

UiO : Department of Mathematics
University of Oslo

KRIGING THE POWER FUTURES CURVE AND PRICING OF OPTIONS

Jonas Enger Melsnes

Master's Thesis, Spring 2019



This master's thesis is submitted under the master's programme *Stochastic Modelling, Statistics and Risk Analysis*, with programme option *Finance, Insurance and Risk*, at the Department of Mathematics, University of Oslo. The scope of the thesis is 60 credits.

The front page depicts a section of the root system of the exceptional Lie group E_8 , projected into the plane. Lie groups were invented by the Norwegian mathematician Sophus Lie (1842–1899) to express symmetries in differential equations and today they play a central role in various parts of mathematics.

Preface

Power futures are delivered over a period as opposed to a fixed delivery date, common for many commodities. In this thesis we use kriging to fit a curve to find daily points for these contract prices. Kriging also gives us an error-function, and this makes each price a normal distribution.

The Black model is used for pricing of options on futures contract. Usually, this model takes in a price for the futures contract together with other parameters, and calculates a fair price from this. We will look at what happens when this futures price is a distribution, and how we can use the quantification of its error. Our results show that also the option price will be normally distributed.

The work on this thesis has been ongoing from February 2018 to May 2019. There are several people who I would like to thank you contributed to the work of this thesis in some way. First and foremost, I would like to express my gratitude for my supervisor Fred Espen Benth, who provided valuable insight and advice during this process and also his flexibility and support even when I was not in the country.

I would also like to thank the fellow students of B1002 for the company, conversation and distractions that they provided me with.

Lastly i would like to give a sincere thank you to my family and close friends who provided me with unwavering moral support and motivation throughout studies.

Contents

Contents	ii
1 Introduction	1
2 The Power Market	2
3 Kriging	5
3.1 Gaussian Random Field and Kriging	5
3.2 Proof of the Simple Kriging form	7
3.3 Semivariogram and the Nugget Effect	10
4 Estimating the Kriged Curve of Futures prices	13
4.1 Estimating the Mean Function	13
4.2 Kriging of Estimated Curve	17
5 Analysis of Parameters and Data	20
5.1 Parameters	20
5.2 Generation of a Gaussian Random Field	27
5.3 Change in Data	29
6 Pricing of Put and Call-options	33
6.1 European options and the Black-76 Formula	33
6.2 Plots of the Option Price	37
6.3 Uncertainty of the Option Price	38
A R-code	41
References	60

1 Introduction

In this thesis we use kriging to predict a curve for the price on power futures. This is a technique from geostatistics which also gives us an estimate for the error of the prediction. This way our futures price will be a distribution.

We will then consider the prices for options on such futures contracts. The option price will necessarily also be a distribution, and we will look deeper into how the error transfers from futures price to option price.

In the beginning of the thesis we introduce the power market together with the different types of contracts we're considering. The power prices varies a lot. Both production and usage, is very dependent on the weather, and therefore the price shows a strong seasonality. We will look at this, and other reasons for variation.

After this we will present a proof of the form of kriging. This is based on the proof in [CW11], but with slight changes. As power is delivered throughout a period, and not everything at once as many other commodities, the covariances of the contracts will be different.

We estimate the kriged curve in the following chapter. First we remove seasonality, and then estimate our mean function, the Nelson-Siegel curve, and then use our models to find the predicted curve.

We then want to look deeper into what elements makes up the curve. In chapter 5 we look at the different parameters of the kriged curve to see what role they all have, and how important our estimate of them are. We will, by generating a field, see if we can be certain of our methods. In the end of this chapter we're looking at how a change in contracts prices affect the curve.

In the last chapter we move over to the pricing of options. We are going to use the Black-76 formula, so we will present this and see that also the option price is normally distributed, when the futures curve is. Then we'll look at the relative variance of the option price in relation to that of the futures contract.

2 The Power Market

In this thesis we will be dealing with futures contracts in the power market. Here we will present power markets, focused on NordPool and EEX, which is the one we have collected data from.

It's possible to trade power in futures contracts, the curve of such contracts is what we want to find an estimation for in this thesis, so we will present what it is. We will also talk about options on these futures contracts.

The problem of seasonality is a bigger issue when trading power than many other commodities, and we will have to take this into account when modelling the curves in later chapters. At the end of this chapter we will present the problem and explain why it's such a big factor in the power market.

Power Markets

For the Nordic countries, the power market was liberalised through the consumer market in the 1990's. This led to competition between the power suppliers. NordPool, owned by national grid companies from Nordic and Baltic countries, was established in 1996 as the first market to trade power between countries. More power markets followed in the time after this, including EEX in 2002, which is the market in focus in this thesis. Although both exchanges trade in many of the same countries, NordPool is the biggest in the Nordic and Baltic area, while EEX is more focused on central-Europe.[Reg; Nor19; EEX]

Today, power is traded between countries everyday and this has led to more stability in the price. This is because, by basic supply and demand, power will be bought and transported from where it's cheap, to where it's more expensive at the time. This transferring between countries has led to a debate on Norway's power use, and its impact on climate. Of the total power production in Norway in 2017, 98% was renewable. While after having transferred power with countries with different natural prerequisites, where the power production has a smaller percentage that is renewable, only 32% of the power used was guaranteed to come from such renewable sources. 46% was fossil power, while the rest was nuclear.[ene18]

Energy companies and private investors trade in these markets, and the prices we pay as private consumers is calculated by the energy companies from these prices.

Futures, Forwards and Options

We have collected data for Phelix-DE futures contracts over a quarterly period from EEX.

A *futures contract* is an agreement to buy or sell a commodity, power in our case, for an agreed upon price, at a given time in the future. As an example On EEX, the latest transaction on a futures contract for the 3rd quarter of 2019 is 11,040 MWh bought for a price of 45.50 Euro/MWh.

We say that the person who has bought the contract holds a *long* position and the one selling is the *short* position holder. Of course, if the price goes over 45.50 before the date of delivery, the person holding the long position has made money, assuming he or she sells it at this point for the new, higher price. This way it can be viewed as a bet on the future price of the commodity.

The futures contract is very similar to the *forward contract*, and so far, the description is exactly the same. The difference, however, between these two contracts is that the futures contracts are marked-to-market every day. That is, the two parties has to settle the difference in price each day up until the agreed delivery time, while the forward contract is settled only at the delivery time. This could make the calculations slightly different for the two, by considering interest rate earned by having the money, but we will look away from this factor in our models. The same amount of money will be transferred between the two in any case.

Historically, these contracts was important to secure production. A person that produced a commodity could sell it for a specified price and time in the future, to secure that the whole production process doesn't end up being worth nothing.

Nowadays the contracts are mostly used for trading, but also on EEX the physical delivery can be arranged between the trading participants. If the physical deliverance of a futures contract is fulfilled it is either delivered all at the same time, as it is for oil futures, or, as in the power case, delivered throughout the period.

In the example above, 11,040 MWh would be delivered over the 3rd quarter of 2019 if the contract were to be fulfilled. The period in which power is delivered can range from a single day to a year.

The fact that it's not all delivered at the same time will make our calculations a bit more tricky, and we will see later explain what adjustments we need to make because of this.

When doing the calculations later, we will use the settlement price of each contract. This is a price calculated from the average exchange prices and average of the best ask and the best bid within a given time and order

quantity by EEX. The formula depends on whether there were orders and trades, and can be found in [EEX18]

EEX also trades *European put* and *call options* on these futures contracts. A call option is the possibility to buy a futures contract for a strike price, K , at an agreed upon time of maturity, T . The fair price of these options are, on EEX, calculated by the Black-76 formula. We will come back to this formula in chapter 6. On EXX the Phelix-de futures, these options are possible to trade on monthly, quarterly and yearly contracts.

On EEX, in addition to these prices, we might find the total volume traded and data for the exchanges made each day.

Although the opening for transaction over borders made the power price more stable than earlier, it still varies more than the prices of many other commodities. There are multiple reasons for this, but as we now will see, the seasonality of the power price is probably the strongest factor.

Variation in the Power Market

The seasonality is a bigger part of the power market than it is for most other markets. This is easy to understand as, for someone living in a warm area, the power usage will increase in summer as more power is needed to cool down their house. Colder areas, like those we've got data for, will do the opposite and spend more power during winter for heating.

Seasonality does not only affect the usage, but also the level of production. For example the production of hydro-power will be higher in rainy seasons.

It is very likely that the climatic issues leads to more extremities in the weather. This means that in the future we could get an even stronger seasonality and more instability in resources needed to produce power.

Other reasons for variation in the power price might be changes in governments, which leads to changes of what tax-level is put on different production techniques. Sometimes even which ones that are legal. Less dramatic than this is that the conditions for transferring of electricity between countries might vary. These are of high importance as electricity is hard to store in an economically efficient way.

In chapter 4 we will look more into modelling a function to describe the seasonality. [Reg]

3 Kriging

One basic assumption of geostatistics is that soil close to a measured point, should be more similar to this measurement than soil further away. In this field the kriging technique occurred as a way of optimising a curve after measured points by interpolating them, and giving an error function to describe how accurate this estimate is.

In this thesis we will to use this technique on the field of Mathematical Finance, with the thought that prices of contracts delivered at dates close in time will be more similar than those delivered at a further time apart. We will use kriging to find a curve that capture the development in the futures prices in the market presented in the previous chapter, but the error is also an important reason for us choosing this curve fitting technique.

First we will present kriging and what a Gaussian Random Field is. Then we will prove that the simple kriging form is an optimal predictor in the mean squared sense. We will also present the form of its error-function.

In our proof we will point out how one assumption made in the proof from [CW11] isn't obvious, but by looking into detail, we find that the results are the same.

After this we will present different types of kriging, as simple kriging isn't the only technique. Before finishing the chapter by presenting the semivariogram. This is an alternative way to look at covariance, and an important part of kriging. This will be presented together with the nugget effect, an explanation for small variance in the data. We will look into whether this is transferable to mathematical finance or not

3.1 Gaussian Random Field and Kriging

Kriging comes from geostatistics, or more specifically gold mining, as a method for spatial prediction. It is named after the South African mining engineer Danie G. Krige, who's work Matheron later developed the theory from. It is an interpolation technique that is found by minimising the mean squared error of a general predictor class and a set of data. Simple kriging looks at predictors on the form $\mathbf{l}'\mathcal{F} + k$, where \mathcal{F} is the dataset, $\mathbf{l}' \in \mathbb{R}^n$ and $k \in \mathbb{R}$, that is all heterogeneously linear predictors.

In kriging, there are few assumptions: we only need a mean a covariance function. Due to this it has been used in multiple other fields as well. In mathematical finance, however, it hasn't been as popular.[Krig]

When we're kriging we are also going to assume that the futures contracts price process is a *Gaussian Random Field*. To understand what this means, we first need to define a *Stochastic Process*:

Definition 1. A *Stochastic Process* is a collection of random variables $\{f(t) : t \in [0, T]\}$.

A *Random Field* is a generalisation of stochastic processes where f and t can be of dimensions higher than 1. We have a (m, n) -*random field* if f is m -dimensional random and t is n -dimensional.

With $f_t = f(t_1, \dots, t_n) = (f_t^1, \dots, f_t^m)$, a *Gaussian Random Field*, or GRF, is a random field where, for any $t \in \mathbb{R}^n$, each f_t^i has the Gaussian distribution.

The form of the GRF is entirely decided by its mean and covariance function. We will look more into the choosing of these in chapter 5. The existence of these two moments, and the fact that it is stable under linear combinations makes the GRF a good choice to describe the process.

Now, every GRF, $f(t)$, can be written as

$$f(t) = m(t) + Y(t)$$

Where $m(t) = \mathbb{E}[f(t)]$ is a mean function, and $Y(t)$ is the zero-mean GRF depending on a covariance function $\text{Cov}(f(x), f(y))$.

With a given dataset of futures prices $\mathcal{F} = \{\hat{F}_1, \dots, \hat{F}_n\}$, we can perform kriging on the GRF and find the predicted curve as the conditional expectation

$$f_{\text{pred}}(t) = \mathbb{E}[f(t) | \hat{F}_1, \dots, \hat{F}_n]$$

We will find the exact form of this in the following section.

We will also find the form of the error $\sigma_{\text{sk}}(t) = \text{std}(f(t) | \hat{F}_1, \dots, \hat{F}_n)$, a conditional standard deviation. This error is one of the main reasons we want to use the kriging technique, and with this we can quantify the uncertainty of the futures price. As we don't have any exact data-points, but only a period-average to estimate our points from, an estimate of the error will be important.

3.2 Proof of the Simple Kriging form

We have seen that the optimal form of our GRF is really a conditional expected value of $f(t)$. As already mentioned *Simple kriging* considers all on the form $\mathbf{l}'\mathcal{F} + k$. We want to find the predictor that is optimal in the mean squared sense. That is, our predicted curve, $f_{\text{pred}}(t)$, is the function $f^*(t)$ that minimises the mean squared error

$$\mathbb{E}[(f^*(t) - (\mathbf{l}'\mathcal{F} + k))^2]$$

Since we are considering contracts delivered over a period and not spot prices, our dataset, \mathcal{F} , isn't values for the actual field $f(t_i)$, but rather for the period average $F(t_i) = \frac{1}{l} \int_{t_i}^{t_i+l} f(z)dz$. Here l is the length of the period. This means that our covariance-function for contracts with delivery period x and y will be

$$\text{Cov}(F(x), F(y)) = \frac{1}{l^2} \int_y^{y+l} \int_x^{x+l} \text{Cov}(f(u), f(v))dudv$$

We will denote the matrix with $\text{Cov}(F(t_i), F(t_j))$ as its entries by C . In the last section we will see that this matrix changes slightly when considering measurement error, but in the proof we will look away from this. Meaning that for now we have $\text{Cov}(\hat{F}(x), \hat{F}(y)) = \text{Cov}(F(x), F(y))$ and $\text{Cov}(f(x), F(y)) = \text{Cov}(f(x), \hat{F}(y))$ for all x, y 's. We let \mathbf{c}_t be the covariance-vector given by

$$\mathbf{c}_t = \begin{pmatrix} \text{Cov}(f(t), F(t_1)) \\ \text{Cov}(f(t), F(t_2)) \\ \vdots \\ \text{Cov}(f(t), F(t_n)) \end{pmatrix} = \begin{pmatrix} \frac{1}{l} \int_{t_1}^{t_1+l} \text{Cov}(f(t), f(z))dz \\ \frac{1}{l} \int_{t_2}^{t_2+l} \text{Cov}(f(t), f(z))dz \\ \vdots \\ \frac{1}{l} \int_{t_n}^{t_n+l} \text{Cov}(f(t), f(z)), dz \end{pmatrix}$$

and σ_f be the variance of the field. With \mathbf{m} as the vector with $\frac{1}{l} \int_{t_i}^{t_i+l} m(z)dz$, for all i 's, as it's entries, we denote the difference $\mathbf{m} - \mathcal{F}$ by $\hat{\mathbf{y}}$ and get the proposition:

Proposition 1. The "simple kriging" - optimal spatial predictor of a Gaussian random field, in the mean squared sense is

$$f_{\text{pred}}(t) = m(t) + \mathbf{c}'_t C^{-1} \hat{\mathbf{y}}$$

Proof. As mentioned we will be looking at minimising the expression $\mathbb{E}[(f(t) - (\mathbf{l}'\mathcal{F} + k))^2]$.

