolling_rng.Offset(1, 0)
                k = k + 1
              End If
            Next i
        End If

        Case "Year"
          If n_sheets >= 2 Then
          For p = 1 To n_sheets - 1
            Set ini_square = Range("D8")
            Set rolling_rng = Range("D7")

            Worksheets("MonteCarlo" & p).Activate
            k = 1
            For i = 1 To days_sim - 1
              Set rolling_rng = rolling_rng.Offset(1, 0)
              year_oneday = Year(Cells(i + 7, "D").Value)
              year_nextday = Year(Cells(i + 8, "D").Value)

              If year_oneday <> year_nextday Then
                With Range("D" & (days_sim + k + 16))
                  .Value = "Year " & k & " Avg"
                  .Font.Bold = True
                End With

                With Range("D" & (days_sim + k + 18 + count_periods))
                  .Value = "Year " & k & " Asian Payoff"
                  .Font.Bold = True
                End With

                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To 7500
                  AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(7500 + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (7500 + 3))))
                  Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (7500 + 3)).Value = AvgPricePeriod(k, j)
                  TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1)
* (7500 + 3)), Cells(days_sim + k + 16, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
                  With Cells(days_sim + k + 16, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                    .Value = TotAvgPricePeriod(k)
                    .Font.Bold = True
                  End With
                  AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                  Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (7500 + 3)).Value =
AsianPayoffPeriod(k, j)

                Next j

                AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5
+ (path_numb - 1) * _
                    (7500 + 3)), Cells(days_sim + k + 18 + count_periods, (7500 + 4) + (path_numb - 1) * (7500 + 3))))
```

```
                    With Cells(days_sim + k + 18 + count_periods, 7500 + 5 + (path_numb - 1) * (7500 + 3))
                        .Value = AvgAsianPayoffPeriod(k)
                        .Font.Bold = True
                    End With

                    'Set period_rng = Range(ini_square, rolling_rng)
                    'period_rng.Select

                    Set ini_square = rolling_rng.Offset(1, 0)
                    k = k + 1
                End If
            Next i
        Next p
        Worksheets("MonteCarlo" & p).Activate
        k = 1
        Set ini_square = Range("D8")
        Set rolling_rng = Range("D7")
        For i = 1 To days_sim - 1
            Set rolling_rng = rolling_rng.Offset(1, 0)
            year_oneday = Year(Cells(i + 7, "D").Value)
            year_nextday = Year(Cells(i + 8, "D").Value)

            If year_oneday <> year_nextday Then
                With Range("D" & (days_sim + k + 16))
                    .Value = "Year " & k & " Avg"
                    .Font.Bold = True
                End With

                With Range("D" & (days_sim + k + 18 + count_periods))
                    .Value = "Year " & k & " Asian Payoff"
                    .Font.Bold = True
                End With

                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To nb_trials_last
                    AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
                    TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                        .Value = TotAvgPricePeriod(k)
                        .Font.Bold = True
                    End With
                    AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                    Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)

                Next j

                AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5 +
(path_numb - 1) * _
                    (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

                With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                    .Value = AvgAsianPayoffPeriod(k)
                    .Font.Bold = True
                End With

                Set period_rng = Range(ini_square, rolling_rng)
                period_rng.Select

                Set ini_square = rolling_rng.Offset(1, 0)
```

```
                k = k + 1
            End If
        Next i
    ElseIf n_sheets = 1 Then
        Worksheets("MonteCarlo1").Activate
        k = 1
        Set ini_square = Range("D8")
        Set rolling_rng = Range("D7")
        For i = 1 To days_sim - 1
            Set rolling_rng = rolling_rng.Offset(1, 0)
            year_oneday = Year(Cells(i + 7, "D").Value)
            year_nextday = Year(Cells(i + 8, "D").Value)

            If year_oneday <> year_nextday Then
                With Range("D" & (days_sim + k + 16))
                    .Value = "Year " & k & " Avg"
                    .Font.Bold = True
                End With

                With Range("D" & (days_sim + k + 18 + count_periods))
                    .Value = "Year " & k & " Asian Payoff"
                    .Font.Bold = True
                End With

                ini_row_period = ini_square.Row
                final_row_period = rolling_rng.Row

                For j = 1 To nb_trials_last
                    AvgPricePeriod(k, j) = WorksheetFunction.Average(Range(Cells(ini_row_period, j + 4 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(final_row_period, j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    Cells(days_sim + k + 16, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPricePeriod(k, j)
                    TotAvgPricePeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 16, 5 + (path_numb - 1) *
(nb_trials_last + 3)), Cells(days_sim + k + 16, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
                    With Cells(days_sim + k + 16, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                        .Value = TotAvgPricePeriod(k)
                        .Font.Bold = True
                    End With
                    AsianPayoffPeriod(k, j) = WorksheetFunction.Max(0, AvgPricePeriod(k, j) - K_asian)
                    Cells(days_sim + k + 18 + count_periods, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
AsianPayoffPeriod(k, j)

                Next j

                AvgAsianPayoffPeriod(k) = WorksheetFunction.Average(Range(Cells(days_sim + k + 18 + count_periods, 5 +
(path_numb - 1) * _
                    (nb_trials_last + 3)), Cells(days_sim + k + 18 + count_periods, (nb_trials_last + 4) + (path_numb - 1) *
(nb_trials_last + 3))))

