Real Options in Energy Investments

Erik Johansson

August 7, 2010

Contents

Abstract

While the net present value method is becoming increasingly unpopular due to its inflexible nature, the concept of real options has slowly but surely been gaining academic ground. To be able to properly valuate a gas fired turbine, it is absolutely essential to take into account the *real option* associated with the freedom to, at all times, switch the turbine on and off. This thesis mainly provides analysis of a binomial tree method that is used to correctly valuate a gas fired turbine. Additionally, a framework for optimal switching problems developed by Paul Fackler is applied to the problem.

It is concluded that the binomial tree method very accurately, with a relative error of up to 1.5%, solves the valuation problem if switching can be done without a cost. For problem settings in which region switching is associated with a fixed cost, the application of Fackler's framework is more suitable. Depending on energy price modeling, it is shown that the introduction of switching costs lower the real option value by up to 30%.

Zusammenfassung

Während die Kritik zur Kapitalwertmethode immer lauter wurde, hat die Idee der Real Option - Betrachtung langsam aber sicher einen festen Platz in akademischen Zirkeln eingenommen. Um das Gewinnpotenzial einer gasbefeuerten Turbine richtig bewerten zu können, ist es unumgänglich die Möglichkeit, zu jeder Zeit den Betrieb abzubrechen bzw. wieder aufzunehmen, als Real Option in Betracht zu ziehen. Die vorliegende Diplomarbeit behandelt hauptsächlich Binomialbaumansätze zur korrekten Validierung gasbefeuerter Turbinen. Zusätzlich wird Paul Facklers Beitrag zu Optimal Switching Problemen auf den vorliegenden Forschungsgegenstand angewandt.

Es stellt sich dabei heraus, dass Binomialbaumansätze bei kostenfreien Zustandsänderungen der Turbine sehr zufriedenstellende Lösungen mit einem relativen Fehler von bis zu 1.5% liefern. Für Problemstellungen, in denen oben genannte Kosten anfallen, liefert Fackler's Arbeit die bessere Herangehensweise. Abhänging von den jeweils gewählten Energiepreismodellen wird gezeigt, dass die Einführung von Zustandsänderungskosten den Optionswert um bis zu 30% reduziert.

Sammanfattning

I takt med att kritiken blivit allt hårdare mot net present value-metoden har analys av så kallade reella optioner vunnit allmänt erkännande i den akademiska världen. För att kunna bestämma värdet hos en gaseldad turbin på ett riktigt sätt är det av yttersta nödvändighet att ta hänsyn till den *reella option* som existerar till följd av att turbinen kan slås av och på vid vilken tidpunkt som helst. Det här examensarbetet syftar huvudsakligen till att analysera huruvida en metod baserad på binomialträd kan tillämpas för att på ett korrekt sätt utvärdera en gaseldad turbin. Därutöver appliceras en metod utformad för optimering av kontrollbeslut utvecklad av Paul Fackler på problemet.

Slutsatsen är att metoden baserad på binomialträd löser, med ett relativt fel på högst 1.5%, problemet under förutsättning att växling av turbinläge kan genomföras kostnadsfritt. För de fall där en fixkostnad måste betalas för att växla turbinläge kan Facklers metod användas för att med stor noggrannhet lösa problemet. Det visar sig att, beroende på hur prisprocesserna för ström och gas modelleras, den reella optionen förlorar upp till 30% av sitt värde om växlingskostnader införs.

1 Introduction

1.1 Background

The most commonly used method for evaluation of possible investment opportunities is the net present value method (NPV). However, for situations in which flexibility plays a role the method systematically underestimates the investment value since it does not take into account that actions such as expanding or contracting a project or company could be a possibility. Naturally, if a project does better than expected, one would consider the possibility to expand the project and thus profit even more. The net present value method can not be used to valuate such possibilities, which is exactly why it fails to properly determine the value of projects with high flexibility.

This is where real option analysis comes into the picture. It provides tools to take possibilities such as expansion, contraction, abandonment and exchange into account.

1.2 Problem description

This thesis deals with the problem of properly evaluating power stations driven by gas fired turbines. Attempting to solve the problem with traditional methods such as the net present value method does in many cases result in an underestimation of the power station value, which indeed may result in the loss of a business opportunity.

The reason why the net present value fails to correctly evaluate the power station is because it does not capture the flexibility provided by the option to turn the turbine on and off. To assume that the turbine always would be operating is much too strong, since that would mean that one would let the turbine run even on days on which the cost of the gas that needed to supply the turbine would exceed the profit made by selling the generated electricity.

Thus, to be able to correctly determine the power station value, one needs to evaluate the switching option embedded in the investment opportunity, which is essentially what this thesis attempts.

In particular, the aim of the thesis is to study the binomial tree method. This is the case for two reasons; mainly because an analysis of the method allows us to be able to determine if it can be used to solve the power station problem when switching costs are present, but also because a modified version of it could be used to quickly give an approximate solution to the problem if data from only a single time series is available.

The fact that switching costs might be present when a gas fired turbine is switched on and off is often neglected in many analyses of power stations, but might of course play a role if they are large enough.

The latter reason, the fact that the method could give a fast approximate solution, is of great interest when one is considering to invest in a power station, since a fast approximate solution can be used as a test to determine whether a more extensive analysis of the investment opportunity could be worthwhile.

2 Literature Study

Even though a recently introduced concept, there are numerous texts available on the subject of real options.

Comprehensive introductions to the theme of real options include the texts by Copeland and Antikarov (2003), Schwartz and Trigeorgis (2003), Dixit and Pindyck (1994) and Brennan and Trigeorgis (2000).

In their book, Copeland and Antikarov discuss valuation techniques of several types of real options. They also present computer methods which can be used to valuate real options numerically. The binomial tree method which is discussed in later sections is inspired by ideas presented in the chapter on switching options.

Similarly, the text books by Trigeorgis and Schwartz and Trigeorgis and Brennen provide robust introductions along with an extensive set of examples.

Dixit and Pindyck discuss, among other topics, dynamic optimization and the solution of optimal stopping problems, which indeed are closely related to optimal switching problems.

The concept of using Monte Carlo simulations to valuate real options is discussed in several mathematical papers. One example is *The Real Option to Fuel Switch in the presence of Expected Windfall Profits under the EU ETS* by Luca Taschini and Simon Urech. Here, a Monte Carlo method is used to evaluate a hybrid power station while taking the European Union Emission Trading Scheme (EU ETS) into account. Some emphasis is given to the modeling of energy price processes.

In Dempster and Hong (2000), a fast Fourier transform method is proposed for valuation of spread options. This theory is applicable to the power station problem.

In their paper *Optimal Switching in an Economic Activity under Uncertainty*, Brekke and Oksendal derive general solution conditions to optimal switching problems. However, these conditions can not be solved explicitly for general switching problems. Paul Fackler (2004) propose a numerical method to solve these methods in his paper *Solving Optimal Switching Models.*

Due to their importance for this thesis, the texts by Copeland and Antikarov (2003), Dempster and Hong (2000), Paul Fackler (2004) and Brekke and Oksendal (1994) are discussed in greater detail in the later sections.

3 Preliminaries

3.1 Mathematical modeling

Of course, in order to be able to deal with the problem mathematically, we first need to model it. The following subsections discuss the problem model and price process models.

3.1.1 The problem model

We are considering a power station which requires k m^3 of gas to generate 1 kWh of electricity. The profit Y at time t, t being measured in days, can then be expressed by the model

$$Y(t) = N\big(P_e(t) - kP_g(t)\big) - OPC(t)$$

Where $P_e(t)$ is the price of 1 kWh of electricity at time t, $P_g(t)$ the price of 1 m^3 of gas at time t and $OPC(t)$ the operating costs of the power station at time t. N denotes the number of kWh of electricity that can be produced per day by the turbine. For simplicity, N will be set to 1 throughout this thesis since it only rescales the profit and therefore does not have an impact on the choice of solution method.

Moreover, the turbine may be switched on and off at any time time t_i, $t_i \in \mathbb{N}$, $i \in \mathbb{N}$. Switching the turbine on is done at an additional cost c. The life time of the power station is denoted with T.

The possibility to switch the turbine on and off can be identified as a real option.

3.1.2 Modeling prices with stochastic processes

Since $P_e(t)$, $P_g(t)$ and even $OPC(t)$ are random as seen from today, we need to find proper stochastic models to capture their behavior. The following discussion is based on information from Geman (2005).

By considering historical data of commodity prices, one will find that the price of most commodities do not have a strong drift, but rather tend to fluctuate around some rough mean value. Gas and electricity are no exceptions here, but their volatilities are higher than most other commodities. Naturally, the prices need also be non-negative at all times. Two different stochastic processes will be used to model the price processes in this thesis; the geometric brownian motion and the geometric Ornstein-Uhlenbeck stochastic process. The geometric brownian motion will be the process of choice to evaluate the binomial tree method, while the geometric Ornstein-Uhlenbeck process will be subject to the alternative methods.

To truly be able to capture the real dynamics of gas and electricity prices, more complex stochastic processes with stochastic volatility, stochastic correlation, seasonality, and jumps are required. Such models, however, will not be treated in this thesis.

Geometric brownian motion

The geometric brownian motion serves as a rough but reasonable model; it is described by the following stochastic differential equation

$$\frac{dS(t)}{S(t)} = \mu dt + \sigma dW(t)$$

Where σ is the volatility and μ the expected value of the daily returns. Since the processes are not expected to have drifts, μ will be set to 0, and by setting

$$f\big(S(t),t\big) = \ln\big(S(t)\big)$$

it follows from Itô's lemma that

$$df\big(S(t),t\big) = \Big(\frac{\partial^2 f\big(S(t),t\big)}{\partial S(t)^2}\frac{\sigma^2}{2}\Big)dt + \frac{\partial f\big(S(t),t\big)}{\partial S(t)}\sigma dW(t) = -\frac{\sigma}{2}dt + \sigma dW(t)$$

and thus $S(t)$ with the analytical solution

$$S(t) = S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(t)}$$

where S_0 is the initial price will be a martingale.

Proof. Set $m(t) := S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(t)}$. We want to show that $\mathbb{E}[m(t)|\mathcal{F}(s)] = m(s)$, $s < t$. $\mathcal{F}(s)$ denotes the natural filtration associated with $W(t)$.

$$\begin{aligned}
\mathbb{E}[m(t)|\mathcal{F}(s)] &= \mathbb{E}[S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(t)} e^{-\sigma W(s)} e^{\sigma W(s)} |\mathcal{F}(s)] \\
&= S_0 \mathbb{E}[e^{\sigma W_{t-s}} e^{\sigma W(s)} |\mathcal{F}(s)] e^{-\frac{\sigma^2}{2}t} \\
&= S_0 \mathbb{E}[e^{\sigma W_{t-s}}] e^{-\frac{\sigma^2}{2}t} e^{\sigma W(s)} \\
&= S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(s)} \frac{1}{\sqrt{2\pi(t-s)}} \int_{-\infty}^{\infty} e^{\sigma x - \frac{x^2}{2(t-s)}} dx \\
&= S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(s)} \frac{1}{\sqrt{2\pi(t-s)}} \int_{-\infty}^{\infty} e^{\frac{[x-\sigma(t-s)]^2+\sigma^2(t-s)^2}{2(t-s)}} dx \\
&= S_0 e^{-\frac{\sigma^2}{2}t+\sigma W(s)+\frac{\sigma^2(t-s)^2}{2(t-s)}} \underbrace{\frac{1}{\sqrt{2\pi(t-s)}} \int_{-\infty}^{\infty} e^{\frac{[x-\sigma(t-s)]^2}{2(t-s)}} dx}_{1} \\
&= m(s)
\end{aligned}$$

$\square$

Geometric Ornstein-Uhlenbeck process

The properties mentioned above are well captured by the geometric Ornstein-Uhlenbeck stochastic process which can be expressed in terms of the following stochastic differential equation

$$\frac{dS(t)}{S(t)} = h(m - S(t))dt + \sigma dW(t) \qquad \sigma,\, m \in \mathbb{R},\, k \in \mathbb{R}^+$$

Where m is the mean value, h the rate at which the process reverts to its mean and σ the volatility of the price.