Of course we have that:

$$\mathbb{E}[(f(t) - (\mathbf{l}'\mathcal{F} + k))^2] = \text{var}(f(t) - (\mathbf{l}'\mathcal{F} + k)) + [\mathbb{E}(f(t) - (\mathbf{l}'\mathcal{F} + k))]^2,$$

and writing out the first term we get:

$$\begin{aligned} \text{var}(f(t) - (\mathbf{l}'\mathcal{F} + k)) &= \text{var}(f(t) - l_1\hat{F}(t_1) - \dots - l_n\hat{F}(t_n) - k) \\ &= \sigma_f^2 - 2 \sum_{i=1}^n l_i \text{Cov}(f(t), \hat{F}(t_i)) + \sum_{i,j} l_i l_j \text{Cov}(\hat{F}(t_i), \hat{F}(t_j)) \\ &= \sigma_f^2 - 2 \sum_{i=1}^n l_i \text{Cov}(f(t), F(t_i)) + \sum_{i,j} l_i l_j \text{Cov}(F(t_i), F(t_j)) \\ &= \sigma_f^2 - 2\mathbf{l}'\mathbf{c}_t + \mathbf{l}'\mathbf{C}\mathbf{l} \end{aligned}$$

The second term will be:

$$\begin{aligned} [\mathbb{E}(f(t) - (\mathbf{l}'\mathcal{F} + k))]^2 &= (m(t) - \mathbf{l}'\mathbf{m} + k)^2 \\ &= m(t) - 2m(t)\mathbf{l}'\mathbf{m} - 2km(t) + \mathbf{l}^2\mathbf{m}^2 + 2k\mathbf{l}'\mathbf{m} + k^2. \end{aligned}$$

And so our expression for the simple kriging-error is now

$$\sigma_{\text{sk}}(t; \mathbf{l}, k) = \sigma_f^2 - 2\mathbf{l}'\mathbf{c}_t + \mathbf{l}'\mathbf{C}\mathbf{l} + m(t) - 2m(t)\mathbf{l}'\mathbf{m} - 2km(t) + \mathbf{l}^2\mathbf{m}^2 + 2k\mathbf{l}'\mathbf{m} + k^2$$

To minimise this error we need to find the optimal \mathbf{l}^* and k^* . We do this by differentiating the error with respect to each of them.

At this point [CW11] chooses a simplified optimisation technique by only looking at the first term when differentiating with respect to \mathbf{l} . This isn't obviously correct at first, so to be clear we have done this more thorough to show that it does actually lead to the optimal \mathbf{l} .

So, we look at \mathbf{l} first:

$$\frac{\partial}{\partial \mathbf{l}} \sigma_{\text{sk}}(t; \mathbf{l}, k) = -2\mathbf{c}_t + 2\mathbf{l}\mathbf{C} - 2m(t)\mathbf{m} + 2\mathbf{l}\mathbf{m}^2 + 2k\mathbf{m}$$

Putting this equal to 0 we get the optimal \mathbf{l} by

$$\mathbf{l}^* = (\mathbf{c}_t + m(t)\mathbf{m} - k\mathbf{m})(\mathbf{m}^2 + \mathbf{C})^{-1}.$$

Doing the same for k we get that:

$$k^* = m(t) - \mathbf{l}'\mathbf{m},$$

and putting k^* into \mathbf{l}^* we get, similarly to [Cressie] that

$$\begin{aligned}\mathbf{l}^* &= (\mathbf{c}_t + m(t)\mathbf{m} - (m(t) - \mathbf{l}^*\mathbf{m})\mathbf{m})(\mathbf{m}^2 + C)^{-1} \\ \mathbf{l}^*(\mathbf{m}^2 + C) - \mathbf{l}^*\mathbf{m}^2 &= \mathbf{c}_t + m(t)\mathbf{m} - m(t)\mathbf{m} \\ \mathbf{l}^* &= \mathbf{c}_t C^{-1}\end{aligned}$$

Further, by putting this new expression back into k^* :

$$k^* = m(t) - \mathbf{c}_t C^{-1} \mathbf{m}$$

So the optimal mean squared predictor is given by:

$$\begin{aligned}f_{\text{pred}}(t) &= \mathbf{l}^{*\prime} \mathcal{F} + k^* \\ &= \mathbf{c}_t C^{-1} \mathcal{F} + m(t) - \mathbf{c}_t C^{-1} \mathbf{m} \\ &= m(t) + \mathbf{c}_t C^{-1} (\mathcal{F} - \mathbf{m}) \\ &= m(t) + \mathbf{c}_t C^{-1} \hat{\mathbf{y}}\end{aligned}$$

□

The mean squared error we started out with solves to

$$\begin{aligned}\sigma_{\text{sk}}^2(t; \mathbf{l}, k) &= \mathbb{E}[(f(t) - (\mathbf{l}^{*\prime} \mathcal{F} + k^*))^2] \\ &= \sigma_f^2 - \mathbf{c}_t' C^{-1} \mathbf{c}_t,\end{aligned}$$

which means that our simple kriging error-function is given by

$$\sigma_{\text{sk}}(t) = \sqrt{\sigma_f^2 - \mathbf{c}_t' C^{-1} \mathbf{c}_t}.$$

Again, if our datapoints weren't periods, this error would be 0 for these time-values.

Other forms of Kriging

Now we have found the form of the predictor we will use in this thesis, but other forms of kriging than that of simple kriging exists. In this section, we will present what is known as *universal kriging*, although without proof. We will also present a special case of this, known as *ordinary kriging*. Similar to the case of simple kriging, both of these forms also have error-functions, but these won't be presented here.

The GRF, $f(t)$, can be generalised to the form $f(t) = \mathbf{x}(t)\boldsymbol{\beta} + Y(t)$. The mean function is now a product of a vector of covariates, $\mathbf{x}(t)$, and a regression parameter $\boldsymbol{\beta}$. This would give the new simple-kriging form

$$f_{\text{pred}}(t) = \mathbf{x}(t)' \boldsymbol{\beta} + \mathbf{c}_t C^{-1} (\mathcal{F} - \mathbf{X} \boldsymbol{\beta})$$

Usually, however, this $\boldsymbol{\beta}$ is unknown. From the proof in the previous section, it's apparent that $\boldsymbol{\beta}$ will only be in the k^* -term. Therefore, when finding the ordinary kriging predictor, only the class of homogeneously linear predictors will be considered. That is, predictors on the form $\boldsymbol{\lambda}\mathcal{F}$. in stead of $\boldsymbol{l}\mathcal{F} + k$ as we did to find the simple kriging predictor.

If we now demand the predictor to be unbiased, that is $\mathbb{E}(f_{\text{pred}}(t)) = \mathbb{E}(f(t)) = \boldsymbol{x}(t)'\boldsymbol{\beta}$, we can find the universal kriging predictor by

$$f_{\text{uni}}(t) = \boldsymbol{x}(t)'\hat{\boldsymbol{\beta}}_{\text{gls}} + \mathbf{c}_t C^{-1}(\mathcal{F} - \mathbf{X}\hat{\boldsymbol{\beta}}_{\text{gls}}).$$

Here $\hat{\boldsymbol{\beta}}_{\text{gls}}$ is the generalised least squares estimation of $\boldsymbol{\beta}$, which is given by $(\mathbf{X}'C^{-1}\mathbf{X})^{-1}\mathbf{X}'C^{-1}\mathcal{F}$.

Now, if we let \boldsymbol{x} be 1-dimensional and equal to 1, while we also put $\boldsymbol{\beta} = m(t)$, we get, with $\mathbf{1} = (1, \dots, 1)$, the form of the ordinary kriging predictor

$$f_{\text{ok}}(t) = m_{\text{gls}}(t) + \mathbf{c}_t C^{-1}(\mathcal{F} - m_{\text{gls}}(t)\mathbf{1}),$$

where $m_{\text{gls}}(t) = (\mathbf{1}C^{-1}\mathcal{F})/(\mathbf{1}C^{-1}\mathbf{1})$ is the generalised least squares estimation of $m(t)$.

The universal kriging predictor, unlike the simple kriging predictor, can be rewritten to be with respect to the semivariogram in stead of covariance function. [CW11]

3.3 Semivariogram and the Nugget Effect

One way to evaluate the covariance-structure of a random field is to look at its semivariogram. Instead of showing the correlation, it shows how much the field varies by distance. In our case it shows how much the futures prices vary by how many periods apart they are being delivered. Intuitively, one would assume that the difference in price would vary more for a bigger time gap. Since two contracts delivered at the same time of course will have the same price, it also makes sense that the variation and so the semivariogram goes towards 0, when $t \rightarrow 0$.

The geostatisticians who used the semivariogram for spatial statistics found that the semivariogram didn't behave like this. In stead it went towards some positive value. They named this discontinuity at the origin *the Nugget Effect*.

In this section we will first look at the semivariogram without this nugget effect. Then present it more thoroughly and show the changes we have to make to include it in our mathematical model. Finishing this section and

chapter I will discuss the relevance of the nugget effect in our context.

We say that a covariance function is *stationary* if, for a process $f(t)$, $\text{Cov}(f(x), f(y))$ is only dependent on the distance $|x - y| = h$. We denote the stationary covariance function for two contracts h periods apart by $C_f(h) = \text{Cov}(f(t + h), f(t))$, for all t . Now we will use the definition from [CW11] for the corresponding stationary semivariogram:

$$\gamma_f(h) = C_f(0) - C_f(h).$$

Example 3.3.1. Consider the exponential covariance, $\text{Cov}(x, y) = \sigma^2 e^{-\rho|x-y|}$, then we have that the corresponding semivariogram is

$$\gamma(h) = C_f(0) - C_f(h) = \sigma^2 - \sigma^2 e^{-\rho|h|} = \sigma^2(1 - e^{-\rho|h|}).$$

Seeing this, it becomes obvious how the semivariogram is just an alternative way of looking at the covariance. The two curves will have an opposite structure.

The optimal predictor we found in the previous section is defined as generally as possible and doesn't actually need the covariance function to be stationary. However, we will use one that is. The reason for this is that, later, we will fit the semivariogram based on this covariance function to the empirical semivariogram to get estimates for the parameters ρ and σ , and this empirical semivariogram is stationary, and with $\text{ave}\{A\}$ denoting the average of a set A , we define it as

$$\gamma_{\mathcal{F}}(h) = \frac{1}{2} \text{ave}\{(\hat{F}_i - \hat{F}_j)^2 : |i - j| = h, i, j = 1, \dots, n\}. \quad (1)$$

As $\hat{F}_i - \hat{F}_j = 0$ when $i = j$, it's clear that the dataset we get from this semivariogram will have a point in the origin. We will now look into why the curve doesn't always behave like this.

The Nugget Effect

As mentioned, the variogram was used to describe variation in soil by distance. The name, nugget effect, comes from the sudden appearance of gold nuggets that would change this data in an unnatural way. This is an example of what would be considered a measurement error, which is one out of two explanations for the nugget effect. The other being the existence of a

micro scale variance in the process. We will denote these two errors by σ_ϵ and σ_0 , respectively, and their sum, the nugget effect, by c_0 . We define it as

$$c_0 = \lim_{h \rightarrow 0} C_{\mathcal{F}}(0) - C_{\mathcal{F}}(h),$$

where $C_{\mathcal{F}}(h) = \text{Cov}(\hat{F}(x+h), \hat{F}(x))$ is the stationary covariance of measured points.

This changes our functions for covariance and semivariogram to

$$\begin{aligned} \text{Cov}(h) &= c_0 \chi_{h=0} + \sigma^2 e^{-\rho|h|} \\ \gamma(h) &= c_0 \chi_{h=0} + \sigma^2 (1 - e^{-\rho|h|}) \end{aligned}$$

Where $\chi_{h=0}$ is the indicator function for zero distance. This way the semivariogram goes towards c_0 instead of 0.

Theoretically we want the sampling to be perfect, meaning no measurement error, but if we were to take it into account, we would have to change the covariance-matrices we saw in the proof.

The new elements of the matrix would be

$$\text{Cov}(\hat{F}(x), \hat{F}(y)) = \begin{cases} \text{Cov}(F(x), F(y)) + \sigma_\epsilon^2, & \text{when } x = y \\ \text{Cov}(F(x), F(y)), & \text{when } x \neq y \end{cases}$$

That is, while we have to change the elements on the diagonal of the covariance matrix, the other elements remains unchanged. Following this, $\text{Cov}(f(t), F(t_i)) = \text{Cov}(f(t), \hat{F}(t_i))$ for all i 's, and so we keep \mathbf{c}_t unchanged.

So, which semivariogram, and which covariance-matrix should we use in our case?

Since our measuring is only reading the settlement prices calculated by EEX, we can look away from σ_ϵ , so C , the covariance-matrix, stays the same.

A variance in the process, however, is possible as it is a market in which people speculate, and so there might be sudden spikes. This would probably be more apparent if we were considering the price process of one future contract. We're considering the price process for the contract delivered at a time t , and we have observed on EEX that the prices on the different futures contracts change together. This makes their relative variance more predictable. We will try to estimate the semivariogram both with and without the nugget effect, but by this we'll see that it in fact turns out to be 0. [CW11]

4 Estimating the Kriged Curve of Futures prices

In this chapter we will use the kriging technique found in the previous chapter on actual data for futures prices read off from EEX.

First we will introduce the *Nelson-Siegel curve*, our choice of mean function, but before we can make an estimate of it, we need to discuss the seasonality. We will consider two different options to capture it.

After having estimated the Nelson-Siegel curve, we need to return to the semivariogram to find estimates for the parameters of the covariance-function. Then, at the end of the chapter, we will see the kriged curve with errors to show how it captures the data in a much more flexible way than the mean function did.

4.1 Estimating the Mean Function

The Nelson-Siegel Curve

We are going to use the Nelson-Siegel curve, introduced in [NS12], as our mean function. It has the form

$$f_{\text{NS}}(t) = \alpha_0 + \alpha_1 e^{-\beta t} + \alpha_2 \beta t e^{-\beta t}. \quad (2)$$

This curve has become popular in a range of fields, and is, together with the extension by Svensson in 1994, still used by banks as a model for term structure of interest rates. [APK12]

The Nelson-Siegel curve was first used to model yield curves to calculate *forward rates*. That is, an interest rate used for a future financial transaction. With these forward rates, one also considers the *yield to maturity*, which is the total return. This is given by the average of the forward rates. For a forward rate with maturity t , $r(t)$, the yield to maturity is given by

$$R(t) = \frac{1}{t} \int_0^t r(z) dz.$$

Now, we remember that our price of the futures contract with maturity at time t were given by

$$F(t) = \frac{1}{t} \int_0^t f(z) dz,$$

where we want to estimate $f(t)$ from data for $F(t)$. Considered this way, $F(t)$ is the average of the futures prices in the period, and so it's easy to

see the similarities between $r(t)$ and $f(t)$. This, and the fact that it is a relatively good estimator for few observed values[APK12], is the reason we choose this curve as our mean.

Looking at 2, we see how it's similar to the solution of a differential equation with one root. The reason for this is simply that this is the class they looked into to find their curve. The solutions of differential or difference equations was already used to describe yield curves, and if a spot rate were given by such a differential equation, the forward rate, $r(t)$, would be the solution of this equation.

So, looking at these solutions they found that the solution to the second-order differential equation with real and unreal roots were over-parametrised, and that the solution to the equation with one root had the same range, while being much easier to estimate[NS12]. Due to concave/convexity-problems, however, the estimation can still be troublesome.

By considering

$$\begin{aligned}\lim_{t \rightarrow \infty} f_{\text{NS}}(t) &= \alpha_0, \text{ and} \\ \lim_{t \rightarrow 0} f_{\text{NS}}(t) &= \alpha_0 + \alpha_1,\end{aligned}$$

we see that α_0 is the long-term component, while α_1 affects the short-term behaviour of the curve. We will look more into each parameter's impact on the curve in the next chapter.

We can't tell much of the other two parameters from this, but we will see later that both α_2 and β affects the middle range of the curve. We will look deeper into this in chapter 5

We know Nelson-Siegel won't be completely accurate. However, we are only using it as a mean function, and we're confident that it will be good enough for this purpose.

Function for Seasonality

We've only got data for 9 quarters. This is not ideal when finding a function for seasonality, but using an average function we can get a rough estimate. As we're only considering two years, and don't see any clear trend we will look away from this.

The structure of the power prices is cyclical. This calls for a sine-function, which would, of course, give a much smoother function than the average-function proposed above. However, as we are considering quarterly periods, it's not clear that a sine-curve would be more helpful. Had we in stead been looking at weekly contracts, or monthly even, a sine-curve could perhaps be

the better option, as, in that case, we would see the development throughout the year clearer. For our quarterly data it's hard to notice any other dynamic than the prices being higher in the two winter quarters, 1 and 4, and low in summer, 2 and 3. Therefore we will use the simple average-function.

252 or 365 days in a year

Before doing the estimations we need to consider that in financial markets there are 252 days a year, and take this into account when creating our models.