                With Cells(days_sim + k + 18 + count_periods, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
                    .Value = AvgAsianPayoffPeriod(k)
                    .Font.Bold = True
                End With

                Set period_rng = Range(ini_square, rolling_rng)
                period_rng.Select

                Set ini_square = rolling_rng.Offset(1, 0)
                k = k + 1
            End If
        Next i
    End If


End Select

'Calculation Asian option payoff
```

```
ReDim AvgPriceInterval(count_periods, nb_trials) As Double

If n_sheets >= 2 Then
    For p = 1 To n_sheets - 1
        Worksheets("MonteCarlo" & p).Activate
        For j = 1 To 7500
            For i = 1 To days_sim
                'Worksheets("RandNumb").Cells(i + 1, j + 3).Value = WorksheetFunction.Round(Worksheets("RandNumb").Cells(i + 1, j + 1), 2)

            Next i
            AvgPrice(j) = WorksheetFunction.Average(Range(Cells(8, j + 4 + (path_numb - 1) * (7500 + 3)), Cells((days_sim + 7), j + 4 + (path_numb - 1) * (7500 + 3))))
            Worksheets("MonteCarlo" & p).Cells(days_sim + 10, j + 4 + (path_numb - 1) * (7500 + 3)).Value = AvgPrice(j)
        Next j

        For j = 1 To 7500
            AsianPayoff(j) = WorksheetFunction.Max(0, AvgPrice(j) - K_asian)
            Worksheets("MonteCarlo" & p).Cells(days_sim + 12, j + 4 + (path_numb - 1) * (7500 + 3)).Value = AsianPayoff(j)

        Next j

        TotAvgPrice = WorksheetFunction.Average(Range(Cells(days_sim + 10, 5 + (path_numb - 1) * (7500 + 3)), Cells(days_sim + 10, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
        With Worksheets("MonteCarlo" & p).Cells(days_sim + 10, 7500 + 5 + (path_numb - 1) * (7500 + 3))
            .Value = TotAvgPrice
            .Font.Bold = True
        End With


        finalAvgPayoff = WorksheetFunction.Average(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (7500 + 3)), Cells(days_sim + 12, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
        With Worksheets("MonteCarlo" & p).Cells(days_sim + 12, 7500 + 5 + (path_numb - 1) * (7500 + 3))
            .Value = finalAvgPayoff
            .Font.Bold = True
        End With

        Dim meanfinalAvgPayoff As Double

        If (Range("B8").Value = "Antithetic Monte Carlo") Then
           meanfinalAvgPayoff = 0.5 * (Cells(days_sim + 12, 7500 + 5) + Cells(days_sim + 12, 2 * 7500 + 8))
           finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5), Cells(days_sim + 12, 7500 + 4)), _
                            Range(Cells(days_sim + 12, 8 + 7500), Cells(days_sim + 12, 2 * 7500 + 7)))
        Else
            finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (7500 + 3)), Cells(days_sim + 12, 7500 + 4 + (path_numb - 1) * (7500 + 3))))
        End If


        With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 4)
           .Value = finalStdPayoff
           .Font.Bold = True
        End With

        With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 3)
           .Value = "Std payoff"
           .Font.Bold = True
        End With

        With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 3)
           .Value = "LBound ConfInt"
           .Font.Bold = True
        End With

        If Range("B8").Value = "Antithetic Monte Carlo" And path_numb = 2 Then

           With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
```

```
            .Value = meanfinalAvgPayoff - 1.96 * finalStdPayoff / Sqr(2 * 7500)
            .Font.Bold = True
         End With

         With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
            .Value = meanfinalAvgPayoff + 1.96 * finalStdPayoff / Sqr(2 * 7500)
            .Font.Bold = True
         End With
      ElseIf (Range("B8").Value = "Naive Monte Carlo") Then
         With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
         .Value = finalAvgPayoff - 1.96 * finalStdPayoff / Sqr(7500)
         .Font.Bold = True
         End With

         With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
            .Value = finalAvgPayoff + 1.96 * finalStdPayoff / Sqr(7500)
            .Font.Bold = True
         End With
      End If
      With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 3)
         .Value = "UBound ConfInt"
         .Font.Bold = True
      End With

      With Cells(days_sim + 26 + 2 * count_periods, "E")
         .Value = "SINGLE PATH - SHEET " & p
         .Font.Bold = True
         .Font.Size = 20
      End With

      If (Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo") Then
         Range("D" & (days_sim + 28 + 2 * count_periods)).Value = "Asian option payoff Whole period"


         For j = 1 To 7500
               Cells(days_sim + 28 + 2 * count_periods, j + 4).Value = _
                  0.5 * (Cells(days_sim + 12, j + 4).Value _
                  + Cells(days_sim + 12, j + 4 + (7500 + 3)).Value)
         Next j

         With Cells(days_sim + 28 + 2 * count_periods, 7500 + 5)
            .Value = WorksheetFunction.Average(Range(Cells(days_sim + 28 + 2 * count_periods, 5), Cells(days_sim + 28
+ 2 * count_periods, 7500 + 4)))
            .Font.Bold = True
         End With