3.1.3 Modeling the running costs

Even though the future running costs of the power station are random as seen from today, their dynamics are much simpler than the stochastic processes describing the energy prices. Indeed, the assumption that the running costs would follow a continuous stochastic process is like a GBM or GOU is too strong. Depending on the power station and the turbine, the daily running costs may follow several distributions, for example the uniform distribution or the log-normal distribution. The possibility even exists that the running costs are deterministic. For the mathematical analysis however, the running costs will not be of importance and therefore no particular model will be discussed.

4 Solution approaches

This section will discuss already existing, as well as recently suggested ideas of how to solve the power station problem. Emphasis will be given to the Copeland-Antikarov real options analysis (ROA) approach since the theoretical part of this thesis mainly deals with the binomial tree solution method which is based on the principles of ROA.

4.1 Simulation

A first naive way to go about solving the power station problem is to assume that the total costs associated with switching the turbine on and off are much smaller than the expected profit of the power station. Then, by simply setting the switching cost, c, to zero, it is possible to use straightforward Monte Carlo simulation to solve the problem.

Let $P_g^k(t)$, $P_e^k(t)$ and $OPC^k(t)$, denote the gas price, electricity price and the running costs at time t, $t \in 1, \dots, T$ of the k:th, $k \in 1, \dots, n$, $n \in \mathbb{R}_+$ realization, respectively.

Then the expected value of the total profit of the power station including the flexibility to turn the turbine on and off is simply expressed by

$$\mathbb{E}[\text{Profit}_{\text{flex}}] = \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{T} \Big[\max \big[P_e^k(t) - P_g^k(t), 0 \big] - OPC^k(t) \Big] e^{-r_f \frac{t}{250}}$$

and the expected profit without flexibility is expressed by

$$\mathbb{E}[\text{Profit}] = \frac{1}{n} \sum_{k=1}^{n} \sum_{t=1}^{T} \Big[P_e^k(t) - P_g^k(t) - OPC^k(t) \Big] e^{-r_f \frac{t}{250}}$$

and hence the switching option would have the value

$$\text{Option value} = \mathbb{E}[\text{Profit}_{\text{flex}}] - \mathbb{E}[\text{Profit}]$$

However, the above approximation is of course only valid if the switching cost is low compared to the profit of the turbine, and if the dynamics of the price processes do not cause the time intervals for which the turbine switched on to be too short.

Furthermore, a Monte Carlo simulation will be very time consuming if T is large and if a large number of simulations are required to get a decent approximation of the option value.

The concept of Monte Carlo simulation is explored further in the binomial tree method section.

4.2 Fast Fourier transform

Dempster and Hong (2000) have developed a method for spread option pricing which can be used in the power station problem setting if the same assumptions are made as in the above section. The method aims to evaluate the following expression

$$V(K,T) = \mathbb{E}_{\mathbb{Q}}\big[e^{-rT}[\mathbf{S}_1(T) - \mathbf{S}_2(T) - K]_+\big]$$

using a fast Fourier transform (FFT) method. In the above expression, $\mathbb{Q}$ is the unique risk-neutral measure. It is unique since it is assumed that the market is complete. T the time to maturity, K the strike, and $\mathbf{S}_1$, $\mathbf{S}_2$ the underlying asset prices.

The FFT-method can be used in problem settings in which the joint characteristic function of the underlying assets is known.

By setting $K = 0$, $\mathbf{S}_1(T) = P_e(T)$ and $\mathbf{S}_2(T) = kP_g(T)$, the expected total profit including the flexibility offered by the option to turn the turbine on and off is expressed by

$$\mathbb{E}[\text{Profit}_{\text{flex}}] = \sum_{t=1}^{T} V(0,t)$$

To calculate the value of the switching option, we also need the expected value of the total profit without flexibility. As discussed in the previous sections, the stochastic processes that model the prices of gas and electricity are often martingales, and if that is the case it holds that

$$\mathbb{E}[\text{Profit}] = \sum_{t=1}^{T} \mathbb{E}[P_e(t) - P_g(t)] = \Big\{P_e,\, P_g \text{ martingales}\Big\} = T(P_e^0 - P_g^0)$$

where P_e^0 and P_g^0 denote the initial prices of electricity and gas, respectively.

The switching option price is thus expressed by

$$\text{Option value} = -T(P_e^0 - P_g^0) + \sum_{t=1}^{T} V(0,t)$$

Note that the running costs $OPC(t)$ has been left out in the above equations since they cancel out in the option value expression. The expression is of course only valid if the energy price processes are martingales. Note however, that even though any mean reverting stochastic process without a drift is not a martingale, it has a permanent expected mean value which means that the above holds given that the initial values coincide with the expected values.

The reason why the FFT-method is superior to Monte Carlo simulation is because it significantly faster delivers more accurate results. Moreover, the computation time does not increase noteworthy if stochastic volatility or stochastic correlation are implemented to the model, which is a great advantage over Monte Carlo simulation. The fact that that stochasticity very easily can be added to the volatilities and correlations is extremely useful in the energy switching option problem setting since the stochastic processes ideally have these properties.

Thus, if the problem model requires advanced stochastic dynamics and the turbine is to be evaluated over a long period of time the FFT-method can significantly increase the option pricing time. However, since the joint characteristic function must be known analytically the method is not always applicable.

4.3 ROA: The Copeland-Antikarov approach

Since this thesis only deals with the evaluation of a switching option, no emphasis will be given to the details of evaluation of any other real option. However, to give some insight into the basic principles of real option analysis the following section gives a brief overview of the real option evaluation techniques presented by Copeland and Antikarov.

4.3.1 General concepts

To be able to determine the value of real options such as expanding, contracting, switching or abandoning a project, one needs to first consider the underlying dynamics of the project without the flexible options. By trying to model the uncertainties involved in the project and by looking at the different possible paths its value may progress through in the future, it is possible to analyze the these dynamics.

Proposed by Copeland and Antikarov (2003), the above concept can be utilized by constructing a binomial tree containing the possible present project values determined by the modeling. The different levels of the tree correspond to discrete time points in the future and the probabilities describe how likely it is that certain jumps in the tree will occur.

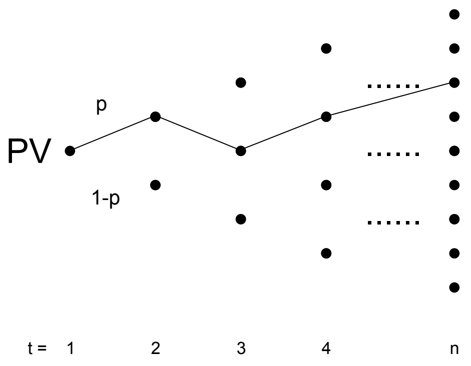

Figure 1: Binomial tree and one possible path the project value may take

Now, having constructed a binomial tree, it is possible to compare the node values without flexibility with the corresponding node values with flexibility. Any values in the binomial tree that are lower than the corresponding flexible value would be replaced by the latter. Thereafter the present project value with flexibility can be computed by reducing the tree, which is done differently depending on what kind of real option it is that should be evaluated. The value of the real option is then simply calculated as $PV_{0,flexibility}$ - PV_0.

4.3.2 Switching options

The Copeland and Antikarov (2003) evaluation approach of a switching option is done roughly as described above, but with the difference that two binomial trees are constructed based on the free cash flows generated by the different operation modus. Thus, two models of possible future cash flows for the two modus are required in order to construct the trees.

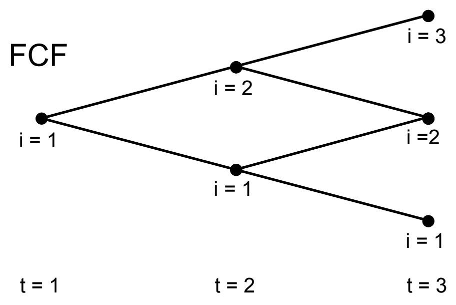

Figure 2: Binomial tree describing the free cash flows of modus A

If the cost for switching is zero, the switching option value can be determined by comparing the trees, and constructing a third tree that has node values equal to the higher of the corresponding node values of the two cash flow trees. The flexible value is then obtained by summing the expected values for all times t.

Let $FCF_n^i(t)$ be the i:th possible state, $i \in 1, 2, 3, \ldots, t$ at time t, $t \in 1, 2, 3, \ldots, T$ for operation modus n, $n \in A, B$ and let p_t^i, be the probability that the process is in state i at time t. By the above argument the flexible project value can be expressed as

$$PV_{0,f} = \sum_{t=1}^{T} \sum_{i=1}^{t} p_t^i \max[FCF_A^i(t), FCF_B^i(t)] e^{-r_f \frac{t}{250}}$$

However, if modus switching only can be done at an additional cost, an optimization problem has to be solved in order to know when to switch between the cash flow trees.

4.4 The Brekke-Oksendal approach

A more general setting of the switching option problem is treated in the paper by Brekke and Oksendal (1994). Namely, a setting in which a set of n discrete operation modus are available. The rate of returns for the different modus are given by a set of reward functions $\{f_k\}$ which may be stochastic. Furthermore, a set of costs associated with switching between the operation modus are introduced.

The power station problem is clearly a special case of the above problem with $n = 2$, reward functions equal to $f_1 = 0$, $f_2 = P_e(t) - kP_g(t)$, and an appropriate set of switching costs.

Here follows the somewhat simplified definitions leading up to the main result of [6] :

Define

$$Z(t) \in \{z_1, \ldots, z_m\}$$

to be the state of the system at time t. $z_1, \ldots, z_m$ denote indicator vectors

$$z = (a_1, \ldots, a_k)$$

where each a_i represents a state of the system. For example, for problems with two states, $a_i \in \{0, 1\}$

Let the $U(t)$ denote the stochastic process in $\mathbb{R}^n$ determining the values of the assets involved in the problem.

$$dU(t) = b(t, U(t), Z(t))dt + \sigma(t, U(t), Z(t))dB(t)$$

where $B(t)$ denotes an m-dimensional Brownian motion.

The system is expressed by the stochastic process

$$X(t) = \begin{bmatrix} t \\ U(t) \\ Z(t) \end{bmatrix}$$

Let

$$w = (\theta_1, \theta_2, \ldots, \theta_k, \ldots; \zeta_1, \zeta_2, \ldots, \zeta_k, \ldots)$$

denote an impulse control where $\theta_1, \ldots$ are stopping times and $\zeta_1, \ldots$ the new states entered at the corresponding stopping times.

Let

$$H(x, \zeta) > 0$$

denote the cost associated with switching from state x to state ζ.

Define the *value function* as

$$J^w(x) = E^x\Big[\int_s^\infty f(X(t)^{(w)})dt - \sum_{j=1}^{\infty} H(X(\theta_{j^-}), \zeta_j)\Big]$$

where $f(x)$ denotes the system returns for state x. The value function can be thought of as the total profit of the system.

Brekke and Oksendal have formulated a set of solution conditions in terms of the value function that must be satisfied in order to solve the problem. The solution conditions can be solved explicitly for certain problems in which geometric brownian motions determine the asset prices, but generally the conditions can not be solved explicitly. In the next section, these conditions are presented and solved using numerical methods proposed by Paul Fackler (2004).

4.4.1 The numerical solution

Paul Fackler (2004) has proposed a numerical solution approach that copes with the fact that the general solution conditions presented by Brekke and Oksendal cannot always be solved explicitly.

The rough idea behind the method is to rewrite the optimal value function conditions to be able to formulate them as an extended vertical linear complementary problem (EVLCP). To be able to to that, the optimal value function needs to be approximated by a family of basis functions. The EVLCP is then solved by a smoothing newton method proposed by Qi and Liao (1999).

To quickly summarize the contents of [3]; according to Brekke and Oksendal it holds that the optimal value function $V(S, R)$[1] satisfies

$$\rho(S)V(S,R) \geq f(S,R) + \mu(S,R)V_S(S,R) + \frac{\sigma^2(S,R)}{2}V_{SS}(S,R)$$

and

$$V(S,R) \geq V(S,x) - C_{Rx} \qquad \forall x \neq R$$

with one of the conditions satisfied with equality. Here, ρ denotes the risk-free interest rate.