Our first estimation, however, is of the price of the contract delivered at time t , and since power is delivered every day of the year, we will still use 365 days. As our period-length of a quarter we will use $\frac{365}{4} \approx 90$. Following this we will also have 90 points for each period when estimating the parameters of the Nelson-Siegel curve.

We actually won't have to consider the issue of 252 days in a year before we're looking at options-pricing in the last chapter.

Estimate of the Nelson-Siegel Curve

Since we are looking at future contracts delivered over a period, we won't have daily points, but rather data for the whole quarter. Our datapoints are the prices for Phelix-DE Baseload Quarter Futures, collected on March 1, 2019. We will use the same value for each day within a period, and this way fit eq. (2) to $9 \cdot 90$ points. We use these to calculate the vector of average seasonal values from \mathcal{F} and use this to create our function for seasonality. We get the vector $\{51.62, 43.00, 46.08, 52.12\}$. As we expected, these prices from Phelix-DE shows a clear seasonality. Now we subtract

$$f_s(t) = \begin{cases} 51.62 & t \in \text{1st quarter} \\ 43.00 & t \in \text{2nd quarter} \\ 46.08 & t \in \text{3rd quarter} \\ 52.12 & t \in \text{4th quarter} \end{cases}$$

from the data-points, and fit the Nelson-Siegel curve to the new, adjusted points by using R's 'nls'-function with a fixed β , as suggested in [HR12].

This method demands initial values, and since we have that

$$\begin{aligned} \lim_{t \rightarrow \infty} f_{\text{NS}}(t) &= \alpha_0 \\ \lim_{t \rightarrow 0} f_{\text{NS}}(t) &= \alpha_0 + \alpha_1, \end{aligned}$$

We will use $\alpha_0^{\text{init}} = \hat{F}_n$ and $\alpha_1^{\text{init}} = \hat{F}_1 - \hat{F}_n$.

We can't determine α_2^{init} from this, but as it turns out, with α_0^{init} and α_1^{init} set, we get the same result for any $\alpha_2^{\text{init}} \in [-100, 100]$, so it's not important to be accurate with this parameter. We also found that the initial values for α_0 and α_1 can vary a lot without giving a different estimation.

To find β , we set the estimated values for the α_i 's and try different values from [NS12] as β^{init} , and see quickly how R's sum of squared errors goes down, as β is chosen to be smaller.

We find the optimal values to be $\alpha_0 = -2.375, \alpha_1 = 1.761, \alpha_2 = 7.537, \beta = 0.00416$. The Nelson-Siegel curve with these parameters is plotted together with the data from EEX in Figure 1. It's important to note that on the t starts on 30. The reason for this is that we collected the data a month before the start of the second period. As we have removed seasonality the y -axis isn't showing the futures prices, but rather the price-dynamics with time.

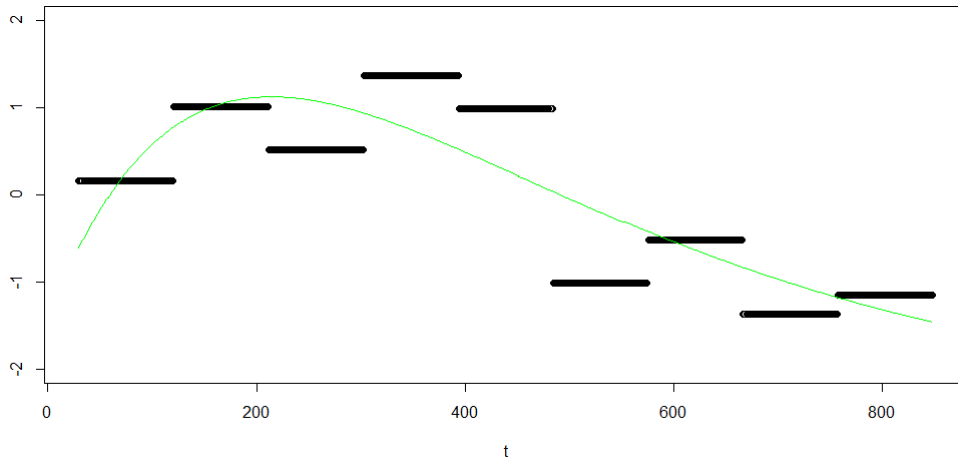


Figure 1: The Nelson-Siegel curve

We see that the Nelson-Siegel curve captures the form fairly well, but that it does not capture "extremities". It's reasonable to use as a mean function for the Kriging that we will do in the following section.

4.2 Kriging of Estimated Curve

With our mean function defined, what we need to perform kriging is to define a covariance function. Initially we will use the exponential covariance function with nugget effect. That is

$$\text{Cov}(f(x), f(y)) = c_0\chi_{h=0} + \sigma^2 e^{-\rho|x-y|},$$

We already know that this function is stationary. It is a fairly simple covariance function, and in the next chapter, we will look into whether or not a more complex covariance is preferable.

To use this we need to find σ and ρ , and we will do so by fitting the function for the semivariogram, $\gamma(h) = c_0\chi_{h=0} + \sigma^2(1 - e^{-\rho h})$, to the points from the empirical semivariogram given by eq. 1.

This is a vector of length 9, and once again we use the nls-method in R, to fit the curve to the points. We find the values $\rho = 0.4567$ and $\sigma^2 = 1.2597$. The nugget-effect turns out to be 0, and therefore we can look away from it. This means that in this specific case the variance goes towards zero as the contracts gets closer in time

The semivariogram is plotted in fig. 2, and we see that the curve captures the form of the data quite good, although not perfect. Compared to [FEB], the variation we get is quite big here, and the lack of data might be a reason for this both from estimating semivariogram and when calculating seasonality. Another explanation could be that the power market just has more fluctuations than the markets looked at in that paper.

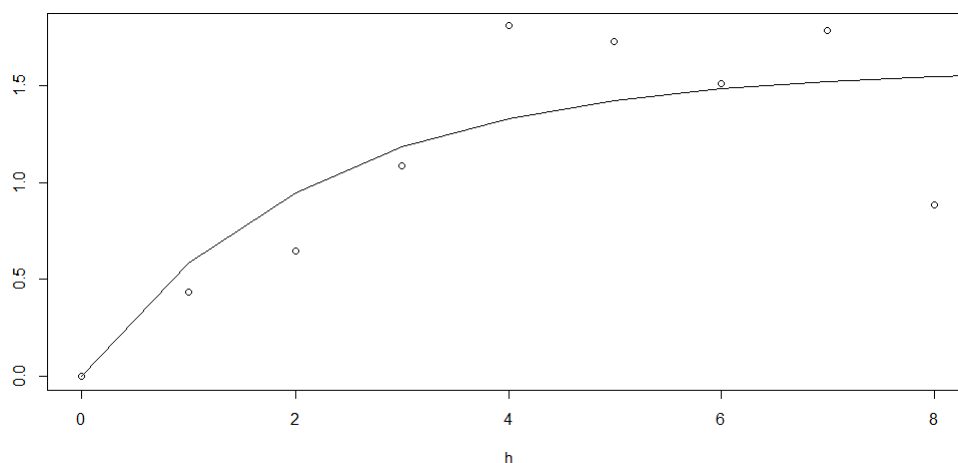


Figure 2: Semivariogram

Now we have all parameters for our Kriged curve, but we're changing the time unit. The semivariogram that we estimated σ and ρ gave us

the variance and correlation of the period, while the predicted curve is given with timesteps measured in days. Therefore our new values will be $\rho' = \frac{\rho}{90} = 0.00507$ and $\sigma'^2 = \frac{\sigma^2}{90} = 0.0176$.

With all parameters set, we plot the kriged curve in fig. 3, and can see how it follows the data much closer. It's a lot more flexible than the Nelson-Siegel curve, and the integral of the predicted curve interpolates the price over the interval. Had we been considering spot prices, the curve would have interpolated each of this points.

This is an estimate for the price of the futures contract delivered t days from now, but of course it's the actual market price that is most important. That's one reason why we want to quantise the error, and why we're using kriging. The curve with added errors is plotted in fig. 4.

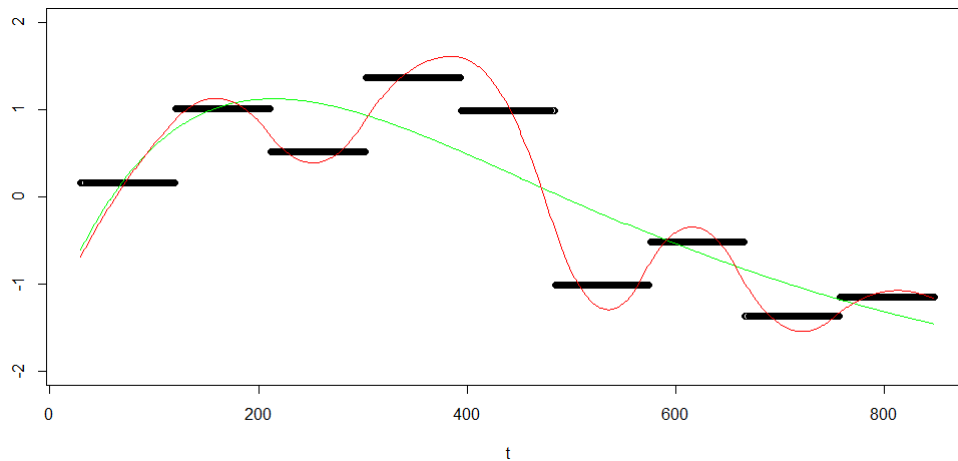


Figure 3: kriged curve

We see that the Nelson-Siegel curve is outside the error-function for a lot of the time.

Similarly to how the predicted curve would interpolate the data points of spot prices, the error would do the same. This is because we would know the value at this points, and of course there wouldn't be any uncertainty. But, as we're considering intervals, we can see that this is not the case.

It's not easy to say much about the development of the error from this plot, so we plot it on its own in fig 5. Here we see that it takes a cyclical shape, being smaller in the middle of the period, and nearly 50% bigger in between. This is hard to spot from the plot of the predicted curve with error.

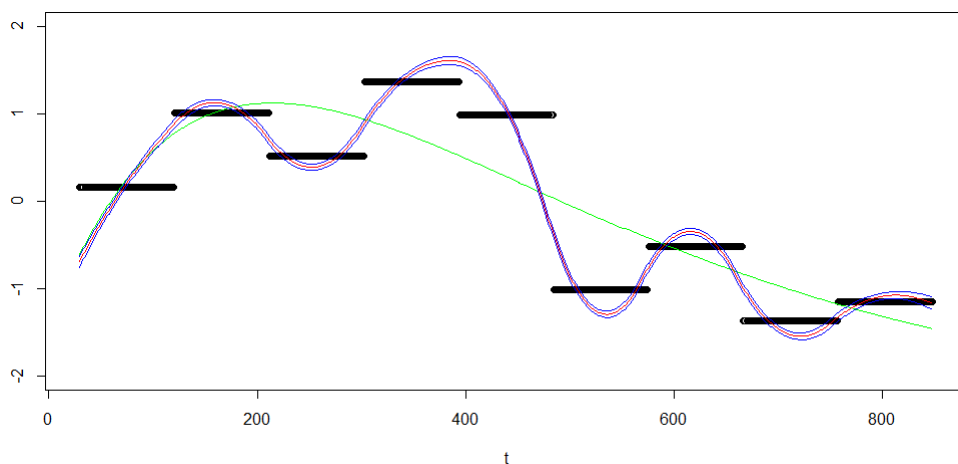


Figure 4: kriged curve with error

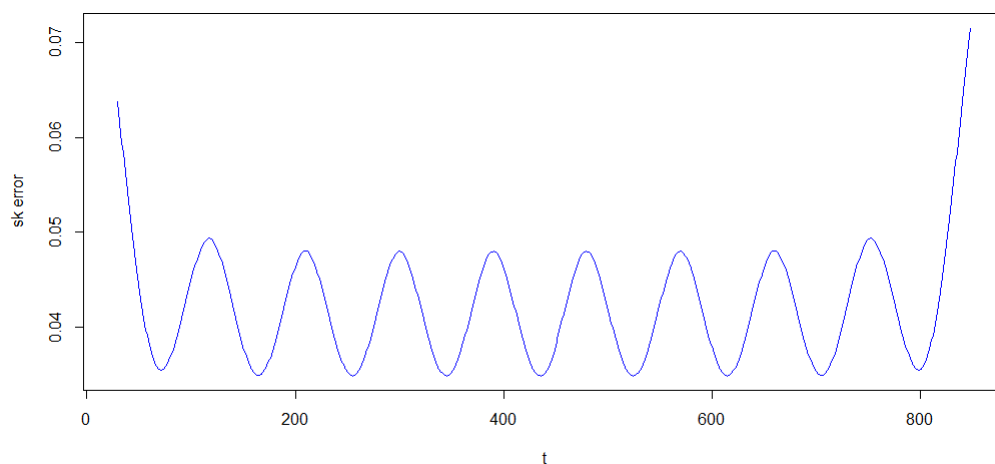


Figure 5: Simple kriging error plotted with time

Splines

Curve fitting has usually been done by either smoothing or interpolation of the data. One popular way to interpolate data is to do this with splines.

We started off with a classic smoothing technique, with a least-squares estimate of the Nelson-Siegel curve, but with kriging we also interpolated the data. Now, we could have used splines, and as Cressie points out in [CW11], kriging is similar to thin plate splines. But doing it this way, we get the error-function we want. And in addition to this, as splines are defined as a piecewise polynomial function, it can have many parameters.

5 Analysis of Parameters and Data

To gain a deeper understanding of the curve we want to look at what role the different parameters play and how important the accuracy is, that is how sensitive the curve is for a wrong valued parameter. This way we can, hopefully, be more certain of our model even with a lack of confidence in one parameter.

After this we will consider how certain we can be of the models we have used to estimate these parameters. We will describe how to generate a random field, and use this to see if our models find the parameters when we know the values of them.

In the last part we will look at what impact of the curve a change in the data will have. We will quickly notice that, because of our average-function for seasonality, a change will impact the curve a lot.

5.1 Parameters

The kriged curve takes in the parameters σ, ρ, β and the α_i 's. In this first section we will consider the parameters of the Nelson-Siegel curve, and explain what each parameter do to this curve. Except for α_0 it has similar effect on the predicted curve, but it's not as easy to spot the differences in this case.

We will in the end of this section show the sensitivity of the α_i 's through plotting their derivative functions.

In the second section we will evaluate our choice of covariance-function with it's parameters, σ and ρ , to see if a more complex function will improve our predicted curve. We will look at both the error functions and the predicted curves for these different covariances, and see that they all, most of the time, look fairly similar.

Parameters of the Nelson-Siegel Curve

We already saw in the previous chapter how α_0 is the long-term component, while α_1 affects the short-term behaviour of the curve, but the limits didn't give us any information of how α_2 and β affects the Nelson-Siegel curve.

In this section we're going to have a closer look at each of the parameters. First, in fig. 6 and 7 we get a confirmation of what we found earlier for α_0 and α_1 . That is, α_0 is the constant part of the curve, and so it doesn't affect the shape of the curve. The whole curve is just shifted upwards or downwards by the element we add to it. The blue line in fig. 6 is with

$\alpha_0 = 0$, while the value is set to -4 in the red line. The green is the same curve we found in the previous chapter in all the figures of this section. That is, in this case $\alpha_0 = -2.375$.

A change in α_1 , on the other hand, changes the form of the curve. we see that the initial values of the curve has changed remarkably, while the curves quickly moves towards similar values for later timesteps. We know that it affects the curve just as much as α_0 does for the first values, but it's obvious that it's effect declines quickly.