         For k = 1 To count_periods
            Range("D" & (days_sim + k + 30 + 2 * count_periods)).Value = select_val_period & k & " Asian Payoff"
            For j = 1 To 7500
               Cells(days_sim + k + 30 + 2 * count_periods, j + 4).Value = _
                  0.5 * (Cells(days_sim + k + 18 + count_periods, j + 4).Value _
                  + Cells(days_sim + k + 18 + count_periods, j + 4 + (7500 + 3)).Value)
            Next j
            With Cells(days_sim + k + 30 + 2 * count_periods, 7500 + 5)
            .Value = WorksheetFunction.Average(Range(Cells(days_sim + k + 30 + 2 * count_periods, 5), Cells(days_sim +
k + 30 + 2 * count_periods, 7500 + 4)))
            .Font.Bold = True
         End With
         Next k
      End If
   Next p

   Worksheets("MonteCarlo" & p).Activate
   For j = 1 To nb_trials_last
      For i = 1 To days_sim
```

```
            'Worksheets("RandNumb").Cells(i + 1, j + 3).Value = WorksheetFunction.Round(Worksheets("RandNumb").Cells(i + 1, j + 1), 2)

          Next i
          AvgPrice(j) = WorksheetFunction.Average(Range(Cells(8, j + 4 + (path_numb - 1) * (nb_trials_last + 3)), Cells((days_sim + 7), j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
          Worksheets("MonteCarlo" & p).Cells(days_sim + 10, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPrice(j)
       Next j

       For j = 1 To nb_trials_last
           AsianPayoff(j) = WorksheetFunction.Max(0, AvgPrice(j) - K_asian)
           Worksheets("MonteCarlo" & p).Cells(days_sim + 12, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AsianPayoff(j)

       Next j

       TotAvgPrice = WorksheetFunction.Average(Range(Cells(days_sim + 10, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 10, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
       With Worksheets("MonteCarlo" & p).Cells(days_sim + 10, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
           .Value = TotAvgPrice
           .Font.Bold = True
       End With


       finalAvgPayoff = WorksheetFunction.Average(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 12, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
       With Worksheets("MonteCarlo" & p).Cells(days_sim + 12, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
           .Value = finalAvgPayoff
           .Font.Bold = True
       End With

       'Dim meanfinalAvgPayoff As Double

       If Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo" Then
          Worksheets("MonteCarlo" & p).Activate
          meanfinalAvgPayoff = 0.5 * (Cells(days_sim + 12, nb_trials_last + 5) + Cells(days_sim + 12, 2 * nb_trials_last + 8))
          finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5), Cells(days_sim + 12, nb_trials_last + 4)), _
                               Range(Cells(days_sim + 12, 8 + nb_trials_last), Cells(days_sim + 12, 2 * nb_trials_last + 7)))
       Else
           finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 12, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
       End If


       With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 4)
          .Value = finalStdPayoff
          .Font.Bold = True
       End With

       With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 3)
          .Value = "Std payoff"
          .Font.Bold = True
       End With

       With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 3)
          .Value = "LBound ConfInt"
          .Font.Bold = True
       End With

       If Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo" And path_numb = 2 Then

          With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
             .Value = meanfinalAvgPayoff - 1.96 * finalStdPayoff / Sqr(2 * nb_trials_last)
             .Font.Bold = True
          End With
```

```
            With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
                .Value = meanfinalAvgPayoff + 1.96 * finalStdPayoff / Sqr(2 * nb_trials_last)
                .Font.Bold = True
            End With
        ElseIf (Range("B8").Value = "Naive MonteCarlo") Then
            With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
                .Value = finalAvgPayoff - 1.96 * finalStdPayoff / Sqr(nb_trials_last)
                .Font.Bold = True
            End With

            With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
                .Value = finalAvgPayoff + 1.96 * finalStdPayoff / Sqr(nb_trials_last)
                .Font.Bold = True
            End With
        End If
        With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 3)
            .Value = "UBound ConfInt"
            .Font.Bold = True
        End With

        With Cells(days_sim + 26 + 2 * count_periods, "E")
            .Value = "SINGLE PATH"
            .Font.Bold = True
            .Font.Size = 20
        End With

        If (Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo") Then
            Worksheets("MonteCarlo" & p).Range("D" & (days_sim + 28 + 2 * count_periods)).Value = "Asian option payoff
Whole period"

            For j = 1 To nb_trials_last
                    Cells(days_sim + 28 + 2 * count_periods, j + 4).Value = _
                        0.5 * (Cells(days_sim + 12, j + 4).Value _
                        + Cells(days_sim + 12, j + 4 + (nb_trials_last + 3)).Value)
            Next j

            With Cells(days_sim + 28 + 2 * count_periods, nb_trials_last + 5)
                .Value = WorksheetFunction.Average(Range(Cells(days_sim + 28 + 2 * count_periods, 5), Cells(days_sim + 28 +
2 * count_periods, nb_trials_last + 4)))
                .Font.Bold = True
            End With