The conditions are rewritten as

$$0 = \min\left(\beta(S,R) - f(S,R), \min_{x \neq R} V(S,R) - V(S,x) + C_{Rx}\right)$$

[1]Note that the notation differs between the papers. Fackler's definition of $S(t)$ corresponds to $U(t)$ with $dS = \mu(S,R)dt + \sigma(S,R)dW(t)$, C_{ij} corresponds to $H(x,\zeta)$, and R corresponds to the system state variable.

where

$$\beta(S,R) = \rho V(S,R) - \mu(S,R)V_S(S,R) - \frac{\sigma^2(S,R)}{2}V_{SS}(S,R)$$

Let $V(S,i)$ be approximated by $\phi(S)\theta_i$ with ϕ representing a set of n basis functions. θ_i denotes the coefficient vector associated with the i:th modus.

Define

$$\beta(S,i) = \rho\phi(S) - \mu(S,i)\phi^{'}(S) + \frac{\sigma^2(S,i)}{2}\phi^{''}(S)$$

after what the conditions can be written as

$$\beta(S,i)\theta_i - f(S,i) \geq 0$$

and

$$\phi(S)\theta_i - \phi(S)\theta_j + C_{ij}(S) \geq 0, \forall j \neq i$$

After finding θ_i for a set nodal state values of S, it is possible to define matrices Φ and B_i in which the functions $\phi(S)$ and $\beta(S,i)$ are evaluated. Let f_i denote $f(S,i)$ evaluated at the nodal state values.

The EVLCP can now be formulated as

$$\prod_{i=1}^{m} w_i = 0$$

with

$$w_i = M_i z + q_i \geq 0 \qquad i = 1, \ldots, m$$

and

$$M_i = e_i e_i^{\mathsf{T}} \otimes B_i + \left(I_m - \mathbf{1}_m e_i^{\mathsf{T}}\right) \otimes \Phi$$

$$q_i = \begin{bmatrix} C_{1i}\mathbf{1}_n \\ \vdots \\ C_{mi}\mathbf{1}_n \end{bmatrix} - [e_i \otimes f_i]$$

The above framework and the smoothing newton method is implemented in a matlab program written by Paul Fackler. The code is available at [7], and in the next section an implementation of the power station case is discussed.

4.4.2 Power station implementation

Fackler's framework is used to find optimal switching boundaries for the two cases where the energy prices follow geometric brownian motions and geometric Ornstein-Uhlenbeck processes. The matlab code that was used to implement the models and present the results can be found in the appendix.

Figure 3 and 4 show the approximate switching boundaries for price processes modeled as geometric brownian motions and geometric Ornstein-Uhlenbeck processes, respectively.

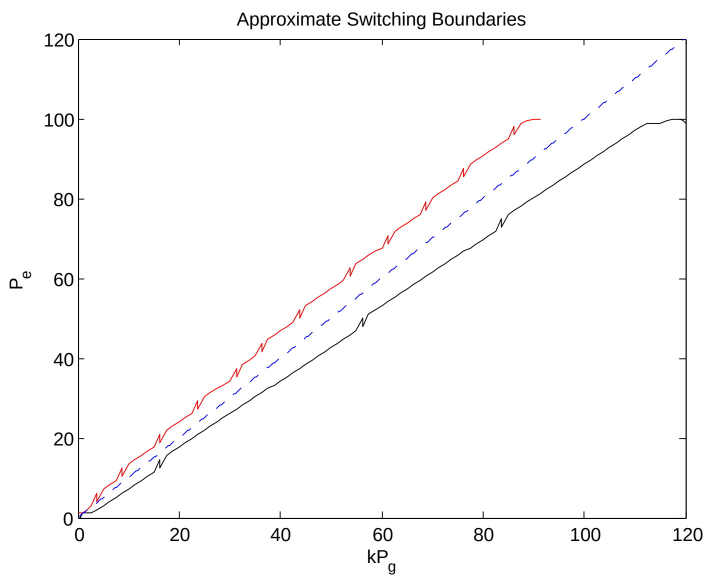

Figure 3: Optimal switching boundaries for a GBM

The blue and red curves represent the regions where the turbine optimally should be switched off and on, respectively. The dashed blue line is simply the function $y(x) = x$, which is plotted to emphasize the positions of the switching boundaries.

The area between the two curves grow larger when the price volatilities increase, since more stochasticity is added. A larger switching cost also increase the gap between the two curves due to the fact that is suboptimal to immediately switch the turbine off if the difference between $P_e(t)$ and $kP_g(t)$ would become negative.

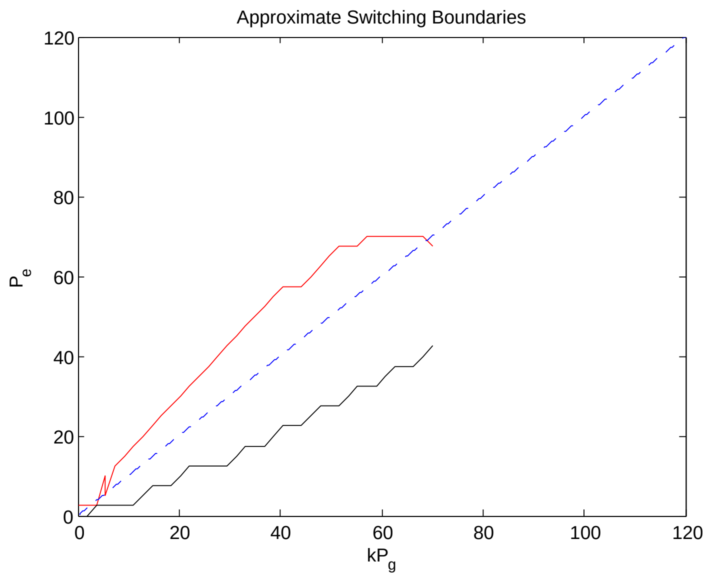

Figure 4: Optimal switching boundaries for a GOU

Due to the fact that geometric Ornstein-Uhlenbeck processes are mean-reverting, it is generally optimal to delay the option to switch the turbine off since it is expected that the processes return to their mean values. Therefore, if the rate of mean reversion is increased, the gap between the two switching boundaries become greater. Note that the red switching boundary crosses the dashed line representing $y(x) = x$ at around $x = 70$. This is a consequence of the fact that the approximate solution is somewhat inaccurate.

To compute the value of the switching option, a Monte Carlo simulation applying the optimal decision strategy obtained by the above solution is used. To do this, polynomials are calibrated to fit the approximate switching boundaries.

All results are presented in the results section.

5 The binomial tree method

This section will present the binomial tree solution idea along with the solution steps leading to the final results.

5.1 Overview

To avoid ambiguity, we define the project value as the present value of all future cash flows associated with the power station excluding the option to power it down. The flexible project value is defined analogously but includes the option to switch the power station on/off at any time t, $t \in 1, \dots, T$. We denote the project value PV_0 and the flexible project value $PV_{0,f,c}$ where c denotes the cost for switching.

We can state that since the running costs $OPC(t)$ associated with the power station always are present irrespective of whether the power station is powered down or not they will not play a significant role in the mathematical analysis of the problem. To account for the running costs, we just have to subtract their total expected value from the project value and the flexible project value, respectively. Thus, by setting

$$G(t) := P_s(t) - kP_g(t) \qquad E_O := \sum_{t=1}^{T} \mathbb{E}[OPC(t)]e^{-r_f \frac{t}{250}}$$

we have

$$\begin{aligned} PV_0 &= \sum_{t=1}^{T} \mathbb{E}[G(t)]e^{-r_f \frac{t}{250}} - E_O \\ PV_{0,f,0} &= \sum_{t=1}^{T} \mathbb{E}\big[\max[G(t), 0]\big]e^{-r_f \frac{t}{250}} - E_O \end{aligned}$$

$PV_{0,f,0}$ will be used in later calculations.

The value of the switching option is denoted $V(c)$, and now we have

$$V(c) = PV_{0,f,c} - PV_0$$

Since we have got stochastic models of the electricity and gas prices, a relatively accurate approximation of PV_0 can be found easily using a Monte

Carlo simulation. The expected value of the total running costs is found by the same argument.

The major difficulty presented by the power station problem is clearly finding $PV_{0,f,c}$. This will be done by considering a switching option as presented in the Copeland-Antikarov approach with $G(t)$ and a zero process as underlyings. Thus, to be able to solve the problem, we need to construct a discrete model of $G(t)$.

$PV_{0,f,c}$ will be obtained through the following steps.

Step 1:	Use a Monte Carlo simulation to draw random samples of $G(t)$
Step 2:	Use the kernel density method to estimate the probability distribution of $G(t)$
Step 3:	Use the kernel density estimate to construct binomial trees describing the discrete dynamics of $G(t)$
Step 4:	Use the binomial trees to solve an optimization problem to find $PV_{0,f,c}$

The above steps are treated theoretically and practically in the following sections.

5.2 Step one: Monte Carlo simulation

As mentioned, Monte Carlo simulations are used to get samples of the random variables $G(t)$ to determine PV_0 and certain probability distributions. All MATLAB code used to produce the results discussed in this section can be found in the appendix.

5.2.1 Theory

The underlying concept behind using a Monte Carlo simulation is to be able to solve problems that otherwise would be near to impossible or even impossible to solve analytically. The basic idea is to model the problem, draw random samples from the domain of feasible inputs and let them run through the model until some conclusion can be drawn from the output, often using arguments involving the law of large numbers. It can be considered a brute force computational algorithm to be used only if no explicit solution can be found. Depending on the complexity of the problem model, a Monte Carlo simulation can be very time consuming.

5.2.2 Application

In the case of simulating $G(t)$, the feasible domain of inputs would equal all realizations of the stochastic processes modeling the prices of electricity and gas.

First, we simulate n realizations $G_1(t), \ldots, G_n(t)$ of $G(t)$, $t = 0, \ldots, T$, thus drawing n paths of $G(t)$ from the domain of feasible paths. Analogously we draw $OPC_1(t), \ldots, OPC_n(t)$.

Taking the arithmetic average of the present values of the cash flows calculated from each of the realized paths of $G(t)$ results in an approximation of the present project value at time 0 by the following argument. Here PV_0^i denotes the i:th simulated value of PV_0. By setting $CF_t^i := G_i(t) - OPC_i(t)$ we have

$$\frac{1}{n}\sum_{i=1}^{n} PV_0^i = \sum_{t=1}^{T}\sum_{i=1}^{n}\frac{1}{n}\frac{CF_t^i}{(1+r)^t} \xrightarrow{n\to\infty} \sum_{t=1}^{T}\frac{\mathbb{E}[CF_t]}{(1+r)^t} = PV_0$$

The limit in the above calculation takes use of the law of large numbers.

In the following section, the n realized paths of $G(t)$ will also be used to find estimations of the probability distributions of $G(1), \dots, G(T)$

5.3 Step two: Probability density estimation

Since we need to estimate the distribution of $G(t)$, the following section discusses the kernel density estimation method.

5.3.1 Theory

There are many methods available for density estimation. The reason why the kernel density estimation method is chosen is because it estimates the density with several normally distributed kernels. The fact that the kernels follow a normal distribution will be used later in the solution when the binomial trees are constructed.

Kernel density estimation

The kernel density estimation of a probability density function $f(x)$ is generally defined as

$$\hat{f}(x) = \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) \text{ with } \int_{-\infty}^{\infty} K(x)dx = 1 \text{ and } K(x) \geq 0 \;\forall\; x \in \mathbb{R}$$

First, we can state that for normally distributed kernels $\hat{f}$ is a probability density function since

$$K(x) \geq 0 \qquad \forall x \in \mathbb{R}$$

and

$$\int_{-\infty}^{\infty} \hat{f}(x)dx = \int_{-\infty}^{\infty} \frac{1}{nh}\sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right) dx$$

$$= \frac{1}{nh}\sum_{i=1}^{n}\int_{-\infty}^{\infty} K\left(\frac{x - X_i}{h}\right) dx = \frac{1}{nh}\sum_{i=1}^{n}\int_{-\infty}^{\infty} \frac{1}{\sqrt{2\pi}} e^{\frac{-\left(\frac{x-X_i}{h}\right)^2}{2}} dx$$

$$= \frac{1}{nh}\sum_{i=1}^{n}\frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} e^{\frac{(x-X_i)^2}{2h^2}} dx = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\frac{1}{\sqrt{2\pi}h}\int_{-\infty}^{\infty} e^{\frac{(x-X_i)^2}{2h^2}} dx}_{1} = \frac{1}{n}\sum_{i=1}^{n} 1 = 1.$$

In the estimation $\hat{f}$, $X_1, \ldots, X_n$ denote the set of n samples drawn from the random variable X. The function K, the *kernel*, is any non-negative function integrating to unity over $\mathbb{R}$. The idea behind kernel density estimation is to center kernels around all data points and then sum them to obtain the probability density function. Specifically, a histogram is closely related to a kernel density estimation with kernels equal to "boxes". Thus, if K is any continuous function the kernel density estimate can be viewed upon as continuous version of a histogram characterized by the choice of kernel function.