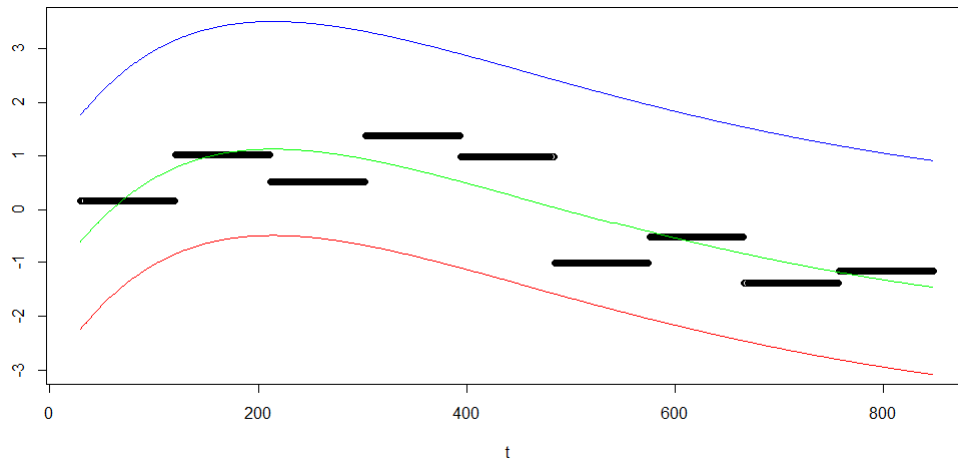


Figure 6: Nelson-Siegel with various values for α_0

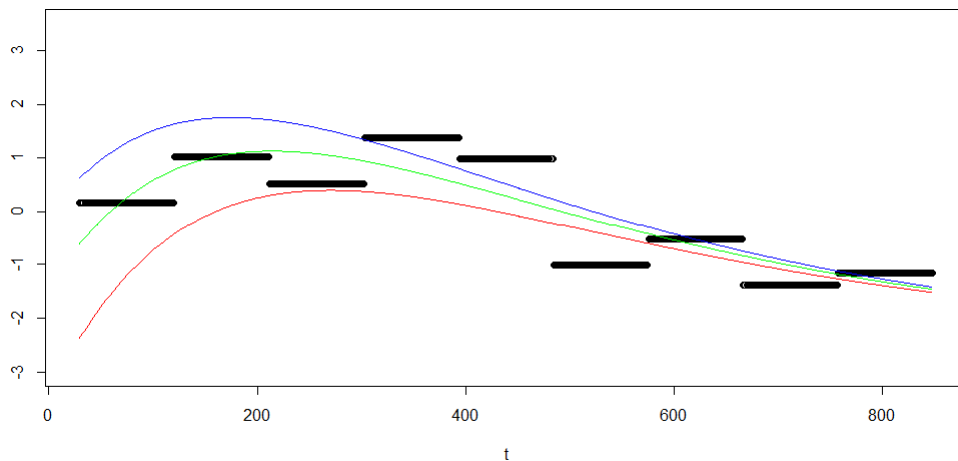


Figure 7: Nelson-Siegel with $\alpha_1 = 3$ (blue line) and $\alpha_1 = 0$ (red)

The curve is plotted for different α_2 's in fig. 8. Here we have $\alpha_2 = 13$ (blue line), 0(red line) and -3.5(black line), and we see that this is affecting the

middle part of the curve. The sign of α_2 tells whether the first hump is concave or convex, while the absolute value describes how big the hump is. We also see that the curve start in the same points for all α_2 , and it closes in towards the endpoint, although not as fast as it does in fig. 7. This implies that this has more importance for later values than α_2 .

This is confirmed in fig. 9, where we have differentiated the Nelson-Siegel with respect to each of the three α_i 's. This plot shows us the factor with which the curve change with a change in the parameter.

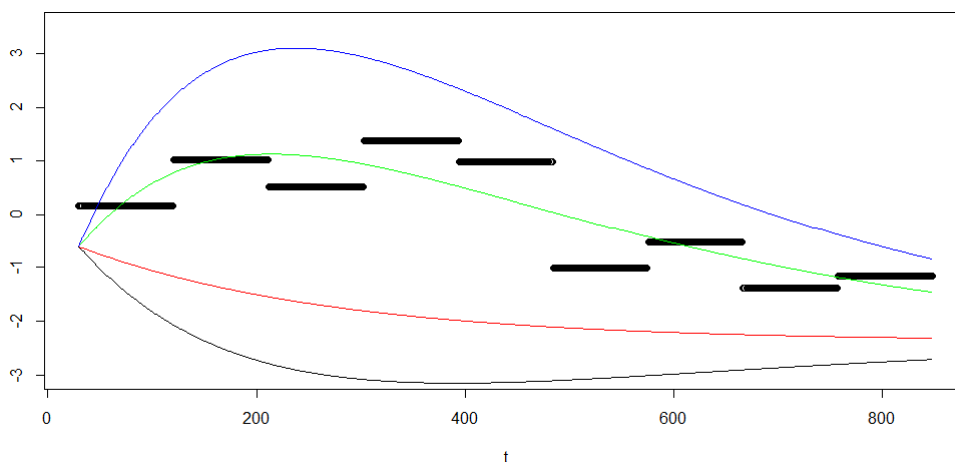


Figure 8: Nelson-Siegel with various values for α_0

We have seen that α_0 changes the curve the whole way through, while α_1 and α_2 has a more local effect. We have that

$$\begin{aligned}\frac{\partial}{\partial \alpha_0} f_{\text{NS}}(t) &= 1 \\ \frac{\partial}{\partial \alpha_1} f_{\text{NS}}(t) &= e^{-\beta t} \\ \frac{\partial}{\partial \alpha_2} f_{\text{NS}}(t) &= \beta x e^{-\beta t}.\end{aligned}$$

Now, with $\beta = 0.00416$, we plot these in figure 9, similarly to what's done in [NS12]. And see that this is correct. What's not as clear from our earlier plot is how much more α_1 affects the curve than α_2 does, but we get a confirmation for that α_2 changes the curve more than α_1 after the first time values. However, α_2 never is a strong factor compared to α_0 , while α_1 has the same importance or the first point, although this quickly decreases. This implies that, in our estimation, it is more important to be accurate when estimating α_0 than the two others.

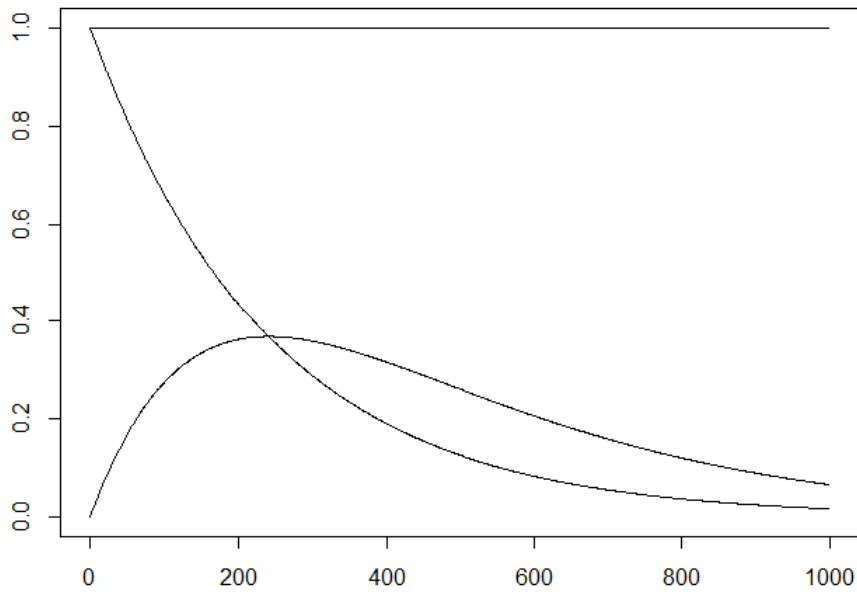


Figure 9

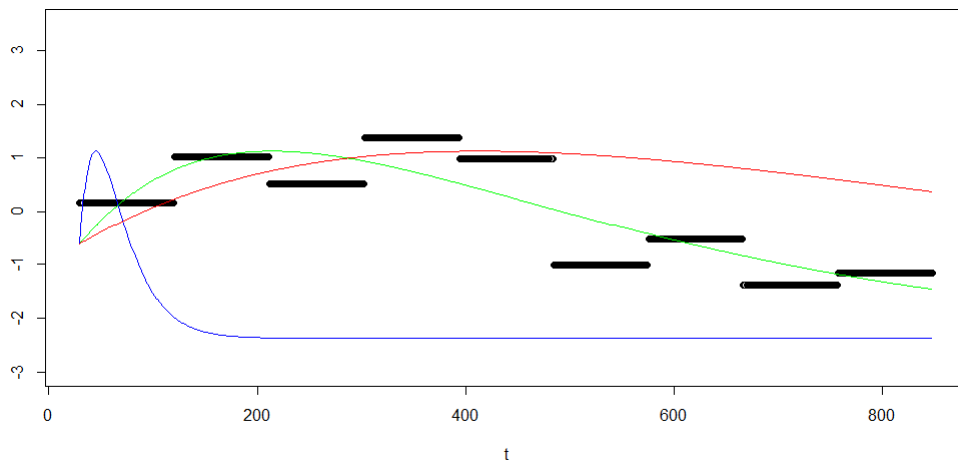


Figure 10: Nelson-Siegel plotted for different β

The last parameter, β , is plotted in fig. 10. The plot shows that β also affects the hump of the curve. That is, it's a part of the medium-term component. This follows from it being a factor in the third term with α_2 . Since β always appear with t , $f_{\text{NS}}(t)$ have the same limits for β as it does for t . Namely, $\alpha_0 + \alpha_1$, when $\beta \rightarrow 0$ and α_0 when $\beta \rightarrow \infty$. Therefore considering a negative β -value is the same as considering a negative time-value, and doesn't make sense.

β describes the placement of the hump and not the size of it. In the plot,

the blue line is the curve with $\beta = 0.05$ and the red line is with $\beta = 0.002$, so a bigger β -value makes an earlier hump. It also describes how fast the curve goes towards α_0 , the limit for when t , or β , goes to ∞ . For the blue line we see that it reaches this value quickly after $t = 200$.

By differentiating $f_{\text{NS}}(t)$ with respect to β , we can locate the maxima or minima of the curve, and find the t -value of the hump:

$$\begin{aligned}\frac{\partial}{\partial \beta} f_{\text{NS}}(t) &= -\alpha_1 t e^{-\beta t} + (\alpha_2 t e^{-\beta t} - \alpha_2 \beta t^2 e^{-\beta t}) \\ &= t e^{-\beta t} (-\alpha_1 + \alpha_2 - \alpha_2 \beta t),\end{aligned}$$

so the hump is at $t = 0$ or $t = \frac{\alpha_1 - \alpha_2}{\alpha_2 \beta}$.

To illustrate the small importance on the predicted curve we include a plot of the predicted curve for various changes of the Nelson-Siegel parameters in fig. 11. Here we see that it mostly affects it at the first two periods and then again at the endpoint, even when changed drastically: for the black line we have used $\beta = 0.02$, the same as for the blue line in fig. 10, and the green line is with a negative α_2 . This implies that the form of our mean function isn't very important for the predicted curve. We will see in the next section how ρ changes this.

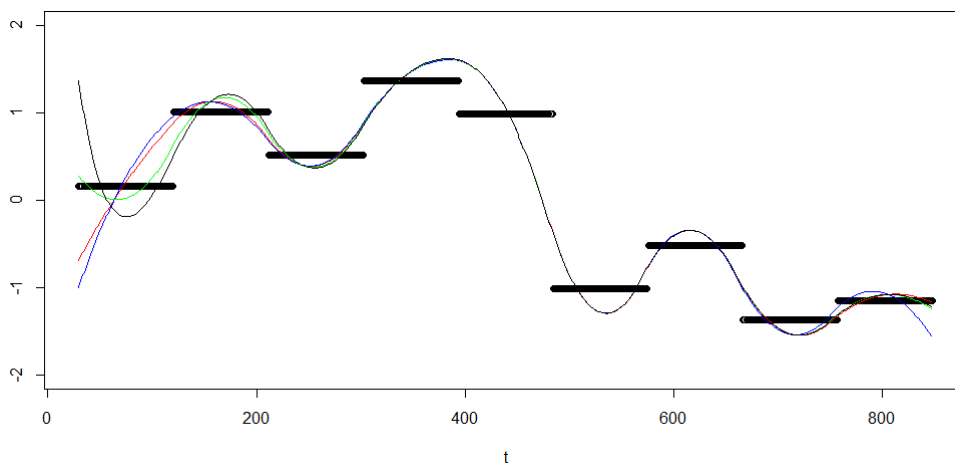


Figure 11: Predicted curve for various β and α_i 's

We've seen that the Nelson-Siegel curve can take many different forms with these four parameters, but of course, as it only creates curves with one hump it won't be able to capture a process like the one we're considering perfectly.

Covariance Function of the Gaussian Random Field

The covariance function has two parameters: σ and ρ . σ describes variance, while ρ describes correlation. Our initial estimation for ρ is 0.00507, which means that the futures price of contracts delivered one day apart is correlated by 0.995. That is, very strongly. By setting a higher value for ρ , hence lowering the correlation between data, we see that the predicted curve isn't as smooth, as it has gotten more dependent on the mean function. This is seen by how, within each period, the predicted curve has the same shape as the mean. This is plotted in fig. 12. We find that by keeping $\rho \leq 0.05$, the kriged curve is fairly smooth.

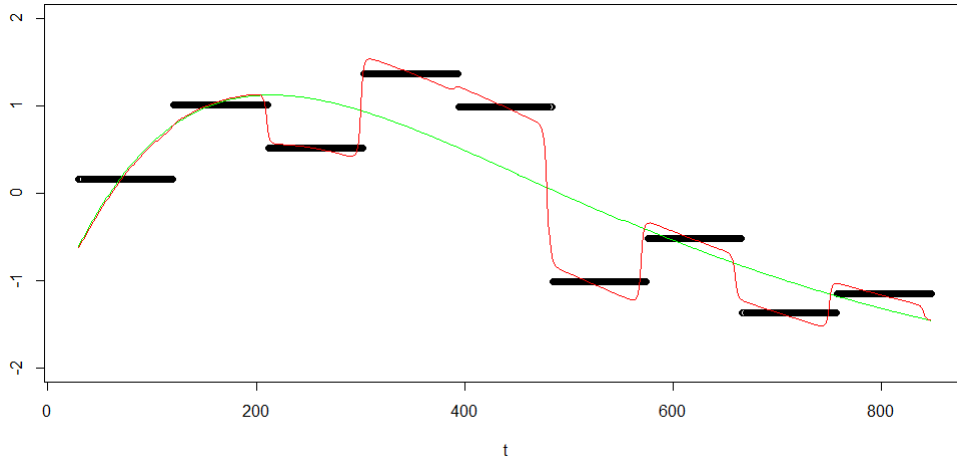


Figure 12: kriged curve with $\rho = 0.5$

A change in σ doesn't change the predicted curve very much, but this is the most important parameter for the error-graph. In fig 13 the predicted curve is plotted in red, with errors with $\sigma_{\text{est}} = 0.1327$, $\sigma = 0.5$, $\sigma = 1.5$ and $\sigma = 5 \cdot \sigma_{\text{est}}$ in green.

From this plot of σ_{sk} we notice that a change in σ doesn't change the form of the curve of σ_{sk} , but only affects the scale. This implies that σ^2 could be factorised from the covariance matrix. Remembering the form of the error being

$$\sigma_{\text{sk}}(t) = \sqrt{\sigma_f^2 - \mathbf{c}_t' C^{-1} \mathbf{c}_t}.$$

We can factorise σ^2 from the two \mathbf{c}_t , while it is also a factor of C , hence $\frac{1}{\sigma^2}$ is a factor of C^{-1} . So leaving the similar matrices based on correlation-functions, $\text{Corr}(x, y) = e^{-\rho|x-y|}$, in stead of the covariance, we can factorise out $\frac{\sigma^4}{\sigma^2}$ from the covariance-matrix, and then σ from σ_{sk} . Thus we have shown that σ is a scaling factor of the error.