            For k = 1 To count_periods
                Range("D" & (days_sim + k + 30 + 2 * count_periods)).Value = select_val_period & k & " Asian Payoff"
                For j = 1 To nb_trials_last
                    Cells(days_sim + k + 30 + 2 * count_periods, j + 4).Value = _
                        0.5 * (Cells(days_sim + k + 18 + count_periods, j + 4).Value _
                        + Cells(days_sim + k + 18 + count_periods, j + 4 + (nb_trials_last + 3)).Value)
                Next j
                With Cells(days_sim + k + 30 + 2 * count_periods, nb_trials_last + 5)
                .Value = WorksheetFunction.Average(Range(Cells(days_sim + k + 30 + 2 * count_periods, 5), Cells(days_sim + k
+ 30 + 2 * count_periods, nb_trials_last + 4)))
                .Font.Bold = True
            End With
            Next k
        End If

    ElseIf n_sheets = 1 Then
        p = 1
        For j = 1 To nb_trials_last
             For i = 1 To days_sim
                 'Worksheets("RandNumb").Cells(i + 1, j + 3).Value =
WorksheetFunction.Round(Worksheets("RandNumb").Cells(i + 1, j + 1), 2)

             Next i
```

```
        AvgPrice(j) = WorksheetFunction.Average(Worksheets("MonteCarlo" & p).Range(Cells(8, j + 4 + (path_numb - 1) * (nb_trials_last + 3)), Cells((days_sim + 7), j + 4 + (path_numb - 1) * (nb_trials_last + 3))))
        Worksheets("MonteCarlo" & p).Cells(days_sim + 10, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AvgPrice(j)
    Next j

    For j = 1 To nb_trials_last
        AsianPayoff(j) = WorksheetFunction.Max(0, AvgPrice(j) - K_asian)
        Worksheets("MonteCarlo" & p).Cells(days_sim + 12, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value = AsianPayoff(j)

    Next j

    TotAvgPrice = WorksheetFunction.Average(Range(Cells(days_sim + 10, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 10, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
    With Worksheets("MonteCarlo" & p).Cells(days_sim + 10, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
        .Value = TotAvgPrice
        .Font.Bold = True
    End With


    finalAvgPayoff = WorksheetFunction.Average(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 12, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
    With Worksheets("MonteCarlo" & p).Cells(days_sim + 12, nb_trials_last + 5 + (path_numb - 1) * (nb_trials_last + 3))
        .Value = finalAvgPayoff
        .Font.Bold = True
    End With

    'Dim meanfinalAvgPayoff As Double

    If (Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo") Then
       Worksheets("MonteCarlo" & p).Activate
       meanfinalAvgPayoff = 0.5 * (Cells(days_sim + 12, nb_trials_last + 5) + Cells(days_sim + 12, 2 * nb_trials_last + 8))
       finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5), Cells(days_sim + 12, nb_trials_last + 4)), _
                      Range(Cells(days_sim + 12, 8 + nb_trials_last), Cells(days_sim + 12, 2 * nb_trials_last + 7)))
    Else
        finalStdPayoff = WorksheetFunction.StDev_S(Range(Cells(days_sim + 12, 5 + (path_numb - 1) * (nb_trials_last + 3)), Cells(days_sim + 12, nb_trials_last + 4 + (path_numb - 1) * (nb_trials_last + 3))))
    End If


    With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 4)
       .Value = finalStdPayoff
       .Font.Bold = True
    End With

    With Worksheets("MonteCarlo" & p).Cells(days_sim + 14, 3)
       .Value = "Std payoff"
       .Font.Bold = True
    End With

    With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 3)
       .Value = "LBound ConfInt"
       .Font.Bold = True
    End With

    If Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo" And path_numb = 2 Then
       Worksheets("MonteCarlo" & p).Activate
       With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
          .Value = meanfinalAvgPayoff - 1.96 * finalStdPayoff / Sqr(2 * nb_trials_last)
          .Font.Bold = True
       End With

       With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
          .Value = meanfinalAvgPayoff + 1.96 * finalStdPayoff / Sqr(2 * nb_trials_last)
          .Font.Bold = True
```

```
            End With
        ElseIf (Range("B8").Value = "Naive MonteCarlo") Then
                    With Worksheets("MonteCarlo" & p).Cells(days_sim + 15, 4)
                .Value = finalAvgPayoff - 1.96 * finalStdPayoff / Sqr(nb_trials_last)
                .Font.Bold = True
            End With

            With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 4)
                .Value = finalAvgPayoff + 1.96 * finalStdPayoff / Sqr(nb_trials_last)
                .Font.Bold = True
            End With
        End If
        With Worksheets("MonteCarlo" & p).Cells(days_sim + 16, 3)
            .Value = "UBound ConfInt"
            .Font.Bold = True
        End With

        With Cells(days_sim + 26 + 2 * count_periods, "E")
            .Value = "SINGLE PATH"
            .Font.Bold = True
            .Font.Size = 16
        End With

        If (Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo") Then
            Worksheets("MonteCarlo" & p).Activate
            Range("D" & (days_sim + 28 + 2 * count_periods)).Value = "Asian option payoff Whole period"

            For j = 1 To nb_trials_last
                    Cells(days_sim + 28 + 2 * count_periods, j + 4).Value = _
                        0.5 * (Cells(days_sim + 12, j + 4).Value _
                        + Cells(days_sim + 12, j + 4 + (nb_trials_last + 3)).Value)
            Next j