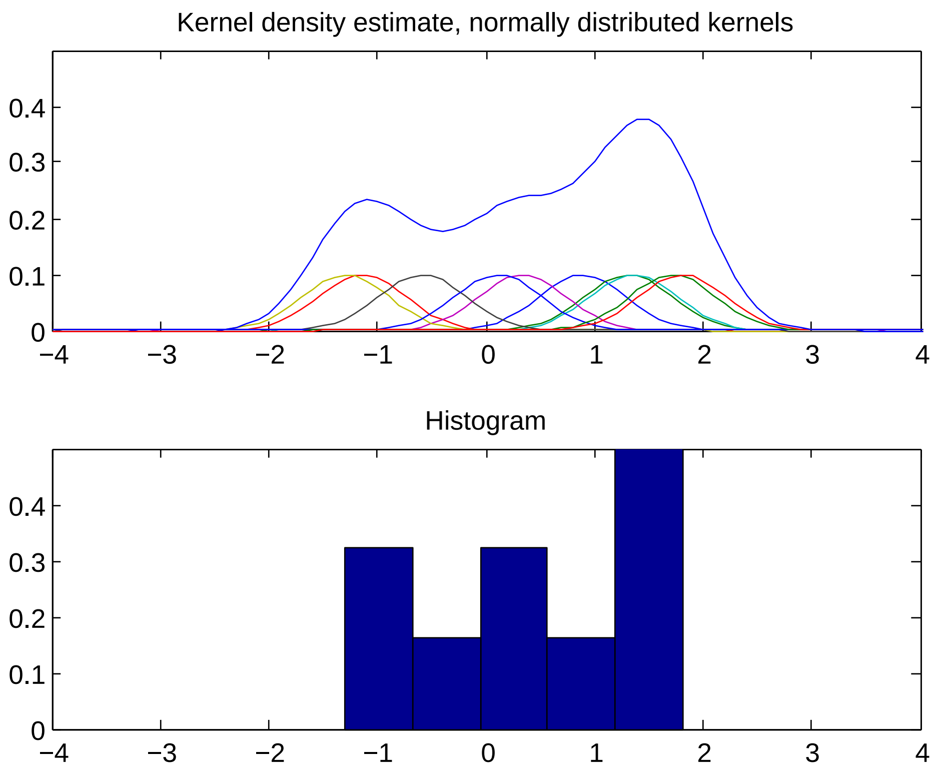

Figure 5: Kernel density estimate and corresponding histogram

The parameter $h \geq 0$ is called the *window width*, and it determines the width of the kernels. The choice of kernel is highly dependent on the density that is to be estimated, but common choices are normally distributed kernels K_n and Epanechnikov kernels K_e.

$$K_n(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \qquad \forall x \in \mathbb{R}$$

$$K_e(x) = \begin{cases} \frac{3}{4\sqrt{5}}(1 - \frac{1}{5}t^2) & -\sqrt{5} \leq x \leq \sqrt{5}, \\ 0 & \text{otherwise.} \end{cases}$$

For a normally distributed kernel, h corresponds to its standard deviation and X_i, the kernel position, corresponds to its expected value.

Optimally choosing h is a problem that has been confronted by many mathematicians. The main difficulty presented by the problem is that the optimal choice of h is dependent of the true density, which indeed is unknown. However, a general recommendation by Silverman (1998) is choosing

$$h = 0.9 \min(\sigma, \frac{IQ}{1.34}) n^{-\frac{1}{5}}$$

where n, σ and IQ are the sample size, standard deviation and interquantile range, respectively.

5.3.2 Application

This thesis will only take use of normally distributed kernels for density estimation due to the fact that binomial distributions can approximate normal distributions very well. For $X \sim B(n,p)$ with $np, n(1-p) \geq 5$ it can be shown that X approximately follows $N(np, np(1-p))$.

The reason why the above fact is useful when trying to discretisize $G(t)$ is best explained by a simple example which then can be extended to the actual problem setting.

Let $G(0) = k$. Assume that $G(1)$ would follow a normal distribution with mean 0 and standard deviation σ (which would be the case if $G(t)$ were an arithmetic brownian motion). That is, if the daily changes of G were to be independent and normally distributed, the distribution of $G(n)$, $n \in \mathbb{N}$ would also be normally distributed with mean k and standard deviation $\sqrt{n}\sigma$ since $G(n)$ could be written $G(0) + \sum_{k=1}^{n} G_k(1)$ where $G_k(1)$ denotes the k:th draw of the random variable $G(1)$. The above argument uses the the fact that a sum of normally distributed variables is again normally distributed.

$$G_k(1) \sim N(0, \sigma^2) \Rightarrow \sum_{k=1}^{n} G_k(1) \sim N(0, n\sigma^2)$$

Thus $G(n) \sim N(k, n\sigma^2)$. Now, if we construct an arithmetic binomial tree B with up-and down movements equal to σ having probabilities $p = 0.5$ and $1-p = 0.5$ for up-and down movements respectively, we can easily show that the n:th level of the tree has expected value k and variance $n\sigma^2$. Then, by the de Moivre-Laplace theorem, we have that the n:th level approximately follows $N(k, n\sigma^2)$ which means that the binomial tree B discretisizes G. The above arguments are discussed in greater detail in the next section.

In our case, the true density of $G(1)$ is estimated by a sum of m normally distributed kernels, and therefore, our goal is to generalize the above argument to construct m binomial trees discretisizing G.

To do that, we first, using the samples obtained from the Monte Carlo simulation, approximate the probability density function of $G(1)$ with a large number n of kernels, $n \geq 10000$ and an optimal window width h_{opt} to get the best possible density estimate. h_{opt} is chosen as suggested by Silverman.

$$\hat{g}_1(x) := \frac{1}{nh_{opt}} \sum_{i=1}^{n} K\left(\frac{x - G_i(1)}{h_{opt}}\right)$$

Trying to discretisize the density using n kernels, and thus n binomial trees, will be computationally too demanding, and thus we seek to approximate the the optimal estimate with an m-kernel estimate $5 \leq m \leq 100$.

Let $G^1(1), \ldots, G^n(1)$ denote the n draws of $G(1)$ obtained through Monte Carlo simulation, sorted by increasing value.

$$G^1(1) \leq G^2(1) \ldots \leq G^n(1)$$

The positions $\mu_1(1), \ldots, \mu_m(1)$ of the m kernels are chosen as

$$\mu_k(1) = \frac{m}{n} \sum_{i=(k-1)\frac{n}{m}+1}^{k\frac{n}{m}} G^i(1)$$

and the kernel window widths $\sigma_1, \ldots, \sigma_m$ are chosen as

$$\sigma_1 = \sigma_2 = \ldots = \sigma_m = h_t$$

Thus, the m-kernel estimate of the probability density function of $G(1)$ is defined as

$$\hat{g}_1^m(x) := \frac{1}{mh_t} \sum_{i=1}^{m} K\left(\frac{x - \mu_i(1)}{h_t}\right)$$

where h_t is chosen such that the integrated square error of $\hat{g}_1^m(x)$

$$ISE = \int_{-\infty}^{\infty} (\hat{g}_1(x) - \hat{g}_1^m(x))^2 dx$$

is minimized.

This is done numerically. Figure 6 shows the integrated square error as a function of h. Since $ISE(h)$ is a convex function there is a unique solution h_t to the minimization problem.

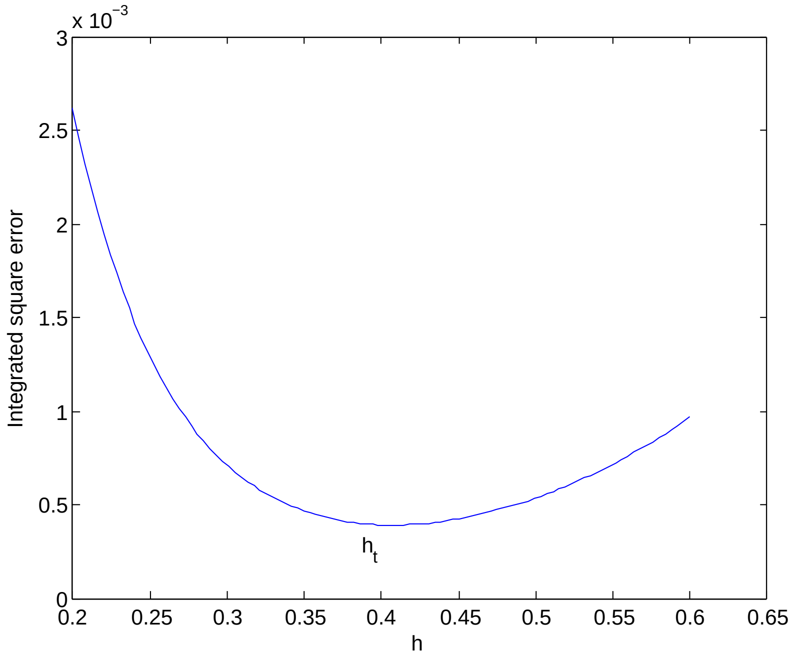

Figure 6: $ISE(h)$

Using the parameters $\mu_1(1), \ldots, \mu_m(1)$ and h_t obtained above, it is possible to construct m binomial trees discretisizing G. However, by studying $\mu_1(t), \ldots, \mu_m(t)$ where $\mu_k(t)$ is defined as

$$\mu_k(t) = \frac{m}{n} \sum_{i=(k-1)\frac{n}{m}+1}^{k\frac{n}{m}} G^i(t)$$

for $t \in 1, \ldots, T$ one will find that

$$\mu_k(i) \neq \mu_k(j) \qquad i, j \in 1, \ldots, T \qquad i \neq j \qquad k \in 1, \ldots, m$$

Indeed, figure 7 shows that the kernels drift away from their starting positions nonlinearly with time. The kernels seem to drift faster the further away from $G(0) = 10$ they start.

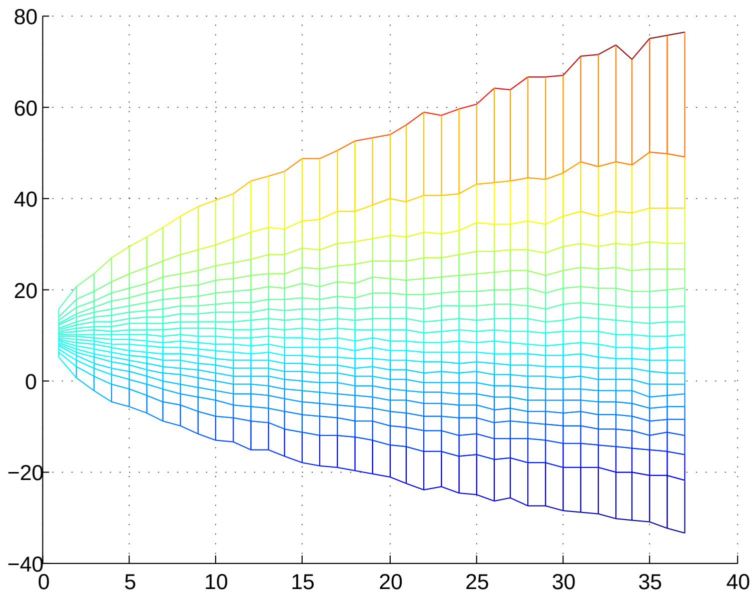

Figure 7: Kernel positions as function of time

To account for the changes in the kernel positions, we introduce linear drifts $d_1, \ldots, d_m$ to the m kernels, hence setting

$$\hat{\mu}_k(t) = \mu_k(1) + (t-1)d_k$$

If one were to use a different set of drifts for each time step, the resulting binomial trees would no longer recombine their branches, thus causing the number of branches to increase at the rate $m2^t$ which would result in extremely time consuming computations. Therefore, we are required to find one set of drifts $d_1, \ldots, d_m$ to approximate the average drifts of the kernels. This will be done by linear regression.