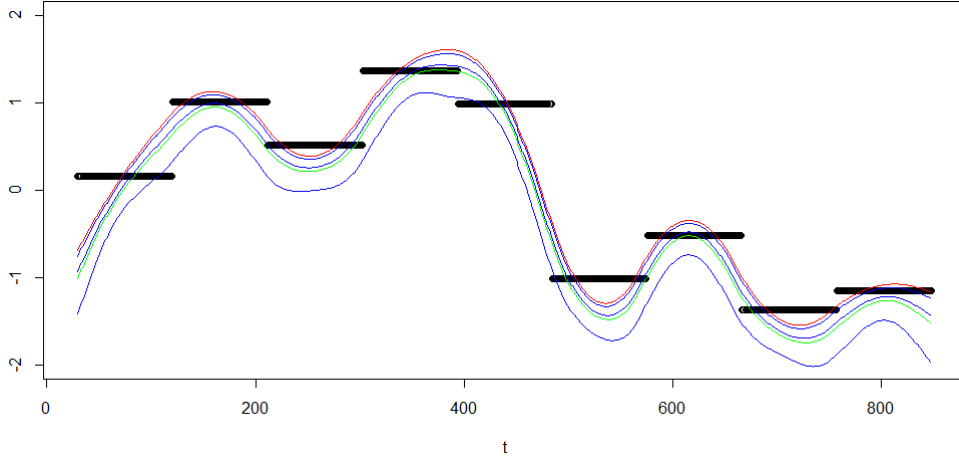


Figure 13: Predicted curves with errors

We get another confirmation for this when we see that the the line for $5 \cdot \sigma_{\text{sk}}^2$ coincide with that of when $\sigma = 5 \cdot \sigma_{\text{est}}$

Alternative Covariance Functions

Our initial choice of covariance function, the exponential covariance, is a simple and not very flexible one. In this section we will look at three more complex functions to see if this will make an improvement to our kriged curve. The covariances we will consider are given by:

Squared exponential covariance: $\text{Cov}(x, y) = e^{\frac{1}{2}\rho^2|x-y|^2}$

Matérn-covariance with $\nu = \frac{3}{2}$: $\text{Cov}(x, y) = (1 + \rho\sqrt{3}|x - y|)e^{-\rho\sqrt{3}|x-y|}$

Matérn-covariance with $\nu = \frac{5}{2}$:

$$\text{Cov}(x, y) = (1 + \rho\sqrt{5}|x - y| + \frac{\rho^2 5|x - y|^2}{3})e^{-\rho\sqrt{5}|x-y|}$$

To be able to look at the impact on the predicted curve from the different covariances, we first need to find each of the functions optimal values for ρ and σ . We will, once more, use the semivariogram for this.

For the squared exponential covariance we get the values $\sigma = -1.2363$ and $\rho = 0.6027$, while for the Matérn-covariances we get $\sigma = -1.2421$ and $\rho = 0.5597$ when $\nu = \frac{3}{2}$ and $\sigma = -1.2393$ and $\rho = 0.5787$ when $\nu = \frac{5}{2}$. That is, fairly similar values for all covariances.

In fig. 14 the old predicted curve is represented by the blue line, while the predicted curve based on the squared exponential covariance is given by

the green line. The Matérn-covariances are represented by the yellow ($\nu = \frac{3}{2}$) and black line ($\nu = \frac{5}{2}$).

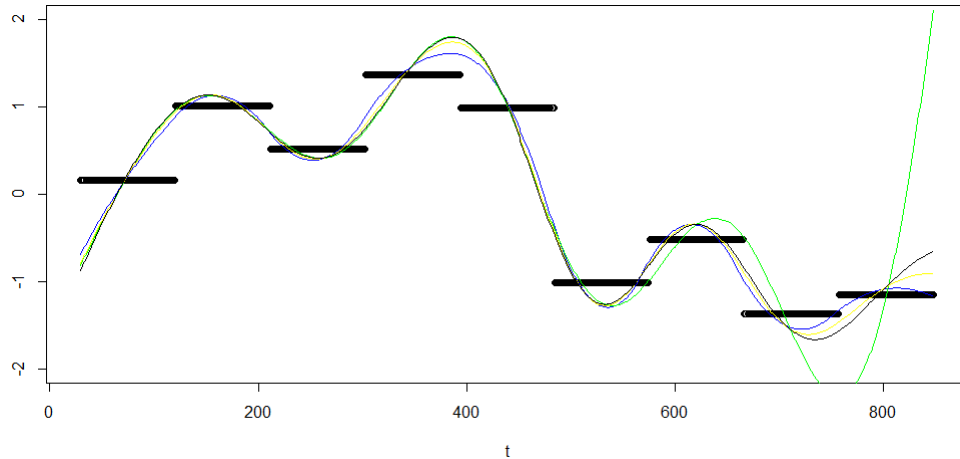


Figure 14: Predicted curves based on the different covariance functions

We see that, except for the squared exponential covariance after $t = 500$, there are small differences in the covariances paths, with the exponential covariance being the covariance that follows the data most tightly.

By plotting the error-curve for each of those covariances in fig. 15, where the red line is the original predicted curve, we see the same behaviour from the squared exponential covariance. It seems that the exponential covariance gives the biggest error. They are all plotted with 5 times the simple kriging error, to better show the differences.

These small differences that we see between most of the curves, tells us that the choice of covariance isn't too important.

5.2 Generation of a Gaussian Random Field

By following the algorithm from [Pic16], we can, with chosen covariance function and parameters σ and ρ , generate a Gaussian Random Field.

We're generating this field to control the method used in chapter 4 to find σ and ρ . We assumed only that the prices were a GRF, without knowing what covariance, ρ or σ the dataset could possibly have. Now we control this, and want to see if the method we used finds the correct values, and so trust our findings in the previous chapter.

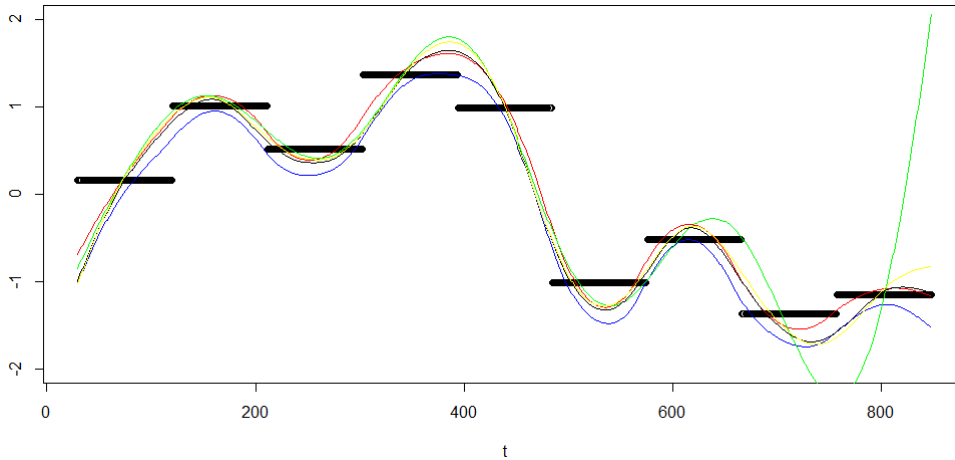


Figure 15: Predicted curves with error- functions based on the different covariance functions

So, with a stationary covariance-matrix, C , similar to what we saw in [ch.3], only that this we're not considering the integral of the covariance-function, the algorithm is given by:

1. Cholesky-factorise $C = BB^T$.
2. Generate $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$, where $\theta_i \sim \mathcal{N}(0, 1)$.
3. Compute $Y = B\boldsymbol{\theta}$ to get a vector of points from the GRF.

Now, we put this algorithm into R, with $n = 1000$, $\rho = 0.05$ and $\sigma = 0.4$ in the exponential covariance function we generate a Y of length 1000.

First we use the simple `sd`-function of R to find that the standard deviation of this vector, Y , is 0.4000175, and so very accurately describes σ .

We want to look at the method we used earlier to find the parameters. So we use the `nls`-method in R on a semivariogram, to find ρ and σ . This time the semivariogram is based on values calculated from the generated vector Y in stead of data points.

We put the standard deviation of Y that we found above as our σ^{init} , while for ρ^{init} we find that we get the same estimate for every $\rho^{\text{init}} \in [0.01, 0.5]$.

R gives us $\sigma = 0.39885$ and $\rho = 0.05043$. This is very accurate, and so this way we can be confident that our method gives us accurate values for ρ and σ from the dataset. However, this is based on $n = 1000$ points, while our assumed GRF is only represented by 9 data points. Putting $n = 9$ into our method, and running multiple times gives us a big range of values, and we can't really tell much from it. Changing the values for n , it seems that we need at least 500 observations to be confident in the results for σ and ρ .

The next step would be to have a method finding the covariance from the data, but as we saw, the different covariance-functions didn't make a great difference to the predicted curve, so we are not going to do this.

5.3 Change in Data

In this section we will look at the data's importance for the curve. Both the kriging-predicted curve and the Nelson-Siegel curve. We will see that because we are considering quarterly periods, seasonality will play a big role, and a change in one data-point affects at least one other, through the average seasonality-function.

The plots are all made with the same ρ and σ . In fig 16 and 17, we have only changed the first contract. In fig. 16 we put the first contract value to be 51, while in 17 we put it at 45. Note that the scaling of the axis is changed, to make it easier to compare fig. 17 to the original curve, as we only changed the contract slightly in this case.

We see that the predicted curve changes a lot, and so do the Nelson-Siegel curve. As kriging is an interpolation technique, it's bound to be very dependent on the data. We quickly notice, is that we're making the curve change the whole way through, and not just locally as one would assume at first. This follows from the way we defined our seasonality-function. By changing one data point we also affect at least one other data point. Had we considered a data set of monthly contracts, this wouldn't have been as apparent.

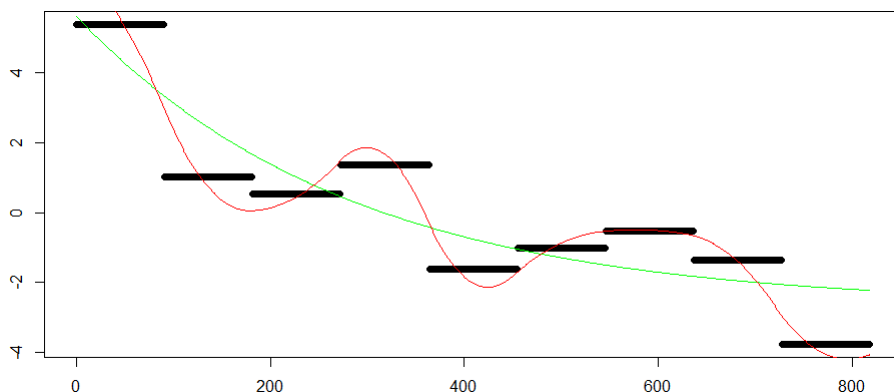


Figure 16: Predicted curve with first contract valued at 51

To avoid letting the seasonality have such an impact on the curve, we look at the curve without having subtracted the seasonality from the data.

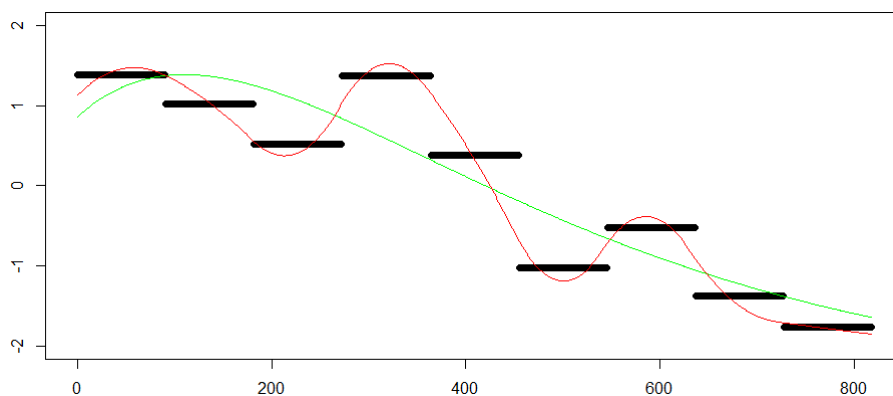


Figure 17: Predicted curve with first contract valued at 45

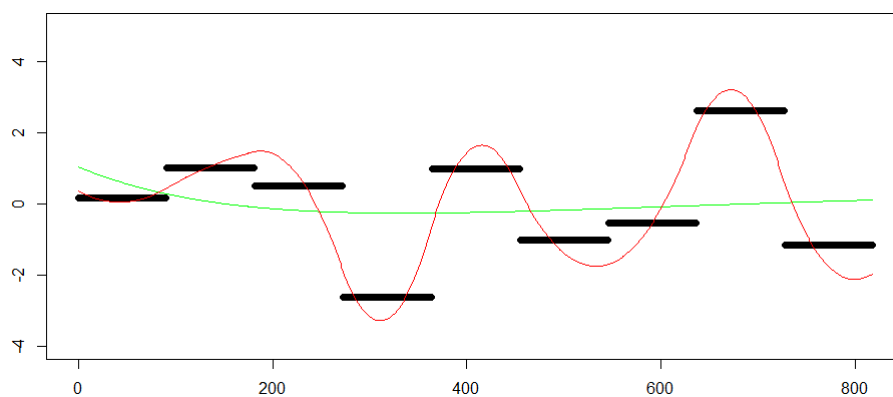


Figure 18: Predicted curve with third contract valued at 45

This makes the Nelson-Siegel curve very inaccurate, and therefore we'll leave it out of the plots.

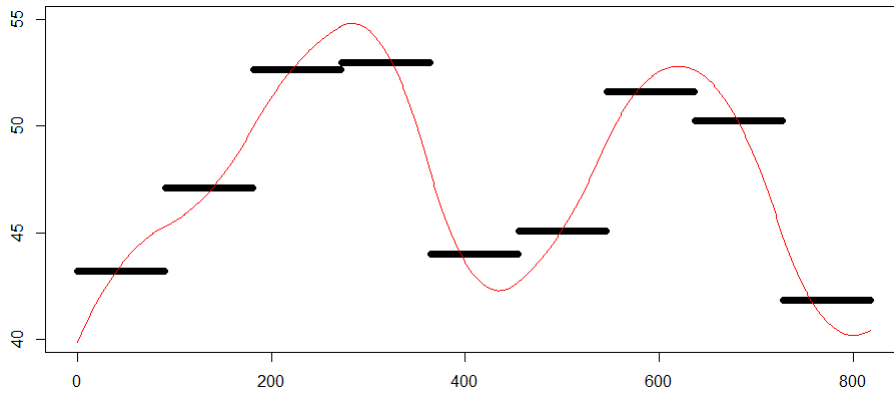


Figure 19: Predicted curve with first contract valued at 51, without seasonality

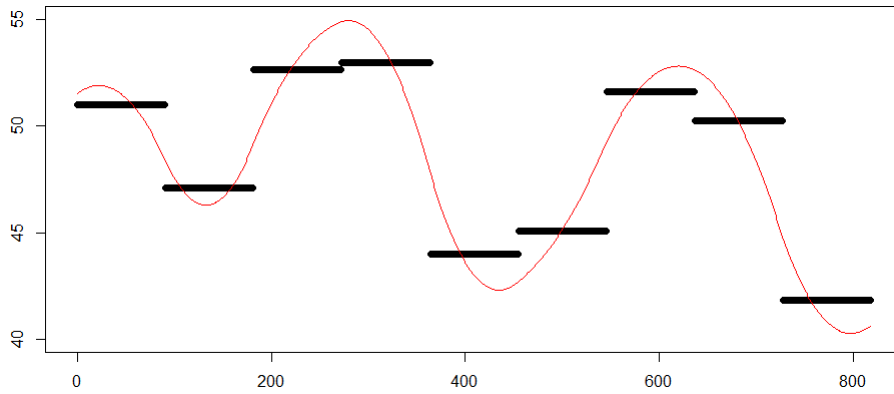


Figure 20: Predicted curve with first contract valued at 45, without seasonality

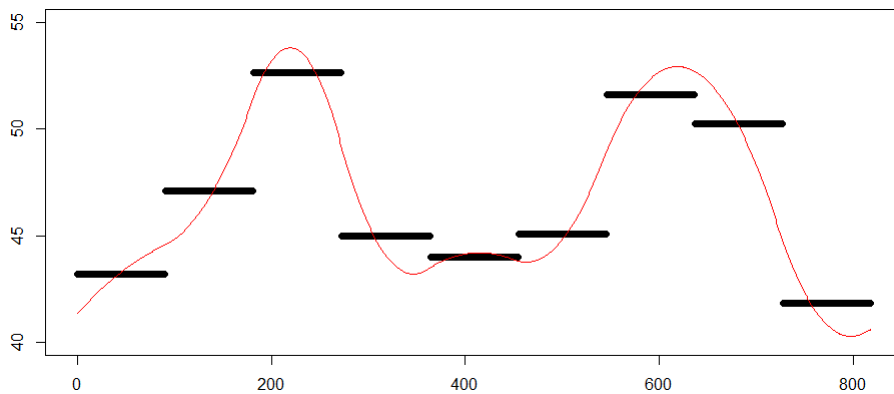


Figure 21: Predicted curve with third contract valued at 45, without seasonality

This time, we see that the curve mostly affects the closest contracts to the one we change, so a drastic change in the middle contract will, of course, make a different curve. However, as the curve interpolates all the data points, it doesn't change much outside of two closest points.