            With Cells(days_sim + 28 + 2 * count_periods, nb_trials_last + 5)
                .Value = WorksheetFunction.Average(Range(Cells(days_sim + 28 + 2 * count_periods, 5), Cells(days_sim + 28 +
2 * count_periods, nb_trials_last + 4)))
                .Font.Bold = True
            End With

            For k = 1 To count_periods
                Range("D" & (days_sim + k + 30 + 2 * count_periods)).Value = select_val_period & k & " Asian Payoff"
                For j = 1 To nb_trials_last
                    Cells(days_sim + k + 30 + 2 * count_periods, j + 4).Value = _
                        0.5 * (Cells(days_sim + k + 18 + count_periods, j + 4).Value _
                        + Cells(days_sim + k + 18 + count_periods, j + 4 + (nb_trials_last + 3)).Value)
                Next j
            With Cells(days_sim + k + 30 + 2 * count_periods, nb_trials_last + 5)
                .Value = WorksheetFunction.Average(Range(Cells(days_sim + k + 30 + 2 * count_periods, 5), Cells(days_sim + k
+ 30 + 2 * count_periods, nb_trials_last + 4)))
                .Font.Bold = True
            End With
            Next k
        End If
    End If

             ''SUMMARY RESULTS
    If (Worksheets("MonteCarlo1").Range("B8").Value = "Antithetic Monte Carlo") Then
        With Worksheets("MonteCarlo1").Cells(days_sim + k + 34 + 3 * count_periods, 4)
            .Value = "SUMMARY RESULTS"
            .Font.Bold = True
            .Font.Size = 20
        End With


        For p = 1 To n_sheets
            summary_average = summary_average + Worksheets("MonteCarlo" & p).Cells(days_sim + 28 + 2 * count_periods,
nb_trials_last + 5)
```

```
        Next p
        summary_average = summary_average / n_sheets
        With Worksheets("MonteCarlo1").Cells(days_sim + k + 36 + 3 * count_periods, 3)
            .Value = "SUMMARY AVERAGE"
            .Font.Bold = True
            .Font.Size = 12
        End With
        With Worksheets("MonteCarlo1").Cells(days_sim + k + 36 + 3 * count_periods, 4)
            .Value = summary_average
            .Font.Bold = True
            .Font.Size = 12
        End With


        "Deviation in each Asian payoff
        With Worksheets("MonteCarlo1").Cells(days_sim + 13, 4)
            .Value = "Asian payoff deviation"
            .Font.Bold = True
            .Font.Size = 12
        End With

        For p = 1 To n_sheets

            Worksheets("MonteCarlo" & p).Activate
            If (p < n_sheets) Then
                For path_numb = 1 To 2
                    For j = 1 To nb_trials_last
                        DeviationAsianPayoff(j) = summary_average - Cells(days_sim + 12, j + 4 + (path_numb - 1) * (7500 +
3)).Value
                        Worksheets("MonteCarlo" & p).Cells(days_sim + 13, j + 4 + (path_numb - 1) * (7500 + 3)).Value =
DeviationAsianPayoff(j)
                    Next j
                Next path_numb
            Else
                For path_numb = 1 To 2
                    For j = 1 To nb_trials_last
                        DeviationAsianPayoff(j) = summary_average - Cells(days_sim + 12, j + 4 + (path_numb - 1) * (nb_trials_last
+ 3)).Value
                        Worksheets("MonteCarlo" & p).Cells(days_sim + 13, j + 4 + (path_numb - 1) * (nb_trials_last + 3)).Value =
DeviationAsianPayoff(j)
                    Next j
                Next path_numb
            End If
        Next p

'       With Worksheets("MonteCarlo1").Cells(days_sim + k + 37 + 3 * count_periods, 3)
'           .Value = "SUMMARY STD DEVIATION"
'           .Font.Bold = True
'           .Font.Size = 12
'       End With
'
'       summary_sum_squares = 0
'       For p = 1 To n_sheets
'           Worksheets("MonteCarlo" & p).Activate
'           If (p < n_sheets) Then
'               summary_sum_squares = summary_sum_squares + WorksheetFunction.SumSq( _
'                   Range(Cells(days_sim + 13, 5), Cells(days_sim + 13, 7500 + 4)), _
'                   Range(Cells(days_sim + 13, 8 + 7500), Cells(days_sim + 13, 2 * 7500 + 7)))
'           Else
'           Worksheets("MonteCarlo" & p).Activate
'               summary_sum_squares = summary_sum_squares + WorksheetFunction.SumSq( _
'                   Worksheets("MonteCarlo" & p).Range(Cells(days_sim + 13, 5), Cells(days_sim + 13, nb_trials_last + 4)), _
'                       Worksheets("MonteCarlo" & p).Range(Cells(days_sim + 13, 8 + nb_trials_last), Cells(days_sim + 13, 2 *
nb_trials_last + 7)))
'           End If
'       Next p
'       nsim = CLng(15000 * (n_sheets - 1) + nb_trials_last * 2)
```

```
'       summaryStdev = Sqr(summary_sum_squares / (nsim - 1))
'
'       With Worksheets("MonteCarlo1").Cells(days_sim + k + 37 + 3 * count_periods, 4)
'           .Value = summaryStdev
'           .Font.Bold = True
'           .Font.Size = 12
'       End With
    End If