We are looking for a set of linear functions $y_k(x) = d_k x + l_k$ with parameters d_k and l_k chosen such that

$$F_k(d_k) = \sum_{x=1}^{T} \big(y_k(x) - \mu_k(x)\big)^2$$

is minimized for $k = 1, \ldots, m$.

Since we want $\mu_1(1), \ldots, \mu_m(1)$ to be the kernels' starting positions we set $l_k = \mu_k(1)$ and translate the x-coordinates by setting $x' = x - 1$ to form

$$F_{k'}(d_k) = \sum_{x'=0}^{T-1} \big(y_k(x') - \mu_k(x'+1)\big)^2 = \sum_{x'=0}^{T-1} \big(d_k x' + \mu_k(1) - \mu_k(x'+1)\big)^2$$

which is to be minimized for $k = 1, \dots, m$.

Seeing that $\{F_{k'}(d_k)\}$ is a set of convex functions, the solutions $d_1^*, \dots, d_m^*$ to the equations

$$\frac{\partial F_{x'}(d_k)}{\partial d_k} = 2\sum_{x'=0}^{T-1} \big(d_k x' + \mu_k(1) - \mu_k(x'+1)\big)x' = 0 \qquad k = 1. \dots, m$$

will minimize $F_{1'}(d_1), \dots, F_{m'}(d_k)$ respectively. Some simple algebra yields

$$d_k^* = \frac{\displaystyle\sum_{x'=1}^{T-1} \big(\mu_k(x'+1) - \mu_k(1)\big)x'}{\displaystyle\sum_{x'=1}^{T-1} x'^2}$$

which is the optimal linear kernel drift.

Note that

$$\sum_{x'=1}^{T-1}\sum_{k=1}^{m} \big(\mu_k(x'+1) - \mu_k(1)\big)x' = \sum_{x'=1}^{T-1}\Big(\underbrace{\sum_{k=1}^{m}\mu_k(x'+1)\,x'}_{\to mG(0)} - \underbrace{\sum_{k=1}^{m}\mu_k(1)\,x'}_{\to mG(0)}\Big) \to 0$$

as the number of simulations $n \to \infty$, which means that

$$\sum_{k=1}^{m} d_k^* \to 0 \qquad \text{as} \qquad n \to \infty$$

The above is the case since $G(t)$ is a martingale. The fact that

$$\sum_{k=1}^{m} \mu_k(\cdot) \to mG(0) \qquad \text{as} \qquad n \to \infty$$

follows directly from the definition of μ_k. In the next section the m binomial trees are constructed based on the parameters $\mu_1(1), \dots, \mu_m(1)$, $d_1^*, \dots, d_m^*$ and h_t.

5.4 Step three: Binomial trees

In the following sections, the proper parameters for the binomial trees are calculated.

5.4.1 Theory

m arithmetic binomial trees will be used to discretisize $G(t)$. We define $B_{i,t,k}$ to be the value of the i:th, $i \in 0, \ldots, t-1$ node of the t:th, $t \in 0, \ldots, T-1$ level in the k:th, $k \in 1, \ldots, m$ binomial tree. The numbering convention that is used for the nodes and levels is shown in figure 8. It holds that

$$B_{i,t,k} = B_{k,0} - t(D_k) + i(U_k + D_k) + td_k$$

where $B_{k,0}$ is the initial value of the k:th tree. U_k and D_k are the up- and down movements of the k:th binomial tree, respectively. d_k represents the drift of the k:th tree.

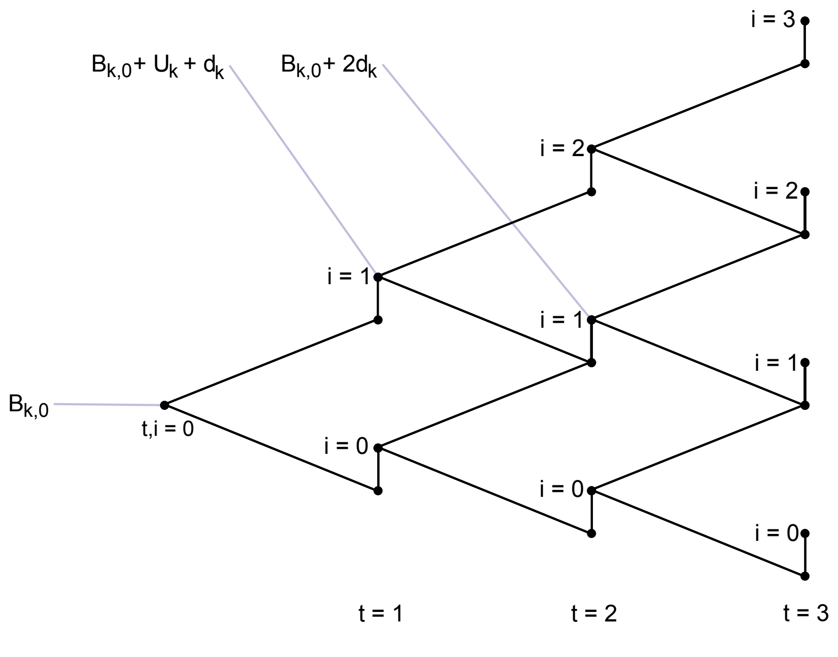

Figure 8: Arithmetic binomial tree with drift

5.4.2 Application

As discussed in the previous section, it is desired that the distribution of the process value at level t in tree k approximates a normal distribution with expected value $\mu_k(1) + td_k^*$ and variance th_t^2. To achieve that, U_k, D_k, d_k and $B_{k,0}$ need to be chosen in a proper way. The following calculations justify the choice $U_k = D_k = h_t$, $d_k = d_k^*$ and $B_{k,0} = \mu_k(1)$. The process value at level t in tree k is expressed by the following random variable

$$B(t,k) := B_{k,0} - tD_k + X(U_k + D_k) + td_k$$

where

$$X \sim Bin\big(t, p = 0.5\big)$$

By the de Moivre-Laplace theorem we have that

$$X \sim Y \text{ where } Y \sim N\big(tp, tp(1-p)\big)$$

asymptotically as $t \to \infty$. But for $t, tp \geq 5$ the above holds as a reasonable approximation. Therefore we approximate

$$B(t,k) \approx B_{k,0} - tD_k + Y(U_k + D_k) + td_k$$

Some simple algebra yields

$$B(t,k) \sim N\big(B_{k,0} + td_k + \frac{t}{2}(U_k + D_k) - tD_k, \frac{t}{4}(U_k + D_k)^2\big)$$

By setting $U_k = D_k = h_t$, $d_k = d_k^*$ and $B_{k,0} = \mu_k(1)$ we obtain the desired result.

$$B(t,k) \sim N\big(\mu_k(1) + td_k^*, th_t^2\big)$$

Hence,

$$B_{i,t,k} = \mu_k(1) - th_t + 2ih_t + td_k^*$$

which means that the discretization is done.

Note that

$$\frac{1}{m}\sum_{k=1}^{m} \mu_k(1) = \frac{1}{m}\sum_{k=1}^{m} B_{0,0,k} \to G(0)$$

and that

$$\sum_{k=1}^{m} d_k^* \to 0$$

as the number of simulations $n \to \infty$, which means that the discrete version of $G(t)$ also is a martingale.

The binomial trees allow us to calculate $PV_{0,f,0}$ which is to be compared with the simulated value of $PV_{0,f,0}$, but more importantly, they allow us to solve the optimization problem and hence calculate $PV_{0,f,c}$.

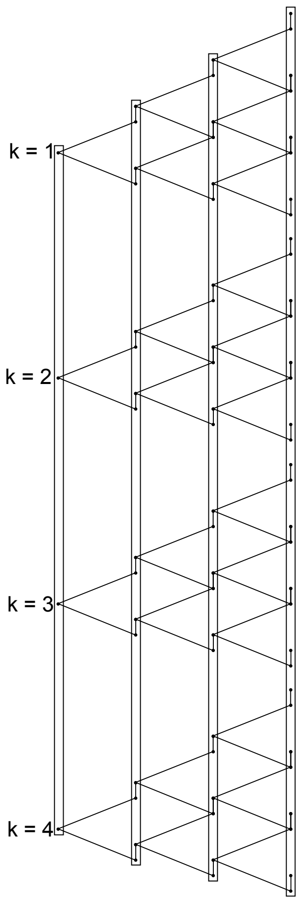

Figure 9: Calculating $\widehat{PV}_{0,f,0}$

We let $\widehat{PV}_{0,f,0}$ denote the value of $PV_{0,f,0}$ calculated using the discrete version of $G(t)$, while $PV_{0,f,0}$ denotes the true value simulated from $G(t)$.

We define

$$\widehat{PV}_{0,f,0} := \frac{1}{m} \sum_{k=1}^{m} \sum_{t=0}^{T-1} \sum_{i=0}^{t-1} \binom{t}{i} 0.5^t \max[B_{i,t,k}, 0] e^{-r_f \frac{t}{250}}$$

Figure 9 displays the way $\widehat{PV}_{0,f,0}$ is calculated. First all negative node values are replaced with zeros. Then each node value inside of each box is weighted with the probability of being in the state corresponding to the node value. Thereafter, for each box, the average of the value contribution of each tree is taken. We call that average value the box value. Finally the box values are summed.

The binomial trees can be used as a framework to solve the problem of finding $PV_{0,f,c}$. This will however not be done in this thesis, see Conclusions for further details.

6 Results

6.1 Paul Fackler's framework

In the following examples, the switching costs were set to 20. That is, switching the turbine on, or off, costs 20.

6.1.1 Geometric brownian motion

Using the model parameters $k = 1$, $\sigma_e = 0.05$, $\sigma_g = 0.02$, $S_{e,0} = 40$ and $S_{g,0} = 30$ for two geometric brownian motions we get approximate switching boundaries which are estimated by the following two linear functions

$$y_{\text{on}} = 1.0862x + 1.6178 \qquad y_{\text{off}} = 0.93609x - 1.4759 \qquad x \geq 0$$

After implementing these functions into a matlab program which performs a Monte Carlo simulation, the switching option value $1.8677 \cdot 10^3$ was obtained. This value is to be compared with value obtained when switching costs are set to zero, namely $1.9845 \cdot 10^3$. Thus, the option is worth roughly 6% more if switching can be done without a cost.

By performing the same procedure, but with volatilities σ_e and σ_g varied according to the table, the following results for $\frac{V(0)}{V(c)}$ were obtained.

$\sigma_e \backslash \sigma_g$	0.01	0.02	0.03
0.01	1.0303	1.0009	1.0105
0.02	1.1618	1.0013	1.0450
0.03	1.2144	1.1378	1.0078

6.1.2 Geometric Ornstein-Uhlenbeck processes

The following values for $\frac{V(0)}{V(c)}$ were obtained by setting $k = 1$, $\sigma_e = 0.02$, $\sigma_g = 0.02$, m_e and $m_g = 30$ and varying h_e and h_g according to the table.