A different function for seasonality would also make the curve act differently. Although, switching a random contract remarkably, changes the seasonality. For example a sine-curve wouldn't be as appropriate if the data were no longer clearly cyclical.

Conclusion

The Nelson-Siegel curve changed clearly for different parameters, and to fit this to data one would appreciate a high accuracy in the estimation, and, as we saw, most important for α_0 . However, the predicted curve interpolates the data tightly, and it seems that only big changes in the parameters changes the curve.

Of course, when data is changed the predicted curve changes remarkably as it interpolates all the points. Especially in the case where we used the average function for seasonality, the whole process is changed, and so the curve will look very different.

Our initial estimates are mathematically valid for α_0 and α_1 , and we saw that α_2 follows from this. But, for σ and ρ we conclude that we can't say much about the certainty of the estimation. It seems that 9 data points isn't enough to do this.

We saw that σ didn't do much to the curve, and ρ had a very clear impact, which just describes the correlation to data, or dependence on mean function. Therefore, we know what the quantification tells us, and how to analyse the result of the two parameters, for different values.

6 Pricing of Put and Call-options

In this chapter we will look at pricing of options. To do this we will use the Black-76 formula.

Usually this takes the current futures price as a parameter, but our kriged estimate of the futures price is a normal distribution. Because of this, the option price will also be a distribution. We will look at how this distribution also turns out to be normal.

In the second section we will show in various plots, how the price of the call option varies for the different parameters, before we, in the last section, look at how the uncertainty of the futures contract is affecting the uncertainty to the option price.

6.1 European options and the Black-76 Formula

A *call option* is a financial contract on a commodity between two parts, where the buyer of the option has the right to buy the commodity for a *strike price* K , before an *exercise time* T , both agreed on at the start of the option. This sounds similar to the futures contract we've been working with earlier, but in this case, the buyer has the option not to buy it. That is, he can choose not to exercise the option. If the market price at the exercise time, $F(T)$, is lower than the strike price he will choose not to buy. In this case he would rather just buy it at market price. If it's higher, then the buyer has made a profit of $F(T) - K$. This gives the payoff-function at the exercise time, T

$$\max(0, F(T) - K),$$

where the function $\max(x, y)$ is equal to the highest number of x and y .

A *put option* is an option where the buyer of the option has the possibility of selling a commodity for a price K at the exercise time. Similarly to the call option, the buyer of the option considers the market price when choosing whether to exercise the option or not. But this time he's making a profit if the market price is lower than the strike price. The put option has the payoff-function

$$\max(0, K - F(T)).$$

So while the call option is a way to secure a buyers position, the put-option secures the seller.

We say that the call option is *in the money* if the current market price is higher than the strike price. Oppositely, the put option is in the money if

the strike price is higher than the market price at the time. This is not the same as saying that the buyer is currently making money, as the *premium*, the price of the option, isn't considered in this.

The call option is *out of the money* if the strike price is higher than the market price, and *at the money* if they are the same. We will look deeper into this when considering different strike prices later.

Example 6.1.1. Consider again the current market price for a power contract on the 3rd quarter of 2019, that is 45.50 euros per MWh. It's possible to buy a call-option for this futures contract. If she buys a call option that secures her 2.000 MWh delivered through this quarter for a price of 40 euros per MWh, then immediately this option is in the money. The exercise time of this option would be sometime before the start of the 3rd quarter. If the price goes under 40, the option will not be exercised, and the buyer will only pay the premium. If the price at the end is still above 40, for example 42. The buyer has made a profit of $2000 \cdot (42 - 40) = 4000$ euros, not considering the premium.

We want to find a fair price for the premium on such an option. To do this we will use the Black-76 formula. Presented by Fischer Black in 1976 [Bla76].

It is similar to the Black-Scholes formula, but while B-S has spot prices at time t as the underlying asset, Black-76 has the current discounted futures price with maturity at time T . It assumes that the futures price process is log-normal. This is the formula used by EEX to price options.

With F being a log-normal futures price with constant volatility σ (note that this is a different volatility than σ_{esk}), K the strike price and Φ the standardised cumulative normal distribution function, define

$$d_1 = \frac{\ln(F/K) + (\sigma^2/2)T}{\sigma\sqrt{T}}$$

$$d_2 = \frac{\ln(F/K) - (\sigma^2/2)T}{\sigma\sqrt{T}},$$

then we have the following proposition.

Proposition 2. The Black-76 fair price for a call and put option on a log-normally distributed futures contract, F , is given by

$$c = e^{-rT}[F\Phi(d_1) - K\Phi(d_2)]$$

and

$$p = e^{-rT}[K\Phi(-d_2) - F\Phi(-d_1)]$$

Proof. See for example [Bla76] □

Option Pricing When Futures Contract Price is a Distribution

Usually the Black-76 takes in the parameters and gives out a single number as the price. But our futures price is, because of the kriging, not a number, but a distribution for each t . We assume that it is normally distributed, and we will use the predictor and error from chapter 3 as our mean and variance. That is $F_t \sim \mathcal{N}(f_{\text{pred}}(t), \sigma_{\text{sk}}^2(t))$.

Now, F is a factor in c , so the price of the call option will also be a distribution, but since F is also a factor in d_1 and d_2 , it's not obvious which one. In the case of the d_i 's, we're considering the logarithm of F , and we have to be careful for when F is a negative number. However, as we're not considering seasonality in this chapter, the values of $f_{\text{pred}}(t)$ will be within $[37.9, 53.2]$ and the distribution will rarely, if ever, have negative numbers. We can just remove the few appearances we find without it making any significant difference, and so it won't affect the results.

To see the distribution, we generate values for F_t as described above and put them into the Black-76 formula for the call option price to create a histogram. We do $n = 10000$ simulations with $t = 300$, $K = 35$, $T = 300$, $\sigma = \frac{0.3}{\sqrt{252}}$ and $r = 0$. The histogram is plotted in fig. 22, and it seems to be normally distributed. We plot the density-adjusted histogram with the Gaussian density function in fig 23 and get this confirmed.

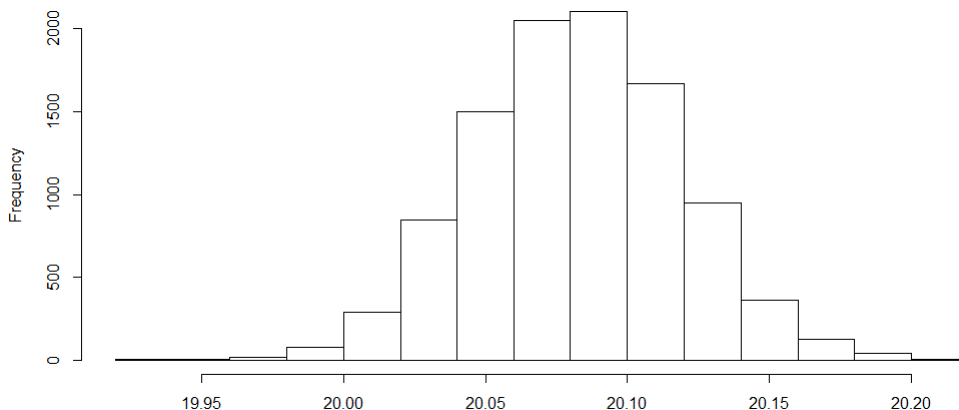


Figure 22: Histogram of option prices

Why the option price has the same type of distribution is explained by how strong F is compared to the d_i 's in c . The logarithm of F from d_1 and d_2 , varies less than F on its own. And when using the normal cdf the difference to F on its own gets even bigger, as $0 \leq \Phi \leq 1$. This means that

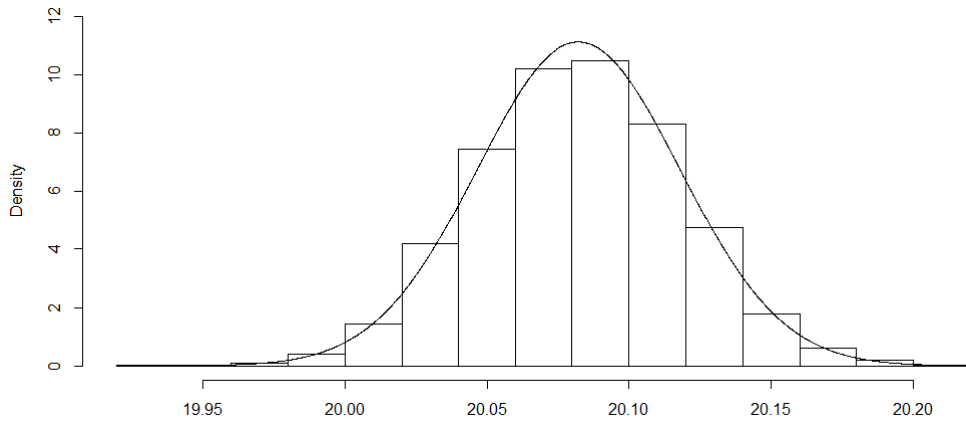


Figure 23: Normally adjusted histogram with pdf-curve

c is close to the form $c = aF + b$. And so it follows from basic statistics that c will be normally distributed when F is.

Now, in fig. 24 we plot the same adjusted histogram for the put price with the corresponding density function, and it's no surprise that this is also normally distributed.

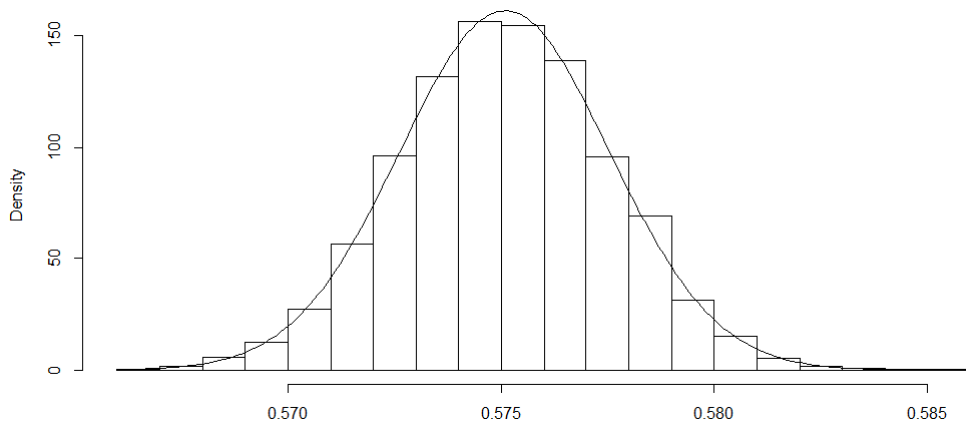


Figure 24: Histogram of put-option

This way we can say something about the uncertainty of an option price, and know that it isn't biased, but we can't quantify it yet, and neither relate it to the variance of the futures contract price.

We will do this in the last section, but first get more familiar with the different parts of the price function, by plotting them and look at how the price changes with them.

6.2 Plots of the Option Price

We put the default values $K = 35, t = T = 300$ and $\sigma = \frac{0.3}{252}$ and consider the expected value while plotting the call option price against the different variables

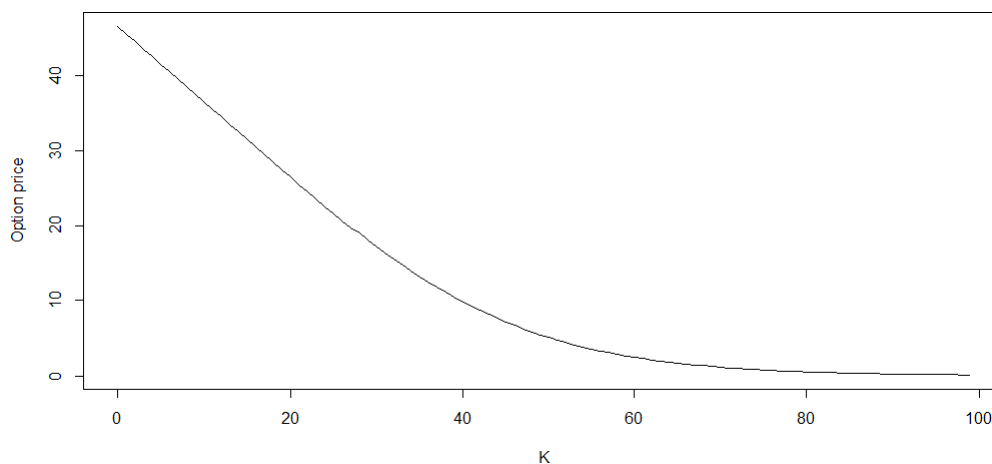


Figure 25: Option price as function of strike price

In fig. 25 the option price is plotted for different values for the strike price, and we see that the option price sinks as K gets bigger. The higher the price you agree to have the option to pay in the future, the more likely it is that the market price is staying lower than that of the strike price. And so the value of the option to buy the contract seems less attractive and the price decrease. Similarly to buy it for a very cheap price in the future is very attractive, and hence, very expensive. From the plot we see that at 0 it is the same as the price of the futures price, and then decreases, and gets close to 0 for very high K .

Fig 26, on the other hand, which shows the option price for an increasing volatility, is growing. This makes sense in the way that if the underlying asset has a high volatility, it's harder to tell what the price might be in the future and so the the price rises towards the actual price of the futures price.

The reason the option price has a upper limit at this price is that it doesn't make sense to pay more for an option on a contract than the contract is worth on its own. You might as well just buy the contract.

The option price increases when the exercise time is getting larger. This follows from that with a longer timespan, bigger difference of the price can occur. Similar to how it increases with volatility.

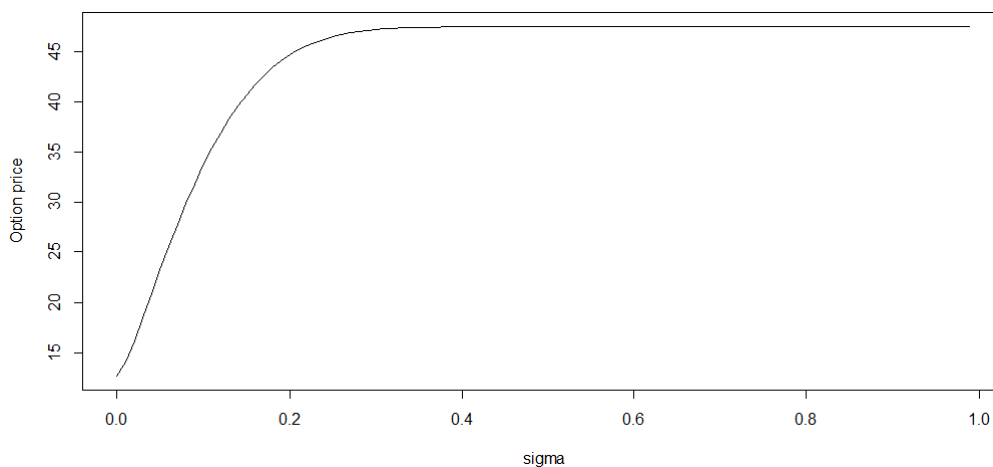


Figure 26: Option price as function of volatility

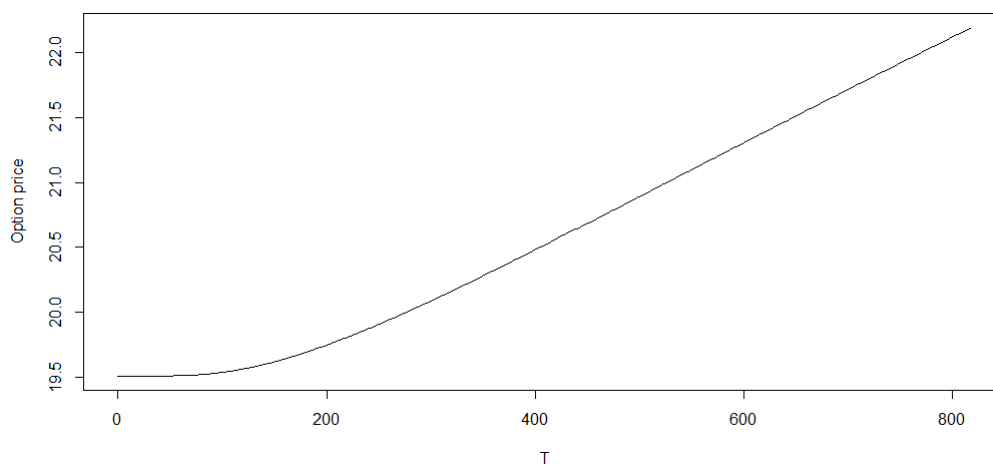


Figure 27: Option price as function of exercise time

6.3 Uncertainty of the Option Price

Since the underlying futures price has an uncertainty, it will be uncertain with any estimation technique, there will also be an uncertainty in c following from this. We have a way to quantify this uncertainty, and want to look deeper into this.