'   With Worksheets("MonteCarlo1").Cells(days_sim + k + 38 + 3 * count_periods, 3)
'       .Value = "LOWER CONF.INT. BOUND"
'       .Font.Bold = True
'       .Font.Size = 12
'   End With
'
'   With Worksheets("MonteCarlo1").Cells(days_sim + k + 38 + 3 * count_periods, 4)
'       .Value = summary_average - summaryStdev * 1.96 / Sqr(nsim)
'       .Font.Bold = True
'       .Font.Size = 12
'   End With
'
'
'   With Worksheets("MonteCarlo1").Cells(days_sim + k + 39 + 3 * count_periods, 3)
'       .Value = "UPPER CONF.INT. BOUND"
'       .Font.Bold = True
'       .Font.Size = 12
'   End With
'
'   With Worksheets("MonteCarlo1").Cells(days_sim + k + 39 + 3 * count_periods, 4)
'       .Value = summary_average + summaryStdev * 1.96 / Sqr(nsim)
'       .Font.Bold = True
'       .Font.Size = 12
'   End With

    'nsim = 15000 * (n_sheets - 1) + nb_trials_last * 2

    With Worksheets("MonteCarlo1").Cells(days_sim + k + 40 + 3 * count_periods, 3)
        .Value = "APROX STD DEVIATION"
        .Font.Bold = True
        .Font.Size = 12
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + k + 40 + 3 * count_periods, 4)
        .Value = Worksheets("MonteCarlo1").Cells(days_sim + 14, 4)
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + k + 41 + 3 * count_periods, 3)
        .Value = "APPROX LOWER CONF.INT. BOUND"
        .Font.Bold = True
        .Font.Size = 12
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + k + 41 + 3 * count_periods, 4)
        .Value = CDbl(summary_average) - CDbl(1.96) * CDbl(Worksheets("MonteCarlo1").Cells(days_sim + 14, 4)) /
Sqr(CDbl(nsim))
        .Font.Bold = True
        .Font.Size = 12
    End With


    With Worksheets("MonteCarlo1").Cells(days_sim + k + 42 + 3 * count_periods, 3)
        .Value = "APPROX UPPER CONF.INT. BOUND"
        .Font.Bold = True
        .Font.Size = 12
    End With
```

```
    With Worksheets("MonteCarlo1").Cells(days_sim + k + 42 + 3 * count_periods, 4)
        .Value = summary_average + 1.96 * CDbl(Worksheets("MonteCarlo1").Cells(days_sim + 14, 4)) / Sqr(CDbl(nsim))
        .Font.Bold = True
        .Font.Size = 12
    End With


    If (p = n_sheets) Then
        MsgBox (summary_average)
    End If
End Sub






Function PriceGeneratorGBM(prev_price As Double, annual_vol As Double, tenor As Double, drift As Double, rand As
Double) As Double
    Dim price As Double
    Dim nrm_var As Double

    If rand = 0 Then
        rand = 0.0001
    End If
    If rand = 1 Then
        rand = 0.9999
    End If
    price = prev_price * Exp((drift - 0.5 * (annual_vol) ^ (2)) * tenor + annual_vol * WorksheetFunction.NormSInv(rand) *
Sqr(tenor))
    PriceGeneratorGBM = price
End Function
Function PriceGeneratorOU(ref_spot_price As Double, prev_price As Double, beta As Double, _
            annual_vol As Double, tenor As Double, rand As Double) As Double
    Dim price As Double
    Dim nrm_var As Double

    Dim rev_term As Double
    Dim stoch_term As Double
    Dim incr_price As Double

    If rand = 0 Then
        rand = 0.0001
    End If
    If rand = 1 Then
        rand = 0.9999
    End If
    rev_term = beta * (ref_spot_price - prev_price) * tenor
    stoch_term = annual_vol * WorksheetFunction.NormSInv(rand) * Sqr(tenor)

    incr_price = rev_term + stoch_term
    price = prev_price + incr_price
    PriceGeneratorOU = price

End Function

Sub Eraser()

    Range("E6", "AAA6").Clear
    Range("C8", "AAA400").Clear
    Range("E7", Range("D7").End(xlToRight)).Clear
    Range("C8", Range("C8").End(xlToRight).End(xlDown).End(xlDown).End(xlDown)).Clear
    Range("D8", "BBB1000").Clear

    Worksheets("Daily Return").Range("A1", "AA200").Clear
End Sub
```

```
Sub WriteTitles(days_sim As Integer, nb_trials As Long)
    With Worksheets("MonteCarlo1").Range("E6")
        .Value = "Spot price generator"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Range("D7")
        .Value = "Date"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(6, nb_trials + 8)
        .Value = "Returns"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Range("C7")
        .Value = "Weekday"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 12, 4)
        .Value = "Asian option payoff"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 10, nb_trials + 7)
        .Value = "Avg Return"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 11, nb_trials + 7)
        .Value = "stdReturn"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 14, nb_trials + 7)
        .Value = "VaR"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 10, 4)
        .Value = "Average"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 9, nb_trials + 5)
        .Value = "Tot. Avg"
        .Font.Bold = True
    End With
End Sub

Sub WriteAntitheticTitles(days_sim As Integer, nb_trials_last As Integer, n_sheets As Integer)

    With Worksheets("MonteCarlo1").Range("E6")
            .Value = "Spot price generator PATH1"
            .Font.Bold = True
        End With