$h_e \backslash h_g$	0.0001	0.0002	0.0003
0.0001	1.2439	1.3222	1.0729
0.0002	1.1113	1.2060	1.2500
0.0003	1.0840	1.1031	1.0836

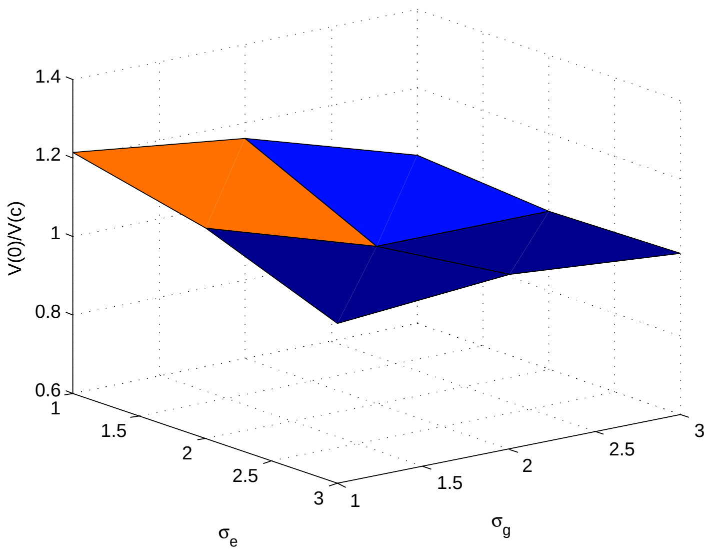

Figure 10: $\frac{V(0)}{V(c)}$ for price processes modeled as geometric brownian motions

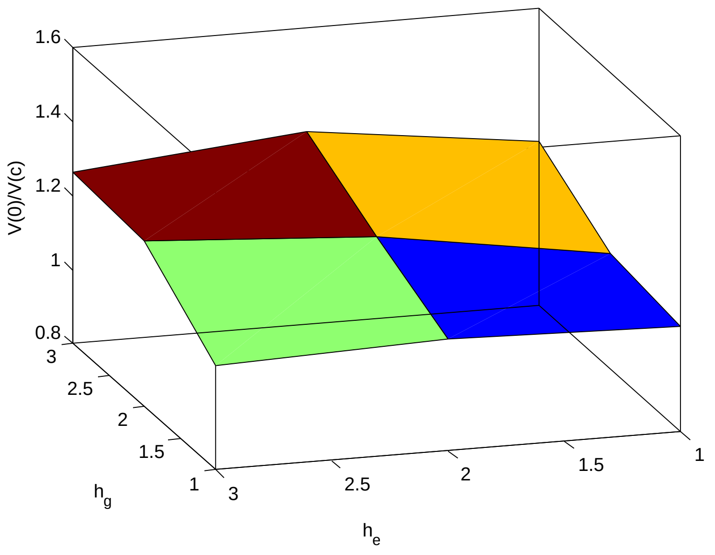

Figure 11: $\frac{V(0)}{V(c)}$ for price processes modeled as geometric Ornstein-Uhlenbeck processes

6.2 Binomial tree method

The following relative errors, calculated as

$$\frac{|\widehat{PV}_{0,f,0} - PV_{0,f,0}|}{PV_{0,f,0}}$$

were obtained using 20 kernels, 10000 simulations, $T = 365$, $r_f = 0$, $k = 1$, $S_{e,0} = 40$, $S_{g,0} = 30$ and by varying the volatilities σ_e and σ_g according to the table.

$\sigma_e \setminus \sigma_g$	0.01	0.02	0.03	0.04
0.01	0.01014	0.00002	0.00052	0.00341
0.02	0.00727	0.00520	0.00450	0.00527
0.03	0.00803	0.00913	0.00012	0.00721
0.04	0.00429	0.00020	0.00657	0.00098

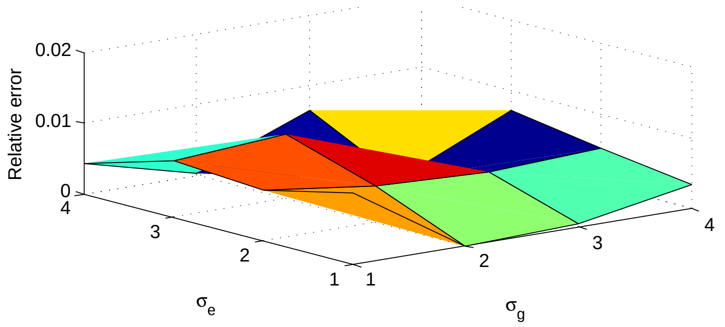

Figure 12: The relative error as a function of σ_e and σ_g

Using 10 kernels, $T = 365$, $r_f = 0$, $k = 1$, $S_{e,0} = 40$, $S_{g,0} = 30$, and fix values for $\sigma_e = \sigma_g = 0.02$ we consider the relative errors obtained when the number of simulations, n is varied.

n	Relative error
5000	0.01529
20000	0.01468
50000	0.01273
100000	0.01176

By setting $n = 5000$, $T = 365$, $r_f = 0$, $k = 1$, $S_{e,0} = 40$, $S_{g,0} = 30$, $\sigma_e = \sigma_g = 0.02$ and varying the number of kernels, the following relative errors were obtained.

#Kernels	Relative error
5	0.02197
20	0.00652
50	0.00308
100	0.00017

7 Conclusions

It is very clear that the level of difficulty associated with solving the power station problem increases dramatically if switching costs are present. Whenever switching can be done without a cost, the problem is easily solved by Monte Carlo methods, or, if the calculation time is an issue, by fast Fourier transform methods. The latter methods have a great advantage if complex stochastic dynamics such as stochastic volatility, stochastic correlation or jumps are added to the model.

Due to the fact that the problem is significantly more difficult to solve whenever switching costs are present, it can be very time saving to simply approximate that the switching costs are zero. This is of course reasonable whenever the switching costs are very low compared to the profit of the power station.

If, however, the dynamics of the energy price processes are such that switching often becomes a reasonable option, and if the costs associated with switching are high, the switching costs can not simply be neglected. In these cases the problem can be solved very accurately by using Fackler's numerical solution to the general solution conditions presented by Brekke and Oksendal. Fackler's numerical solution provides approximate switching boundaries which can be used to calculate a very exact option value by Monte Carlo simulation.

A bottleneck in Fackler's numerical solution is that a smoothing newton method must be used to solve the EVLCP which in some problem settings ends up running in an infinite loop. However, this problem can often be bypassed by changing the basis functions approximating the optimal value function, or by slightly changing the model parameters.

Furthermore, if it is impossible to express the approximate switching boundaries in closed form the Monte Carlo implementation can be tedious to perform.

As for the results obtained using Fackler's framework, it can be observed that the ratio $V(0)/V(c)$ decreases as the volatilities increase. This can be explained by the fact that the real option becomes more valuable when the price processes have high volatilities. Thus, both $V(0)$ and $V(c)$ increase, therefore causing the ratio $V(0)/V(c)$ to decrease. For geometric Ornstein-Uhlenbeck processes, the same effect can be observed as the rates of reversion are changed. Low rates of reversion does in some sense correspond to more volatile processes, and as can be seen in figure 11, $V(0)/V(c)$ decreases for smaller values of h_e, h_g.

Concerning the binomial tree method; It is clear that the number of kernels is the main factor influencing the relative error. With an increasing number of kernels, the relative error rapidly decreases. Naturally, if the number of simulations is increased, the approximation also gets better. However, increasing the number of simulations significantly increases the simulation time. Due to the fact that the generated realizations must be sorted, the simulation time increases as $n \log n$ under the assumption that an $n \log n$ sorting algorithm is used. More noteworthy is that the runtime hardly is changed if the number of kernels is increased. Using only $n = 5000$ simulations and 100 kernels yields a relative error of 0.017%, which is extremely low. Hence, a low number of simulations can be sufficient to obtain a very accurate discretization if the number of kernels is large enough. It also seems like the quality of the approximation is independant of the specific combination of volatilities that is chosen. However, dispite that fact that the discretization of G is very accurate, it is not possible to directly use the binomial trees to find $PV_{0,f,c}$ since they do not share nodes.

An idea worth exploring is to find an algorithm that merges the binomial trees together into one discrete structure. Figure 13 shows the principles of above idea; each cluster of nodes marked with blue circles are replaced by one node, and thus the binomial trees are merged into one discrete structure.

Exactly how the merging should be done in order to achieve optimal results could be a subject of further research.

Suppose however that such an algorithm would exist and that it has been used to merge the binomial trees into a discrete structure in which all nodes are connected. In that case, the discrete structure could be used to find $PV_{0,f,c}$ through an optimization method suggested by Copeland and Antikarov (2003).

One of the major drawbacks of the binomial tree method is that it to some extent fails to recognize that for many choices of the stochastic processes $P_e(t)$ and $P_g(t)$, the dynamics of $G(t+s)$ not only depends on $G(t)$ but rather on the exact values of $P_e(t)$ and $P_g(t)$. Looking at geometric brownian motions, the absolute change of $G(t)$ between two time points t and $t+s$ is expected to be much larger if the processes have high values at the time t.

For example, if $G(t) = 0$, one is only given the information that $P_e(t) = P_g(t)$, but nothing about the exact values of the processes. Therefore, it

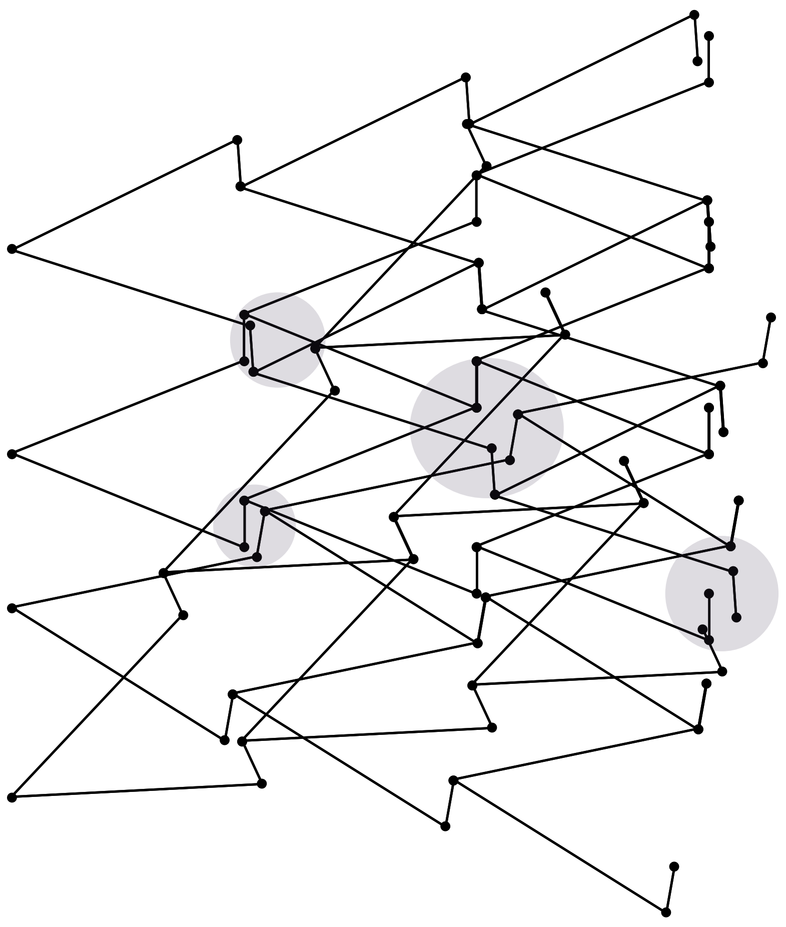

Figure 13: Merging algorithm

is essentially impossible to say something about the dynamics of $G(t+s)$ given only $G(t)$. However, it is sometimes essential to know the dynamics of $G(t+s)$ to be able to answer the question of whether the turbine should be powered down or not at a certain point in time.

At a first glance, the above might look like a very big problem. Yet, it is very hard to say something about the quality of the potential final results of the binomial tree method, since these would be very dependent of the algorithm used to merge the binomial trees.

Furthermore, if the binomial tree parameters can be estimated reasonably well using the data of a time series of energy prices, the binomial tree method can be used to give an extremely fast approximate option value. Note that no Monte Carlo simulations and no exact assumptions about the stochastic processes would have to be made. This use of the binomial tree method is currently a subject of research.

To summarize, the real option is very valuable under essentially any reasonable model assumptions. Under the model assumptions used in this thesis, the option does in many cases increase the total profit by more than 50%. Even if one assumes that the gas price has a higher expected value than the electricity price, the power station can be a profitable investment if the real option is considered. Only in rare cases in which the switching-and running costs are very high and the energy price volatilities very low does the power station become unprofitable. Generally, the option becomes more valuable if the energy price volatilities are high, and of course, if the gap between the expected values of the electricity and gas prices is large.

For the binomial tree method, the only source of error is the Monte Carlo method. However, it is easily verified that the error becomes very small as the number of simulations n becomes large. The absolute error of the Monte Carlo method decreases as $\frac{1}{\sqrt{n}}$. In the problem settings discussed in this thesis, the error becomes insignificant ($\leq 1\%$) if n is chosen larger than 10000.