We plot the variance of the option price together with different values for simple kriging error in fig 28, and see that the two has a linear relationship.

Of course the variance of the option price is much smaller than the variance of the futures price, so what's more interesting is to consider the relative variance. That is, the standard deviation of our call price values

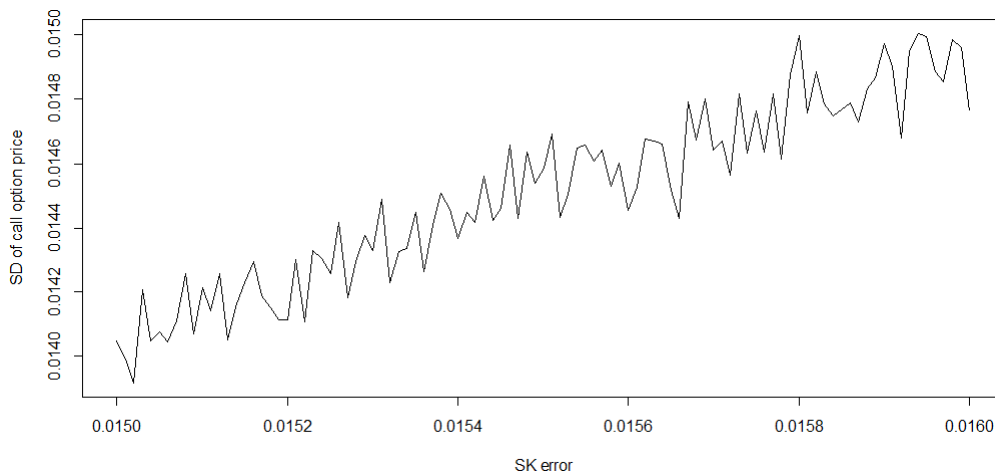


Figure 28: Relationship of kriging error and deviation of option price

divided by the mean.

We find that this value is nearly six times bigger than the relative variance of the kriged estimate of the futures price. This makes sense, when thinking that the variance of the option price is affected by F in addition to other varying elements.

But trying different values for parameters, we find that this relative variance changes a lot. For example, by letting $T \rightarrow 800$, we have already seen that the call option price rises. At the same time this leads to $\Phi(d_1)$ getting close to 1 and $\Phi(d_2)$, and so the variance sinks, making the relative variance change a lot.

As already mentioned, F varies much more than $\Phi(d_1)$ and $\Phi(d_2)$. In fact the variation are at times 100 times bigger, and therefore we want to look at the relative variance of the call option as if it actually were a function on the form $c = aF + b$.

We know that then $\text{sd}(c) = a^2\text{sd}(F)$, and so we have

$$\frac{\text{Var}(c(t))}{\mathbb{E}(c(t))} = \sigma_{\text{rel}}^2(t) = \frac{\Phi(d_1)^2}{\Phi(d_1)f_{\text{pred}}(t) - K\Phi(d_2)}\sigma_{\text{sk}}^2(t) \quad (3)$$

Of course it's not easy to read this form, but considering, for example when $\Phi(d_2)$ goes towards 0, the relative variance of the price option is actually smaller than that of the futures price.

Conclusion

As the futures price is normally distributed with no bias, the size of the kriging-error doesn't actually affect the expected price of the option, it might as well turn out to be worth more than less, but considering the risk for bankruptcy more important to avoid than to gain the same positive value.

We have here found the form of the uncertainty of the put and call price. The fair price is calculated and presented on EEX, based on their settlement price, but there's no quantification of the uncertainty. We have quantified this, through firstly using kriging to describe the futures price, and then the kriging error for its uncertainty. Either through generation, or from eq. 3.

Of course this uncertainty isn't strictly about power prices, but for any commodity traded with futures contracts.

A R-code

```
avlest_data <- c(43.17, 47.10, 52.64, 52.99, 43.99,  
  45.06, 51.59, 50.24, 41.84)
```

```
l <- 90
```

```
#Finding average function for seasonality;
```

```
n <- 4
```

```
ses_vec <- numeric(n)
```

```
ses_vec_teller <- numeric(n)
```

```
data_nu <- numeric(n)
```

```
ses_vec_nu <- numeric(length(avlest_data))
```

```
j <- 2 #data starts with 2nd quarter
```

```
for(i in 1:length(avlest_data)) {  
  ses_vec[j] <- ses_vec[j] + avlest_data[i]  
  ses_vec_teller[j] <- ses_vec_teller[j] + 1  
  if(j == n) {  
    j = 1  
  }  
  else{j = j + 1}  
}
```

```
ses_vec <- ses_vec/ses_vec_teller
```

```
j = 2
```

```
for (i in 1:length(ses_vec_nu)) {  
  ses_vec_nu[i] <- ses_vec[j]  
  if(j == n) {  
    j = 1  
  }  
  else{j = j + 1}  
}
```

```

data_nu <- avlest_data - ses_vec_nu
#-----

avlest_data <- data_nu

n <- length(avlest_data)
avlest_data_nu <- (rep(avlest_data, each = 1+1))

t_nu <- c(0, 90, 180, 270, 360, 450, 540, 630, 720,
          810)
t <- seq(0, length(avlest_data_nu)-1, by = 1)

f_NS <- function(x, alpha_0, alpha_1, alpha_2, beta) {
  alpha_0 + alpha_1*exp(-beta*x)+alpha_2*beta*x*
  exp(-beta*x)
}

#Finding alpha_i's

beta_init <- 0.0042
a_0_init <- avlest_data[length(avlest_data)]
a_1_init <- avlest_data[1] - a_0_init
a_2_init <- 3

nls(avlest_data_nu~alpha_0 + alpha_1*exp(-beta_init*t)
    +alpha_2*beta_init*t*
    exp(-beta_init*t),
    start = list(alpha_0 = a_0_init, alpha_1 = a_1
    _init, alpha_2 = a_2_init))

#Finding beta
a_0 <- -2.375
a_1 <- 1.761
a_2 <- 7.537

```

```

nls(avlest_data_nu~a_0 + a_1*exp(-beta*t)+a_2*beta*t*
    exp(-beta*t),
    #data = est_data,
    start = list(beta = beta_init))

```

```
b <- 0.00416
```

```
#plot of Nelson-Siegel curve
```

```

b <- 0.00416
a_0 <- -2.375
a_1 <- 1.761
a_2 <- 7.537

```

```

k <- 30 #30 days between reading data and period
      starts

```

```
t <- seq(k, k + length(avlest_data)-1, by = 1)
```

```

plot(t, avlest_data_nu, ylim = c(-2,2), ylab = "")
lines(t, f_NS(t-30, a_0, a_1, a_2, b), type = "l", col
      = "green")

```

```
#Semivariogram
```

```
semivar_t <- seq(0, length(avlest_data)-1, by = 1)
```

```
Y_val <- avlest_data
```

```

semivar <- numeric(length(avlest_data))
for (i in 1:length(avlest_data)) {
  for (j in i:length(avlest_data)) {
    semivar[j-i+1] <- semivar[j-i+1] + ((Y
      _val[j]-Y_val[i])^2)
  }
}

```

```

for (i in 1:length(semivar)) {
  semivar[i] <- (semivar[i]/(length(semivar)-i
+1))
}

semivar <- (1/2)*semivar #semivariogram, not variogram

#Covariance functions fitted for semivariogram

Cov <- function(x, y, rho){
  exp(-rho*abs(x-y))
}

#least-squares estimate
est_data_sigma = data.frame(semivar_t, semivar)

rho_init <- 0.4027
sigma_init <- 0.4805
nug_init <- 0.1

nls(semivar ~ (sigma^2)*(1 - exp(-rho*semivar_t) + nug
),
    data = est_data_sigma,
    start = list(sigma = sigma_init, rho =
rho_init, nug = nug_init),
    lower = list(-Inf, -Inf, 0), algorithm
= "port")

#without nugget-effect

rho_init <- 0.4027
sigma_init <- 0.4805

nls(semivar ~ (sigma^2)*(1 - exp(-rho*semivar_t)),
    data = est_data_sigma,
    start = list(sigma = sigma_init, rho =

```



```
rho_init))
```

```
semivar_func <- function(cov_func, x, y, sigma, rho,
  nug){
  (sigma^2)*(1 - cov_func(x, y, rho))
}
```

```
#Plot of semivariogram:
```

```
sigma <- 1.2597
rho <- 0.4567
nug <- 0
```

```
y <- seq(0, length(semivar_t), by = 1)
h <- semivar_t
plot(h, semivar, ylab = "", ylim = c(0,1.8))
lines(y, semivar_func(Cov, y, 0, sigma, rho), type = "
  l")
```

```
sigma_nu <- sigma/sqrt(90)
rho_nu <- rho/90
```

```
#Covariance-functions
```

```
Cov <- function(x, y, sigma, rho){
  sigma^2*exp(-rho*abs(x-y))
}
```

```
C_Gaussian <- function(x, y, sigma, rho){
  sigma^2*exp(-(1/2)*(rho^2)*(x-y)^2)
}
```

```
C_Matern3 <- function(x, y, sigma, rho){
  sigma^2*(1 + rho*sqrt(3)*abs(x-y))*exp(-rho*(
    sqrt(3))*abs(x-y))
}
```

```

C_Matern5 <- function(x, y, sigma, rho){
  sigma^2*(1 + rho*sqrt(5)*abs(x-y) + (1/3)*(rho
    ^2)*5*
    (x-y)^2)*exp(-rho*(sqrt(5))*abs(x-y))
}

```

#Covariance-vector

```

c_x <- function(cov_func, x, x_i, sigma, rho, l){
  (1/l)*integrate(cov_func, x = x, sigma = sigma
    , rho = rho,
    upper = x_i +
      l, lower =
        x_i) [[1]]
}

```

```

y_hat <- function(x_i, alpha_0, alpha_1, alpha_2, beta
  , avlest_data, l)
  {avlest_data - (1/l)*integrate(f_NS,
    alpha_0 = alpha_0,
    alpha_1 = alpha_1,
    alpha_2 = alpha_2,
    beta = beta,
    upper = x_i + l,
    lower = x_i) [[1]]
}

```

#double integral

```

d_int <- function(cov_func, x_i, x_j, sigma, rho, l) {
  integrate(function(x) {
    sapply(x, function(x) {
      integrate(function(y)
        cov_func(x, y, sigma,
          rho), lower = x_j,
          upper = x_j+l)$
        value
      })
    }, lower = x_i, upper = x_i+l)
}

```

```

#multiplication matrices for predicted curve and error
mat_mult <- function(cov_func, y, alpha_0, alpha_1,
  alpha_2, beta, sigma, rho, avlest_data, t_nu, l) {
  y_hat_vec <- numeric(length(avlest_data))
  C_temp <- matrix(rep(0, len=length(avlest_data)
    )^2), nrow = length(avlest_data))

  for (i in 1:length(avlest_data)) {
    for(j in 1:length(avlest_data)) {
      C_temp[i,j] <- (1/l^2)*d_int(
        cov_func, t_nu[i], t_nu[j],
        sigma, rho, l)[[1]]
    }
    y_hat_vec[i] <- y_hat(t_nu[i], alpha_
      0, alpha_1, alpha_2, beta, avlest_
      data[i], l)
  }

  C_mat <- solve(C_temp)

  temp_vec <- numeric(length(avlest_data))
  for (i in 1:length(avlest_data)) {
    temp_vec[i] <- sum(C_mat[i,]*y_hat_vec
      )
  }
  vector_mult <- numeric(length(y))
  vector_mult_alt <- numeric(length(y))
  for (i in 1:length(y)){
    temp_sum <- 0
    for (j in 1:length(avlest_data)) {
      temp_sum <- temp_sum +
        c_x(cov_func, y[i], t_
          nu[j], sigma, rho,
          l)*temp_vec[j]
    }
    vector_mult[i] <- temp_sum
  }
  vector_mult
}

```

```

mat_mult_std <- function(cov_func, y, alpha_0, alpha_
  1, alpha_2, beta, sigma, rho, avlest_data, t_nu, l)
  {
    C_temp <- matrix(rep(0, len=length(avlest_data
      )^2), nrow = length(avlest_data))

    for (i in 1:length(avlest_data)) {
      for(j in 1:length(avlest_data)) {
        C_temp[i,j] <- (1/l^2)*d_int(
          cov_func, t_nu[i], t_nu[j],
          sigma, rho, l)[[1]]
      }
    }
    C_mat <- solve(C_temp)

    temp_vec_2 <- numeric(length(avlest_data))
    vector_mult <- numeric(length(y))
    vector_mult_alt <- numeric(length(y))
    for (i in 1:length(y)){
      temp_sum <- 0
      for (k in 1:length(avlest_data)) {
        temp_vec <- numeric(length(
          avlest_data))
        for (j in 1:length(avlest_data
          )) {
          c_x_vec <- c_x(cov_
            func, y[i], t_nu[j
            ], sigma, rho, l)
          temp_vec[j] <- c_x(cov
            _func, y[i], t_nu[j
            ], sigma, rho, l)*C
            _mat[j,k]
        }
        temp_vec_2[k] <- sum(temp_vec)
      }
      for (j in 1:length(avlest_data)) {
        temp_sum <- temp_sum +
          temp_vec_2[j]*c_x(cov_

```

```

                                func , y[i] , t_nu[j
                                ], sigma , rho , l)
                                }
                                vector_mult[i] <- sum(temp_vec*c_x(cov
                                _func , y[i] , t_nu[j] , sigma , rho , l
                                ))
                                vector_mult_alt[i] <- temp_sum
                                }
                                vector_mult_alt
}

```

```

f_pred <- function(cov_func , x , alpha_0 , alpha_1 ,
alpha_2 , beta , sigma , rho , avlest_data , t , l) {
  f_NS(x , alpha_0 , alpha_1 , alpha_2 , beta) +
  mat_mult(cov_func , x , alpha_0 , alpha_1 , alpha_
  2 , beta , sigma , rho , avlest_data , t , l)
}

```

```

std_error <- function(cov_func , x , alpha_0 , alpha_1 ,
alpha_2 , beta , sigma , rho , avlest_data , t , l) {
  sqrt(sigma^2 - mat_mult_std(cov_func , x , alpha
  _0 , alpha_1 , alpha_2 , beta ,
                                sigma , rho ,
                                avlest_data
                                , t , l))
}

```

#plot values

```

beta <- 0.00416
alpha_0 <- -2.375
alpha_1 <- 1.761
alpha_2 <- 7.537
sigma <- sigma_nu
rho <- rho_nu
l <- 90