        With Worksheets("MonteCarlo1").Range("D7")
            .Value = "Date"
            .Font.Bold = True
        End With

        With Worksheets("MonteCarlo1").Cells(6, 7500 + 8)
            .Value = "Spot price generator PATH2"
```

```
        .Font.Bold = True
    End With

'    With Worksheets("MonteCarlo1").Cells(6, 2 * 7500 + 11)
'        .Value = "Returns PATH 1"
'        .Font.Bold = True
'    End With
'
'        With Worksheets("MonteCarlo1").Cells(6, 3 * 7500 + 14)
'        .Value = "Returns PATH 2"
'        .Font.Bold = True
'    End With

    With Worksheets("MonteCarlo1").Range("C7")
        .Value = "Weekday"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 12, 4)
        .Value = "Asian option payoff Path 1"
        .Font.Bold = True
    End With

'    With Worksheets("MonteCarlo1").Cells(days_sim + 10, 2 * 7500 + 10)
'        .Value = "Avg Return Path 1"
'        .Font.Bold = True
'    End With
'
'    With Worksheets("MonteCarlo1").Cells(days_sim + 11, 2 * 7500 + 10)
'        .Value = "stdReturn Path 1"
'        .Font.Bold = True
'    End With
'
'    With Worksheets("MonteCarlo1").Cells(days_sim + 14, 2 * 7500 + 10)
'        .Value = "VaR Path 1"
'        .Font.Bold = True
'    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 10, 4)
        .Value = "Average Path 1"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 9, 7500 + 5)
        .Value = "Tot. Avg Path 1"
        .Font.Bold = True
    End With

    With Worksheets("MonteCarlo1").Cells(days_sim + 12, 7 + 7500)
        .Value = "Asian option payoff Path 2"
        .Font.Bold = True
    End With

'    With Worksheets("MonteCarlo1").Cells(days_sim + 10, 3 * 7500 + 13)
'        .Value = "Avg Return Path 2"
'        .Font.Bold = True
'    End With
'
'    With Worksheets("MonteCarlo1").Cells(days_sim + 11, 3 * 7500 + 13)
'        .Value = "stdReturn Path 2"
'        .Font.Bold = True
'    End With
'
'    With Worksheets("MonteCarlo1").Cells(days_sim + 14, 3 * 7500 + 13)
'        .Value = "VaR Path 2"
'        .Font.Bold = True
'    End With
```

```
    With Worksheets("MonteCarlo1").Cells(days_sim + 10, 7 + 7500)
        .Value = "Average Path 2"
        .Font.Bold = True
    End With

'    With Worksheets("MonteCarlo1").Cells(days_sim + 9, 2 * 7500 + 8)
'        .Value = "Tot. Avg Path 2"
'        .Font.Bold = True
'    End With

If n_sheets >= 2 Then
    For i = 1 To n_sheets
        With Worksheets("MonteCarlo" & n_sheets).Range("E6")
            .Value = "Spot price generator PATH1"
            .Font.Bold = True
        End With

        With Worksheets("MonteCarlo" & n_sheets).Range("D7")
            .Value = "Date"
            .Font.Bold = True
        End With

        With Worksheets("MonteCarlo" & n_sheets).Cells(6, nb_trials_last + 8)
            .Value = "Spot price generator PATH2"
            .Font.Bold = True
        End With

    '    With Worksheets("MonteCarlo"&n_sheets).Cells(6, 2 * nb_trials_last + 11)
    '        .Value = "Returns PATH 1"
    '        .Font.Bold = True
    '    End With
    '
    '        With Worksheets("MonteCarlo"&n_sheets).Cells(6, 3 * nb_trials_last + 14)
    '        .Value = "Returns PATH 2"
    '        .Font.Bold = True
    '    End With

        With Worksheets("MonteCarlo" & n_sheets).Range("C7")
            .Value = "Weekday"
            .Font.Bold = True
        End With

        With Worksheets("MonteCarlo" & n_sheets).Cells(days_sim + 12, 4)
            .Value = "Asian option payoff Path 1"
            .Font.Bold = True
        End With

    '    With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 10, 2 * nb_trials_last + 10)
    '        .Value = "Avg Return Path 1"
    '        .Font.Bold = True
    '    End With
    '
    '    With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 11, 2 * nb_trials_last + 10)
    '        .Value = "stdReturn Path 1"
    '        .Font.Bold = True
    '    End With
    '
    '    With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 14, 2 * nb_trials_last + 10)
    '        .Value = "VaR Path 1"
    '        .Font.Bold = True
    '    End With

        With Worksheets("MonteCarlo" & n_sheets).Cells(days_sim + 10, 4)
            .Value = "Average Path 1"
            .Font.Bold = True
        End With
```

```
            With Worksheets("MonteCarlo" & n_sheets).Cells(days_sim + 9, nb_trials_last + 5)
                .Value = "Tot. Avg Path 1"
                .Font.Bold = True
            End With

            With Worksheets("MonteCarlo" & n_sheets).Cells(days_sim + 12, 7 + nb_trials_last)
                .Value = "Asian option payoff Path 2"
                .Font.Bold = True
            End With