The results produced using Fackler's framework have three sources of error; the newton method, the polynomial fitting of the switching boundaries and the Monte Carlo method. The newton method has quadratic convergence and the polynomial fitting can be done arbitrarily well if higher order polynomials are chosen. Therefore, the bottleneck is the Monte Carlo method. As discussed above however, the error can be kept very low if the number of simulations is chosen large enough.

As for the mathematical modeling, the models assumed in this thesis can, as discussed in previous sections, be extended with jump dynamics and stochastic correlation to more properly imitate reality, but should serve as a reasonable starting point of analysis.

References

[1] Copeland, T. & and Antikarov, V. (2003). *Real Options: A Practitioners Guide*, Cengage Learning, New York

[2] Silverman, B.W. (1998). *Density Estimation for Statistics and Data Analysis*, Chapman & Hall, Boca Raton

[3] Fackler, P. (2004) *Solving Optimal Switching Models*, http://sisla06.samsi.info/fmse/ci/switch.pdf : 31/7/2010

[4] Geman, H. (2005). *Commodities and Commodity derivatives: Modeling and Pricing for Agriculturals, Metals, and Energy*, John Wiley & Sons Ltd, Chichester

[5] Dempster, M.A.H. & Hong, S.S.G. (2000) *Spread Option Valuation and The Fast Fourier Transform*, http://www.jbs.cam.ac.uk/research/working_papers/2000/wp0026.pdf, 31/7/2010

[6] Brekke, K.A. & Oksendal, B. (1994) *Optimal Switching in an Economic Activity under Uncertainty*, SIAM Journal of Optimization , 32, pp 1021-1036

[7] Fackler, P. http://www4.ncsu.edu/ pfackler/ : 31/7/2010

[8] Taschini, L. & Urech, S. (2009) *The Real Option to Fuel Switch in the presence of Expected Windfall Profits under the EU ETS*, The Journal of Energy Markets, Vol. 3, No. 1, pp. 1-21

[9] Dixit, A.K. & Pindyck, R.S. (1994) *Investment under Uncertainty*, Princeton University Press, Princeton

[10] Brennan, M.J. & Trigeorgis, L. (2000) *Project Flexibility, Agency, and Competition: New Developments in the Theory and Application of Real Options*, Oxford University Press, New York

[11] Schwartz, E.S. & Trigeorgis, L. (2001) *Real Options and Investment under Uncertainty: Classical Readings and Recent Contributions*, MIT Press Books, The MIT Press, edition 1, volume 1, number 0262693186, June.

[12] Qi, H-D. & Liao, L-Z. (1999) *A Smoothing Newton Method for Extended Vertical Linear Complementarity Problems* Journal of Matrix Analysis and Applications, 21, pp 45-66

Appendices

A GBMISE.m

Contents of GBMISE.m which is used to calculate h_t:

```
clear all
clf
% number of timesteps
n = 365;
% number of realizations in the monte carlo simulation
m = 30000;

nkernels = 50;%number of kernels

%window width span
h_test_a = 0.1;
h_test_b = 0.6;

%number of timesteps per day
timestep = 1;

%electricity price drift per day
n1 = 0;

%gas price drift per day
n2 = 0;


%electricity price volatility (per day)
sg1 = 0.02;

%gas price volatility (per day)
sg2 = 0.02;

r = 0; %risk—free interest rate
% electricity price GBM
s1 = (1:n+1);
% gas price GBM
s2 = (1:n+1);

s1(1) =40;
s2(1) =30;

%perform monte carlo simulation
```

```
for (k = 1:m)

    s1(1) = 40;
    s2(1) = 30;

    N1 = randn(1,n);
    N2 = randn(1,n);

    N1 = sqrt(timestep)*N1;
    N2 = sqrt(timestep)*N2;
    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t1(1,i) = sum(N1(1,1:i-1),2);
    end
    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t2(1,i) = sum(N2(1,1:i-1),2);
    end

    % construct the electricity price GBM
    for (i = 2:length(s1))
        s1(i) = s1(1)*exp((n1-sg1^2/2)*i*timestep+sg1*t1(1,i));
    end
    % construct the gas price GBM
    for (i = 2:length(s2))
        s2(i) = s2(1)*exp((n2-sg2^2/2)*i*timestep+sg2*t2(1,i));
    end

    CF1(k) = s1(2)-s2(2);

    %calculate the PV without flexibility
    PVk0(k) = 0;

    for (i = 2:length(s1))
        PVk0(k) = PVk0(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
    end


    %calculate the PV with flexibility

    PVk0F(k) = 0;

    for (i = 2:length(s1))
        if ((s1(i)-s2(i)) > 0)
            PVk0F(k) = PVk0F(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
        end
    end
    PVk0F(k);
```

```
end

%PV without flex
PV0 = sum(PVk0)/m
%PV with flex
PV0F = sum(PVk0F)/m

%%%% For the manual estimation with #nkernels

%sort the contents of CF1

for (j = 2:length(CF1))
    for (i = 2:length(CF1)-j+2)
        if (CF1(i-1) > CF1(i))
            tmp = CF1(i-1);
            CF1(i-1) = CF1(i);
            CF1(i) = tmp;
        end
    end
end

for (j = 2:length(CF2))
    for (i = 2:length(CF2)-j+2)
        if (CF2(i-1) > CF2(i))
            tmp = CF2(i-1);
            CF2(i-1) = CF2(i);
            CF2(i) = tmp;
        end
    end
end

%creates intervals and assigns #nkernels kernels

dens = length(CF1)/nkernels;
for (i = 1:nkernels)
    kernEV(i) = CF1(dens/2+(i-1)*(dens));
end

%estimate the density with optimal bandwidth
[bw,den,xmesh] = kde(CF1,2^12,4,16);

%choose bandwidth for the #nkernels

%Grid for the density estimates
z = 4:((16-4)/(2^12-1)):16;

%--- ISE analysis

num = 100;
```

```
grid = h_test_a:(h_test_b-h_test_a)/(num-1):h_test_b;
for (j = 1:num)
    h_test = h_test_a+(j-1)*(h_test_b-h_test_a)/(num-1);

    for (i =1:length(kernEV)) %construct #nkernels kernels
        K(:,i) = 1/(sqrt(2*pi)*nkernels*h_test)
        *exp(-power((z-kernEV(i))/h_test,2)/2);
    end

    T=sum(K,2); %sum the kernels

    ISE(j) = sum(power((den-T),2)*(16-4)/(2^12-1));
end
%---

plot(grid,ISE)
```

B GBMSimOnly.M

Contents of GBMSimOnly.m which is used to perform Monte Carlo simulations and hence calculate $PV_{0,f,0}$:

```
clear all
clf
% number of timesteps
n = 365;
% number of realizations in the monte carlo simulation
m = 50000;

%number of timesteps per day
timestep = 1;

%electricity price drift per day
n1 = 0;
%gas price drift per day
n2 = 0;

%electricity price volatility (per day)
sg1 = 0.02;

%gas price volatility (per day)
sg2 = 0.02;

%risk-free interest rate
```

```
r = 0;

% electricity price GBM
s1 = (1:n+1);
% gas price GBM
s2 = (1:n+1);

s1(1) =40;
s2(1) =30;

%perform monte carlo simulation

for (k = 1:m)

    s1(1) = 40;
    s2(1) = 30;

    N1 = randn(1,n);
    N2 = randn(1,n);

    N1 = sqrt(timestep)*N1;
    N2 = sqrt(timestep)*N2;

    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t1(1,i) = sum(N1(1,1:i-1),2);
    end

    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t2(1,i) = sum(N2(1,1:i-1),2);
    end
    % construct the electricity price GBM
    for (i = 2:length(s1))
        s1(i) = s1(1)*exp((n1-sg1^2/2)*i*timestep+sg1*t1(1,i));
    end
    % construct the gas price GBM
    for (i = 2:length(s2))
        s2(i) = s2(1)*exp((n2-sg2^2/2)*i*timestep+sg2*t2(1,i));
    end

    %calculate the PV without flexibility

    PVk0(k) = 0;

    for (i = 2:length(s1))
        PVk0(k) = PVk0(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
    end
    %calculate the PV with flexibility

    PVk0F(k) = 0;

    for (i = 2:length(s1))
        if ((s1(i)-s2(i)) > 0)
            PVk0F(k) = PVk0F(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
        end
    end

    PVk0F(k);
end

%PV without flex
PV0 = sum(PVk0)/m
%PV with flex
PV0F = sum(PVk0F)/m

%%%%

%price processes
p1 = plot (s1);
hold on
p2 = plot (s2);
hold on
set(p1,'Color','red','LineWidth',1)
```

C GBMVec.m

Contents of GBMVec.m which is used to calculate $\{\mu_k\}$:

```
clear all
clf
% number of timesteps
n = 365;

% number of realizations in the monte carlo simulation
m = 20000;

%number of kernels
nkernels = 50;

%window width
h_test =0.2162;
```

```
%number of timesteps per day
timestep = 1;

%electricity price drift per day
n1 = 0;

%gas price drift per day
n2 = 0;

%electricity price volatility (per day)
sg1 = 0.01;

%gas price volatility (per day)
sg2 = 0.03;

%risk-free interest rate
r = 0;

% electricity price GBM
s1 = (1:n+1);

% gas price GBM
s2 = (1:n+1);

s1(1) =40;
s2(1) =30;

%perform monte carlo simulation
for (k = 1:m)

    s1(1) = 40;
    s2(1) = 30;

    N1 = randn(1,n);
    N2 = randn(1,n);

    N1 = sqrt(timestep)*N1;
    N2 = sqrt(timestep)*N2;

    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t1(1,i) = sum(N1(1,1:i-1),2);
    end

    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t2(1,i) = sum(N2(1,1:i-1),2);
    end
    % construct the electricity price GBM
    for (i = 2:length(s1))
        s1(i) = s1(1)*exp((n1-sg1^2/2)*i*timestep+sg1*t1(1,i));
    end

    % construct the gas price GBM
    for (i = 2:length(s2))
        s2(i) = s2(1)*exp((n2-sg2^2/2)*i*timestep+sg2*t2(1,i));
    end

    CF1(k) = s1(2)-s2(2);

    %calculate the PV without flexibility
    PVk0(k) = 0;

    for (i = 2:length(s1))
        PVk0(k) = PVk0(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
    end

    %calculate the PV with flexibility

    PVk0F(k) = 0;

    for (i = 2:length(s1))
        if ((s1(i)-s2(i)) > 0)
            PVk0F(k) = PVk0F(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
        end
    end
    PVk0F(k);
end

%PV without flex
PV0 = sum(PVk0)/m
%PV with flex
PV0F = sum(PVk0F)/m


%%%% For the manual estimation with #nkernels

%sort the contents of CF1

for (j = 2:length(CF1))
    for (i = 2:length(CF1)-j+2)
        if (CF1(i-1) > CF1(i))
            tmp = CF1(i-1);
            CF1(i-1) = CF1(i);
            CF1(i) = tmp;
        end
    end
end

for (j = 2:length(CF2))
    for (i = 2:length(CF2)−j+2)
        if (CF2(i−1) > CF2(i))
            tmp = CF2(i−1);
            CF2(i−1) = CF2(i);
            CF2(i) = tmp;
        end
    end
end

%creates intervals and assigns #nkernels kernels

dens = length(CF1)/nkernels;
for (i = 1:nkernels)
    %kernEV(i) = sum(CF1((1+(i−1)*dens):i*dens))/dens;
    kernEV(i) = CF1(dens/2+(i−1)*(dens));
end


%estimate the density with optimal bandwidth
[bw,den,xmesh] = kde(CF1,2^12,4,16);

%choose bandwidth for the #nkernels

%Grid for the density estimates
z = 4:((16−4)/(2^12−1)):16;

%construct #nkernels kernels

for (i =1:length(kernEV))
    K(:,i) = 1/(sqrt(2*pi)*nkernels*h_test)
    *exp(−power((z−kernEV(i))/h_test,2)/2);
end

%sum the kernels
T=sum(K,2);

%−−− ISE analysis

%ISE = sum(power((den−T),2)*(16−4)/(2^12−1));

%−−−

%plot kde estimation
kdeplot = plot(xmesh,den);
set(kdeplot,'Color','red','LineWidth',1)
```

```
hold on

%Plot n-Kernel estimation
plot(z,T);
hold on
%---


%{
% single kernels
plot(z,K);
hold on
%}
```