```

```

t_nu <- c(0 , 90 , 180 , 270 , 360 , 450 , 540 , 630 , 720 ,

```

```

      810)
y <- seq(0, length(avlest_data_nu)-1, by = 1)

plot_t <- (rep(avlest_data, each = l+1))

k <- 30 #30 days between reading data and period
      starts
t <- seq(k, k + length(avlest_data_nu)-1, by = 1)

plot(t, plot_t, ylim = c(-2,2), ylab = "")
lines(t, f_NS(t-30, alpha_0, alpha_1, alpha_2, beta),
      type = "l", col="green")
lines(t, f_pred(Cov, t-30, alpha_0, alpha_1, alpha_2,
      beta, sigma, rho,
      avlest_data, t_nu, l), type = "l",
      col="red")

lines(y, f_pred(C_Gaussian, y, alpha_0, alpha_1, alpha_
      _2, beta, sigma,
      rho, avlest_data, t_nu, l),
      type = "l", col="blue")
lines(y, f_pred(C_Matern3, y, alpha_0, alpha_1, alpha_
      2, beta, sigma,
      rho, avlest_data, t_nu, l),
      type = "l", col="black")
lines(y, f_pred(C_Matern5, y, alpha_0, alpha_1, alpha_
      2, beta, sigma,
      rho, avlest_data, t_nu, l),
      type = "l", col="yellow")

#Error
lines(t, f_pred(Cov, t-30, alpha_0, alpha_1, alpha_2,
      beta, sigma, rho, avlest_data, t_nu, l) +
      std_error(Cov, t-30, alpha_0, alpha_1, alpha_
      2, beta, sigma, rho, avlest_data, t_nu, l)
      ,
      type = "l", col = "blue")

plot(t, std_error(Cov, t-30, alpha_0, alpha_1, alpha_

```

```

2, beta, sigma, rho, avlest_data, t_nu, l) ,
    type = "l", col ="blue", ylab = "sk error")

lines(t, f_pred(Cov, t-30, alpha_0, alpha_1, alpha_2,
    beta, sigma, rho, avlest_data, t_nu, l) -
    std_error(Cov, t-30, alpha_0, alpha_1, alpha_
        2, beta, sigma, rho, avlest_data, t_nu, l)
    ,
    type = "l", col ="blue")

lines(t, f_pred(Cov, t-30, alpha_0, alpha_1, alpha_2,
    beta, sigma, rho, avlest_data, t_nu, l) -
    std_error(Cov, t-30, alpha_0, alpha_1, alpha_
        2, beta, sigma, rho, avlest_data, t_nu, l)
    ,
    type = "l", col ="blue")

lines(t, f_pred(Cov, t-30, alpha_0, alpha_1, alpha_2,
    beta, sigma, rho, avlest_data, t_nu, l) -
    5*std_error(Cov, t-30, alpha_0, alpha_1, alpha_
        2, beta, sigma, rho, avlest_data, t_nu, l)
    ,
    type = "l", col ="blue")

lines(y, f_pred(C_Gaussian, y, alpha_0, alpha_1, alpha
    _2, beta, sigma, rho, avlest_data, t_nu, l) -
    5*std_error(C_Gaussian, y, alpha_0, alpha_1,
        alpha_2, beta, sigma, rho, avlest_data, t_
            nu, l) ,
    type = "l", col ="blue")

lines(y, f_pred(C_Matern3, y, alpha_0, alpha_1, alpha_
    2, beta, sigma, rho, avlest_data, t_nu, l) -
    5*std_error(C_Matern3, y, alpha_0, alpha_1,
        alpha_2, beta, sigma, rho, avlest_data, t_
            nu, l) ,
    type = "l", col ="black")

lines(y, f_pred(C_Matern5, y, alpha_0, alpha_1, alpha_
    2, beta, sigma, rho, avlest_data, t_nu, l) -
    5*std_error(C_Matern5, y, alpha_0, alpha_1,

```

```

        alpha_2, beta, sigma, rho, avlest_data, t_
        nu, 1) ,
type = "l", col ="yellow")

avlest_data <- c(43.17, 47.10, 52.64, 52.99, 43.99,
                45.06, 51.59, 50.24, 41.84)

#finding beta and alpha_i's

n <- length(avlest_data)
avlest_data_nu <- (rep(avlest_data, each = 1+1))

t_nu <- c(0, 90, 180, 270, 360, 450, 540, 630, 720,
          810)
t <- seq(0, length(avlest_data_nu)-1, by = 1)

f_NS <- function(x, alpha_0, alpha_1, alpha_2, beta) {
  alpha_0 + alpha_1*exp(-beta*x)+alpha_2*beta*x*
  exp(-beta*x)
}

beta_init = 0.013
a_0_init <- avlest_data[length(avlest_data)]
a_1_init <- avlest_data[1] - a_0_init
a_2_init <- 8

nls(avlest_data_nu~alpha_0 + alpha_1*exp(-beta_init*t)
    +alpha_2*beta_init*t*
    exp(-beta_init*t),
    start = list(alpha_0 = a_0_init, alpha_1 = a_1
    _init, alpha_2 = a_2_init))

nls-values:
alpha_0 <- 47.883

```



```

alpha_1 <- -9.348
alpha_2 <- 6.659
beta <- 0.013
rho <- 0.04805
l <- 90

ave_F <- sum(avlest_data)/length(avlest_data)

t_nu <- c(0, 90, 180, 270, 360, 450, 540, 630, 720,
          810)
y <- seq(0, length(avlest_data_nu)-1, by = 1)

n <- 10000

rho <- rho_nu
T <- 300
K <- 35
sigma <- sigma_nu
sigma_log <- 0.3/sqrt(252)
r <- 0

f_t <- f_pred(Cov, T, alpha_0, alpha_1, alpha_2, beta,
              sigma, rho, avlest_data, t_nu, l)
err_t <- std_error(Cov, T, alpha_0, alpha_1, alpha_2,
                   beta, sigma, rho, avlest_data, t_nu, l)
F <- rnorm(n, f_t, err_t)

d_2 <- function(F, T, K, sigma_log) {
  (log(F/K) - ((sigma_log^2)/2)*T)/(sigma_log*
  sqrt(T))
}
d_1 <- function(F, T, K, sigma_log) {
  (log(F/K) + ((sigma_log^2)/2)*T)/(sigma_log*
  sqrt(T))
}
price_call <- function(F, T, K, sigma_log, r) {
  (F*pnorm(d_1(F, T, K, sigma_log), 0, 1) - K*
  pnorm(d_2(F, T, K, sigma_log), 0, 1))*

```

```

        exp(-r*T)
    }
    price_put <- function(F, T, K, sigma_log, r) {
        (K*pnorm(-d_2(F, T, K, sigma_log), 0, 1) - F*
         pnorm(-d_1(F, T, K, sigma_log), 0, 1))*
        exp(-r*T)
    }

```

```

p_c <- price_call(F, T, K, sigma_log, r)
p_p <- price_put(F, T, K, sigma_log, r)

```

```

sigma_2 <- sd(p_c)
mu <- sum(p_c)/n

```

```

#relative variance
err_t/f_t
sigma_2/mu

```

```

hist(p_c)

```

```

x <- seq(min(p_c),max(p_c), by = 0.0001)
hist(p_c, freq = FALSE, ylim = c(0,12))
lines(x, dnorm(x, mean(p_c), sd(p_c)))

```

```

x <- seq(min(p_p),max(p_p), by = 0.0001)
hist(p_p, freq = FALSE)
lines(x, dnorm(x, mean(p_p), sd(p_p)))

```

```

#Various plots

```

```

t <- seq(0, length(avlest_data_nu)-1, by = 1)

```

```

t_1 <- seq(0, 99, by = 1)
t_2 <- seq(0, 0.99, by = 0.01)

#plots with changing exercise time
mu_vec_T <- numeric(length(y))
sd_vec_T <- numeric(length(y))

for(i in 1:length(y)) {
  mu_vec_T[i] <- mean(price_call(F, i, K, sigma_
    log, r))
  #sd_vec_T[i] <- sd(price_call(F, i, K, sigma, r
    ))
}
plot(t, mu_vec_T, type = "l", xlab = "T", ylab = "
  Option price")
#plot(y, sd_vec_T, type = "l")

f_y <- f_pred(Cov, y, alpha_0, alpha_1, alpha_2, beta,
  sigma, rho, avlest_data, t_nu, 1)
err_y <- std_error(Cov, y, alpha_0, alpha_1, alpha_2,
  beta, sigma, rho, avlest_data, t_nu, 1)

#plots with changing delivery time
mu_vec_t <- numeric(length(y))
sd_vec_t <- numeric(length(y))
for(i in 1:length(y)) {
  F <- rnorm(n, f_y[y], err_y[y])
  mu_vec_t[i] <- mean(price_call(F, T, K, sigma_
    log, r))
  #sd_vec_t[i] <- sd(price_call(F, T, K, sigma, r
    ))
}
plot(y, mu_vec_t, type = "l")
#plot(y, sd_vec_t, type = "l")

```

```

#plots with changing exercise price
mu_vec_K <- numeric(100)
sd_vec_K <- numeric(100)
for(i in 1:100) {
  mu_vec_K[i] <- mean(price_call(F, T, i, sigma_
    log, r))
  #sd_vec_K[i] <- sd(price_call(F, T, i, sigma, r
    ))
}
plot(t_1, mu_vec_K, type = "l", xlab = "K", ylab = "
  Option price")
#plot(y_1, sd_vec_K, type = "l")

#plots with changing futures price
mu_vec_F <- numeric(100)
sd_vec_F <- numeric(100)
for(i in 1:100) {
  mu_vec_F[i] <- mean(price_call(i, T, K, sigma_
    log, r))
  #sd_vec_F[i] <- sd(price_call(i, T, K, sigma, r
    ))
}
plot(t_1, mu_vec_F, type = "l", xlab = "Futures price",
  ylab = "Option price")
#plot(y_1, sd_vec_F, type = "l")

#plots with changing sigma
mu_vec_sig <- numeric(100)
sd_vec_sig <- numeric(100)
for(i in 1:100) {
  mu_vec_sig[i] <- mean(price_call(F, T, K, i*
    0.01, r))
  #sd_vec_sig[i] <- sd(price_call(F, T, K, i*
    0.01, r))
}
plot(y_2, mu_vec_sig, type = "l", xlab = "sigma", ylab
  = "Option price")
#plot(y_1, sd_vec_sig, type = "l")

```

```

#changing of relative variances
err_vec <- seq(0.015,0.016, by =0.00001)
t_1 <- seq(1, length(err_vec), by = 1)
sd_vec_sig_2 <- numeric(length(err_vec))
mu_vec_sig_2 <- numeric(length(err_vec))
sd_pnorm_1 <- numeric(length(err_vec))
mu_pnorm_1 <- numeric(length(err_vec))
sd_pnorm_2 <- numeric(length(err_vec))
mu_pnorm_2 <- numeric(length(err_vec))
for(i in 1:length(err_vec)) {
  F <- rnorm(n, f_t, err_vec[i])
  sd_vec_sig_2[i] <- sd(price_call(F, T, K,
    sigma_log, r))
  mu_vec_sig_2[i] <- mean(price_call(F, T, K,
    sigma_log, r))
#   sd_pnorm_1[i] <- sd(pnorm(d_1(F, T, K, sigma),
#     0, 1))
#   mu_pnorm_1[i] <- mean(pnorm(d_1(F, T, K, sigma)
#     ), 0, 1))
#   sd_pnorm_2[i] <- sd(pnorm(d_2(F, T, K, sigma),
#     0, 1))
#   mu_pnorm_2[i] <- mean(pnorm(d_1(F, T, K, sigma)
#     ), 0, 1))
}

sum(err_vec/f_t)/n
sum(sd_pnorm_1/mu_pnorm_1)/n
sum(sd_pnorm_2/mu_pnorm_2)/n
sum(sd_vec_sig_2/mu_vec_sig_2)/n

plot(err_vec, sd_vec_sig_2, type = "l", xlab = "SK
  error", ylab = "SD of call option price")

```

```

sigma <- 0.4
rho <- 0.05
n <- 999

#covariance-functions:
Cov <- function(x, y, sigma, rho){
  sigma^2*exp(-rho*abs(x-y))
}

#Generation of field:
C_temp <- matrix(rep(0, len=n^2), nrow = n)

for(i in 1:n) {
  for(j in 1:n) {
    C_temp[i,j] <- Cov(i, j, sigma, rho)
  }
}

B <- chol(C_temp)

Y_gen <- numeric(n)
theta <- rnorm(n, 0, 1)

for(i in 1:n){
  Y_gen[i] <- sum(B[i,]*theta)
}

#Checking values:
sigma      #0.4
sd(Y_gen)  #0.4025704

#Semivariogram with Y_gen similar to data points when
  finding f_pred:

Y_val <- Y_gen
semivar_t <- seq(0, n-1, by = 1)

```

```

semivar <- numeric(n)
for (i in 1:n) {
  for (j in i:n) {
    semivar[j-i+1] <- semivar[j-i+1] + ((Y
      _val[j]-Y_val[i])^2)
  }
}
for (i in 1:length(semivar)) {
  semivar[i] <- (semivar[i]/(length(semivar)-i
    +1))
}

semivar <- 1/2*semivar #finding semivariogram and not
  variogram

semivar_func <- function(cov_func, x, y, sigma, rho){
  (sigma^2)*(1 - cov_func(x, y, sigma, rho))
}

#least-squares method for finding parameters:
sigma_init <- sd(Y_gen)
rho_init <- 0.05
est_data_sigma = data.frame(semivar_t, semivar)

nls(semivar ~ (sigma^2)*(1 - exp(-rho*semivar_t)),
  data = est_data_sigma,
  start = list(sigma = sigma_init, rho =
    rho_init))

#plot of semivariogram:
sigma <- 0.39885
rho <- 0.05043

y <- seq(0, length(semivar_t), by = 0.1)

plot(semivar_t, semivar)
lines(y, semivar_func(Cov, y, 0, sigma, rho), type = "
  l")

```

References

- [APK12] Zdravka Aljinovic, Tea Poklepovic and Kristina Katalinic. “Best fit model for yield curve estimation”. In: *Croatian Operational Research Review* 3.1 (2012), pp. 28–40.
- [Bla76] Fischer Black. “The pricing of commodity contracts”. In: *Journal of Financial Economics* 3.1 (1976), pp. 167–179. ISSN: 0304-405X. DOI: [https://doi.org/10.1016/0304-405X\(76\)90024-6](https://doi.org/10.1016/0304-405X(76)90024-6). URL: <http://www.sciencedirect.com/science/article/pii/0304405X76900246>.
- [CW11] Noel Cressie and Christopher K. Wikle. *Statistics for spatio-temporal data*. Wiley Series in Probability and Statistics. John Wiley & Sons, Inc., Hoboken, NJ, 2011, pp. xxii+588. ISBN: 978-0-471-69274-4.
- [EEX] EEX. *Phelix-DE Futures*. URL: <https://www.eex.com/en/market-data/power/futures/phelix-de-futures#!/2019/03/01> (visited on 14/05/2019).
- [EEX18] EEX. *Settlement Pricing Procedure*. 2018. URL: <https://www.eex.com/blob/61950/6868332159cb71d90850ba2ab45ffdea/settlement-pricing-procedure-data.pdf> (visited on 14/05/2019).
- [ene18] Norges vassdrags- og energidirektorat. *Nasjonal varedeklarasjon 2017*. 2018. URL: <https://www.nve.no/reguleringsmyndigheten-for-energi-rme-marked-og-monopol/varedeklarasjon/nasjonal-varedeklarasjon-2017/?ref=mainmenu> (visited on 13/05/2019).
- [HR12] Hana Hladíková and Jarmila Radová. “Term Structure Modelling by Using Nelson-Siegel Model”. eng. In: *European Financial and Accounting Journal* 7.2 (2012), pp. 36–55. ISSN: 1805-4846. DOI: 10.18267/j.efaj.9. URL: <http://hdl.handle.net/10419/109890>.
- [Nor19] NordPool. *NordPool About us*. 2019. URL: <https://www.nordpoolgroup.com/About-us/> (visited on 12/05/2019).
- [NS12] Charles R. Nelson and Andrew F. Siegel. “Parsimonious modeling of yield curves [reprint of J. Bus. 60 (1987), no. 4, 473–489]”. In: *Internat. Lib. Crit. Writ. Econ.* 267 (2012), pp. 671–687.
- [Pic16] Géraldine Pichot. *Algorithms for Gaussian random field generation*. Technical Report RT-0484. INRIA Paris, Dec. 2016. URL: <https://hal.inria.fr/hal-01414707>.

[Reg] Regjeringen. *Kraftmarkedet og strømpris*. URL: <https://www.regjeringen.no/no/tema/energi/stromnettet/kraftmarkedet-og-strompris/id2076000/> (visited on 14/05/2019).