        '   With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 10, 3 * nb_trials_last + 13)
        '       .Value = "Avg Return Path 2"
        '       .Font.Bold = True
        '   End With
        '
        '   With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 11, 3 * nb_trials_last + 13)
        '       .Value = "stdReturn Path 2"
        '       .Font.Bold = True
        '   End With
        '
        '   With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 14, 3 * nb_trials_last + 13)
        '       .Value = "VaR Path 2"
        '       .Font.Bold = True
        '   End With

            With Worksheets("MonteCarlo" & n_sheets).Cells(days_sim + 10, 7 + nb_trials_last)
                .Value = "Average Path 2"
                .Font.Bold = True
            End With

        '   With Worksheets("MonteCarlo"&n_sheets).Cells(days_sim + 9, 2 * nb_trials_last + 8)
        '       .Value = "Tot. Avg Path 2"
        '       .Font.Bold = True
        '   End With
        Next i
    End If

End Sub



Sub WriteDates(today_date As Date, days_sim As Integer, n_sheets As Integer)
    For p = 1 To n_sheets
        Worksheets("MonteCarlo" & p).Activate
        For i = 1 To days_sim
            If i = 1 Then
                With Cells(i + 7, "D")
                    .Value = today_date
                    .Font.Bold = True
                End With

                With Cells(i + 7, "C")
                    .Value = WeekdayName(Weekday(today_date), True, vbSunday)
                    .Font.Bold = True
                End With

            Else
                With Cells(i + 7, "D")
                    .Value = DateAdd("d", 1, Worksheets("MonteCarlo1").Cells(i + 6, "D").Value)
                    .Font.Bold = True
                End With

                With Cells(i + 7, "C")
                .Value = WeekdayName(Weekday(Cells(i + 7, "D")), True, vbSunday)
                .Font.Bold = True
```

```
            End With
        End If
    Next i
  Next p
End Sub

Sub WriteTrialsHeader(nb_trials As Long)

  For j = 1 To nb_trials
    With Cells(7, j + 4)
      .Value = "T" & j
      .Font.Bold = True
      .HorizontalAlignment = xlCenter
    End With

  Next j
End Sub

Sub WriteAntitheticTrialsHeader(nb_trials_last As Integer, n_sheets As Integer)
  If n_sheets >= 2 Then
    For i = 1 To n_sheets - 1
      For j = 1 To 7500
        With Worksheets("MonteCarlo" & i).Cells(7, j + 4)
          .Value = "T" & j
          .Font.Bold = True
          .HorizontalAlignment = xlCenter
        End With


        With Worksheets("MonteCarlo" & i).Cells(7, j + 7500 + 7)
          .Value = "T" & j
          .Font.Bold = True
          .HorizontalAlignment = xlCenter
        End With
      Next j
    Next i

    For j = 1 To nb_trials_last
      With Worksheets("MonteCarlo" & n_sheets).Cells(7, j + 4)
      .Value = "T" & j
      .Font.Bold = True
      .HorizontalAlignment = xlCenter
      End With

      With Worksheets("MonteCarlo" & n_sheets).Cells(7, j + nb_trials_last + 7)
      .Value = "T" & j
      .Font.Bold = True
      .HorizontalAlignment = xlCenter
      End With

    Next j

  ElseIf n_sheets = 1 Then
    For j = 1 To nb_trials_last
      With Worksheets("MonteCarlo" & n_sheets).Cells(7, j + 4)
      .Value = "T" & j
      .Font.Bold = True
      .HorizontalAlignment = xlCenter
      End With

      With Worksheets("MonteCarlo" & n_sheets).Cells(7, j + nb_trials_last + 7)
      .Value = "T" & j
      .Font.Bold = True
      .HorizontalAlignment = xlCenter
      End With

    Next j
```

```
    End If

End Sub

Function CountWeeks(days_sim As Integer)
    Dim count_weeks As Integer

    count_weeks = 1
    For i = 1 To days_sim
        If (StrComp(Cells(i + 7 + 1, 3), "lu.", vbTextCompare) = 0) Then
        count_weeks = count_weeks + 1
        End If
    Next i
    CountWeeks = count_weeks
End Function

Function CountYears(today_date As Date, final_date As Date)
    Dim start_year As Integer

    start_year = Year(today_date)
    final_year = Year(final_date)
    CountYears = final_year - start_year + 1
End Function

Function CountMonths(today_date As Date, final_date As Date, count_years As Integer)
    Dim start_month As Integer
    Dim final_month As Integer

    start_month = Month(today_date)
    final_month = Month(final_date)
    CountMonths = final_month - start_month + 1 + (count_years - 1) * 12
    Range("H1").Value = CountMonths

End Function

Function CountPeriods(select_val_period As String, days_sim As Integer, count_weeks As Integer, _
        count_months As Integer, count_years As Integer)

    If (StrComp(select_val_period, "Day", vbTextCompare) = 0) Then
            CountPeriods = days_sim
        ElseIf (StrComp(select_val_period, "Week", vbTextCompare) = 0) Then
            CountPeriods = count_weeks
        ElseIf (StrComp(select_val_period, "Month", vbTextCompare) = 0) Then
            CountPeriods = count_months
        ElseIf (StrComp(select_val_period, "Year", vbTextCompare) = 0) Then
            CountPeriods = count_years
    End If

End Function
```