D GBMVecKernM.m

Contents of GBMVecKernM.m which gathers data through Monte Carlo simulation. The data is used to calculate $\{d_k^*\}$:

```
clear all
clf
% number of timesteps
n = 365;
% number of realizations in the monte carlo simulation
m = 20000;

%number of kernels
nkernels = 50;

%window width
h_test =0.3;

%for the normal distribution
EVn = 10;
sgn = 1.65;

%number of timesteps per day
timestep = 1;

%electricity price drift per day
n1 = 0;

%gas price drift per day
n2 = 0;
```

```
%electricity price volatility (per day)
sg1 = 0.02;
%gas price volatility (per day)
sg2 = 0.02;

r = 0; %risk-free interest rate

% electricity price GBM
s1 = (1:n+1);
% gas price GBM
s2 = (1:n+1);

s1(1) =40;
s2(1) =30;

%perform monte carlo simulation
for (kn = 5:10:365)
    for (k = 1:m)

        s1(1) = 40;
        s2(1) = 30;

        N1 = randn(1,n);
        N2 = randn(1,n);

        N1 = sqrt(timestep)*N1;
        N2 = sqrt(timestep)*N2;
    % construct two regular brownian motions for the GBM
        for (i = 2:n+1)
            t1(1,i) = sum(N1(1,1:i-1),2);
        end
    % construct two regular brownian motions for the GBM
        for (i = 2:n+1)
            t2(1,i) = sum(N2(1,1:i-1),2);
        end
    % construct the electricity price GBM
        for (i = 2:length(s1))
            s1(i) = s1(1)*exp((n1-sg1^2/2)*i*timestep+sg1*t1(1,i));
        end
    % construct the gas price GBM
        for (i = 2:length(s2))
            s2(i) = s2(1)*exp((n2-sg2^2/2)*i*timestep+sg2*t2(1,i));
        end

        %
        CF1(k) = s1(kn)-s2(kn);
        % --
    end

    %%%% For the manual estimation with #nkernels

    %sort the contents of CF1

    CF1 = sort(CF1);

        %creates intervals and assigns #nkernels kernels

        dens = length(CF1)/nkernels;
        for (i = 1:nkernels)
            kernEV((kn+5)/10,i) = sum(CF1((1+(i−1)*dens):i*dens))/dens;
            %kernEV((kn+5)/10,i) = CF1(dens/2+(i−1)*(dens));
        end
end
```

E KernAnalysis.m

Contents of KernAnalysis.m which is used to calculate $\{d_k^*\}$:

```
load('20kKE2−250.mat');

KE = kernEV;

load('mu(2)20k2−250.mat');

mesh(KE);


i = (1:37);
j = (i−1)*10+3;

for(st = 1:50)
    drifts(st) = (KE((1:37),st)'*j(1:37)'
    −kernEV(st)*sum(j(1:37)))/(j(1:37)*j(1:37)')
end
```

F CBMultiKDrO.m

Contents of CBMultiKDrO.m which is used to construct binomial trees and to calculate $\widehat{PV}_{0,f,0}$.

```
warning('off','MATLAB:nchoosek:LargeCoefficient')
T = 365;

load('mu(2)20k2-250.mat');
load('drifts20k2-250.mat');
drifts = drifts - sum(drifts)/length(drifts);
B = [1:(T*(T+1)/2)];
u=0.2162;
d=0.2162;
drift = drifts;
p = 0.5;
S = kernEV;
EV_Flex = 0;
EV = 0;
for (nk = 1:length(S))
    %Construct the tree, Tree(i,k) <=> B((i-1)i/2+k)
    for (i = 1:T)
        for (k = 1:i)
            B((i-1)*i/2+k) = S(nk)+drift(nk)*(i-1)-(i-1)*d+(k-1)*(u+d);
            %calc bin coefs
            binc(i,k) = nchoosek(i-1,k-1)*p^(k-1)*(1-p)^(i-k);
        end
    end
    %replace negative entries
    for (i = 1:T*(T+1)/2)
       if (B(i) < 0)
           C(i) = 0;
           %C(i) = -B(i);
       else
           C(i) = B(i);
           %C(i) = 0;
       end
    end
    %calculate EV of cash flows
    %with flex
    tmp = 0;
    for (i = 1:T)
        for (k = 1:i)
            tmp = tmp + (1/length(S))*binc(i,k)*C((i-1)*i/2+k);
            %tmp = tmp + binc(i,k)*C((i-1)*i/2+k);
        end
        EV_Flex = EV_Flex + tmp;
        tmp = 0;
    end
    %without flex
    %{
    tmp = 0;
    for (i = 1:T)
        for (k = 1:i)
            tmp = tmp + (1/length(S))*nchoosek(i-1,k-1)
        *p^(k-1)*(1-p)^(i-k)*B((i-1)*i/2+k);
        end
        EV = EV + tmp;
        tmp = 0;
    end
    %}
    nk
end
EV
EV_Flex
```

G GBMSimOnlySB.m

Contents of GBMSimOnlySB.m which is used to calculate the flexible project value when switching boundaries are known.

```
clear all
clf
% number of timesteps
n = 365;
% number of realizations in the monte carlo simulation
m = 10000;

%switching boundaries

a_on = 1.0862;
b_on = 0;
c_on = 1.6178;

a_off = 0.93609;
b_off = 0;
c_off = -1.4759;

scost_on = 20;
scost_off = 20;
```

```
%number of timesteps per day
timestep = 1;

%electricity price drift per day
n1 = 0;

%gas price drift per day
n2 = 0;


%electricity price volatility (per day)
sg1 = 0.05;
%gas price volatility (per day)
sg2 = 0.02;

%risk-free interest rate
r = 0;

% electricity price UOP
s1 = (1:n+1);

% gas price UOP
s2 = (1:n+1);

s1(1) =40;
s2(1) =30;

absscost = 0;

%perform monte carlo simulation
for (k = 1:m)

    running = 1;
    scost_tot = 0;

    s1(1) = 40;
    s2(1) = 30;

    N1 = randn(1,n);
    N2 = randn(1,n);

    N1 = sqrt(timestep)*N1;
    N2 = sqrt(timestep)*N2;

    % construct two regular brownian motions for the GBM
    for (i = 2:n+1)
        t1(1,i) = sum(N1(1,1:i-1),2);
    end
```

```
% construct two regular brownian motions for the GBM
for (i = 2:n+1)
    t2(1,i) = sum(N2(1,1:i-1),2);
end

% construct the electricity price GMB
for (i = 2:length(s1))
    s1(i) = s1(1)*exp((n1-sg1^2/2)*i*timestep+sg1*t1(1,i));
end

% construct the gas price GBM
for (i = 2:length(s2))
    s2(i) = s2(1)*exp((n2-sg2^2/2)*i*timestep+sg2*t2(1,i));
end

%calculate the PV without flexibility
PVk0(k) = 0;

for (i = 2:length(s1))
    PVk0(k) = PVk0(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
end

%calculate the PV with flexibility
t_off = zeros(1,length(s1));
t_on = zeros(1,length(s1));

PVk0F(k) = 0;

for (i = 2:length(s1))
    if (running == 0 && s1(i) ≥ (a_on*s2(i)+b_on*s2(i)^2+c_on))
       running = 1;
       t_on(i) = 1;
       scost_tot = scost_tot + scost_on;
    end

    if (running == 1 && s1(i) ≤ (a_off*s2(i)+b_off*s2(i)^2+c_off))
       running = 0;
       t_off(i) = 1;
       scost_tot = scost_tot + scost_off;
    end

    if (running == 1)
        PVk0F(k) = PVk0F(k) + (s1(i)-s2(i))/(1+r)^((i-1)/360);
    end
end

PVk0F(k) = PVk0F(k) - scost_tot;
absscost = absscost + scost_tot;
scost_tot = 0;
```

```
end

%PV without flex
PV0 = sum(PVk0)/m
%PV with flex
PV0F = sum(PVk0F)/m
```

H osGBM.m

Contents of osGBM.m which uses Fackler's framework to find approximate switching boundaries for price processes modeled as geometric brownian motions.

```
mu=0.01;
sigma=0.01;
rho=0.04;
h=2;
k=1;
C12=20;
C21=20;

clear model
model.func='turbine';
model.params={mu,sigma,rho,h,k};
model.C=[0 C12;C21 0];

% solve as a two dimensional problem
fspace=fundefn('spli',[41 41],[0 0],[120 100]);
[cv,snodes,x]=ossolve(model,fspace);
z=getboundaries(snodes,x);
z12=z{1,2};
z21=z{2,1};
z12(:,2)=-z12(:,2); z12=sortrows(z12,[1 2]);z12(:,2)=-z12(:,2);
z21(:,2)=-z21(:,2); z21=sortrows(z21,[1 2]);z21(:,2)=-z21(:,2);

figure(1);
plot(z12(:,1),z12(:,2),'-r',z21(:,1),z21(:,2),'-k')
hold on
x = [0:0.1:120];
plot(x,x,'--')
title('Approximate Switching Boundaries')
xlabel('kP_g')
```

```
ylabel('P_e')

xlim([0 120])
ylim([0 120])

figure(1)
```

I osGOU.m

Contents of osGOU.m which uses Fackler's framework to find approximate switching boundaries for price processes modeled as geometric Ornstein-Uhlenbeck processes.

```
mu=0.01;
sigma=0.1;
rho=0.04;
h=30;
k=0.01;
C12=20;
C21=0;

clear model
model.func='turbinegou';
model.params={mu,sigma,rho,h,k};
model.C=[0 C12;C21 0];

% solve as a two dimensional problem
fspace=fundefn('lin',[20 15],[0 0],[70 70]);
[cv,snodes,x]=ossolve(model,fspace);
z=getboundaries(snodes,x);
z12=z{1,2};
z21=z{2,1};
z12(:,2)=-z12(:,2); z12=sortrows(z12,[1 2]);z12(:,2)=-z12(:,2);
z21(:,2)=-z21(:,2); z21=sortrows(z21,[1 2]);z21(:,2)=-z21(:,2);

figure(1);
plot(z12(:,1),z12(:,2),'-r',z21(:,1),z21(:,2),'-k')
hold on
x = 0:0.01:120;
plot(x,x,'--')
title('Approximate Switching Boundaries')
xlabel('kP_g')
ylabel('P_e')
```

```
xlim([0 120])
ylim([0 120])


figure(1)
```

J turbine.m

Contents of turbine.m which defines the geometric brownian motion model.

```
function out=turbine(flag,s,x,mu,sigma,rho,h,k)
  switch flag
    case 'f'
      out=(s(:,2)-s(:,1)).*(x==2);
    case 'mu'
      out=[zeros(size(s,1),1) zeros(size(s,1),1)];
      %out=[mu*s(:,1) mu*s(:,2)];
      %
    case 'sigma'
      out=[2*sigma*s(:,1) zeros(size(s,1),2) 5*sigma*s(:,2)];
      %zeros(size(s,1),3)
    case 'rho'
      out=rho;
    case 'c'
      out=zeros(size(s,1),2);
      if x==1, out(:,2)=20; else, out(:,1)=20; end
  end
```

K turbineGOU.m

Contents of turbineGOU.m which defines the geometric Ornstein-Uhlenbeck model.

```
function out=turbinegou(flag,s,x,mu,sigma,rho,h,k)
  switch flag
    case 'f'
      out=(s(:,2)-s(:,1)).*(x==2);
    case 'mu'
      out=[k*(h-s(:,1)).*s(:,1) k*((h+10)-s(:,2)).*s(:,2)];
      %out=[mu*s(:,1) mu*s(:,2)];
      %
    case 'sigma'
      out=[5*sigma*s(:,1) zeros(size(s,1),2) 2*sigma*s(:,2)];
      %zeros(size(s,1),3)
    case 'rho'
      out=rho;
    case 'c'
      out=zeros(size(s,1),2);
      if x==1, out(:,2)=5; else, out(:,1)=2; end
  end
```