Numerical Methods for Pricing Swing Options in the Electricity Market

Master’s Thesis in Financial Mathematics

Matilda Guo, Maria Lapenkova
Numerical Methods for Pricing Swing Options in the Electricity Market

Matilda Guo, Maria Lapenkova

Halmstad University
Project Report IDE1025

Master’s thesis in Financial Mathematics, 15 ECTS credits
Supervisor: Prof. Dr. Matthias Ehrhardt
Examiner: Prof. Dr. Ljudmila A. Bordag
External referees: Prof. Dr. Mikhail Babich

November 28, 2010

Department of Mathematics, Physics and Electrical Engineering
School of Information Science, Computer and Electrical Engineering
Halmstad University
Preface

The energy market is relatively new, compared to the other financial markets, sparking an interest in finding out how derivatives work in this particular industry. Working on numerical methods also gave rise to an opportunity to hone programming skills as well as gain the experience of using actual market information in order to solve actual financial problems, bridging the gap between theory and practice, the bane of applied academics.

In order to proceed with this notion, the following have given much of their time and insights, in aid of this thesis.

First and foremost, I would like to express my utmost gratitude and appreciation for the guidance our thesis supervisor, Prof. Dr. Matthias Ehrhardt, has shown us and for all the time he has spent on the problems and questions we had throughout our research. He has been a most supportive and encouraging supervisor, giving us the freedom to work with our notions, yet always assisting us with great enthusiasm.

I would also like to thank my partner, Maria Lapenkova, for her enthusiasm and complementing talents. She has been a great pleasure to work with. I am also indebted to the following people. My friend, Ian Soon, for painstakingly proof-reading this thesis together with me and my coursemates Marek Uhliarik and Dmitry Shcherbakov for their guidance and assistance whenever we had difficulty with our programming. Our course Professors, Prof. Dr. Ljudmila A. Bordag and Prof. Dr. Mikhail Nechaev, for their interest and suggestions to better shape the direction of our thesis.

I could not have accomplished this much if not for the support from my all loved ones in Singapore and Korea. Thank you, for all your patience, confidence and faith, in all that I have chosen to undertake.
Last but not least, I would like to thank God for all that He has blessed me with and given unto me. To God be the glory, the best is yet to be.

Matilda Guo

First and foremost, I would like to thank our supervisor, who suggested to us to work on this topic. He gave us an excellent opportunity to apply our knowledge to solve a very important problem. During the period of our work, he tirelessly helped and advised us. I am very glad that with his help, I discovered a very interesting topic which I would like to develop further. I would like to thank him, especially for the time that he dedicated to us in Wuppertal. It really was very interesting and intensive work.

I would also to thank my partner Matilda Guo for all the time that we worked on our master thesis. It was very nice and easy to work with her.

Just a special thanks to Dmitry Shcherbakov, Sylwia Szwaczkiewicz, Marek Uhliarik and ChaSing Ngow for their help and support during our education in Sweden.

Very much I would like to thank our professors for their help and interest to our thesis. Special thanks to Michael Nechaev and the director of our program, Ludmilla Bordag.

And, of course, I would like to thank my parents for their love and constant support.

Maria Lapenkova
Abstract

Since the liberalisation of the energy market in Europe in the early 1990s, much opportunity to trade electricity as a commodity has arisen. One significant consequence of this movement is that market prices have become more volatile instead of its tradition constant rate of supply. Spot price markets have also been introduced, affecting the demand of electricity as companies now have the option to not only produce their own supply but also purchase this commodity from the market. Following the liberalisation of the energy market, hence creating a greater demand for trading of electricity and other types of energy, various types of options related to the sales, storage and transmission of electricity have consequently been introduced.

Particularly, swing options are popular in the electricity market. As we know, swing-type derivatives are given in various forms and are mainly traded as over-the-counter (OTC) contracts at energy exchanges. These options offer flexibility with respect to timing and quantity.

Traditionally, the Geometric Brownian Motion (GBM) model is a very popular and standard approach for modelling the risk neutral price dynamics of underlyings. However, a limitation of this model is that it has very few degrees of freedom, as it does not capture the complex behaviour of electricity prices. In short the GBM model is inefficient in the pricing of options involving electricity. Other models have subsequently been used to bridge this inadequacy, e.g. spot price models, futures price models, etc.

To model risk-neutral commodity prices, there are basically two different methodologies, namely spot and futures or so-called term structure models. As swing options are usually written on spot prices, by which we mean the current price at which a particular commodity can be bought or sold at a specified time and place, it is important for us to examine these models in order to more accurately inculcate their effect on the pricing of swing options.
Monte Carlo simulation is also a widely used approach for the pricing of swing options in the electricity market. Theoretically, Monte Carlo valuation relies on risk neutral valuation and the technique used is to simulate as many (random) price paths of the underlying(s) as possible, and then to average the calculated payoff for each path, discounted to today’s prices, giving the value of the desired derivative. Monte Carlo methods are particularly useful in the valuation of derivatives with multiple sources of uncertainty or complicated features, like our electricity swing options in question. However, they are generally too slow to be considered a competitive form of valuation, if any analytical techniques of valuation exist. In other words, the Monte Carlo approach is, in a sense, a method of last resort.

In this thesis, we aim to examine a numerical method involved in the pricing of swing options in the electricity market. We will consider an existing and widely accepted electricity price process model, use the finite volume method to formulate a numerical scheme in order to calibrate the prices of swing options and make a comparison with numerical solutions obtained using the theta-scheme. Further contributions of this thesis include a comparison of results and also a brief discussion of other possible methods.
# Contents

1 **Introduction**  
1.1 Background of the Electricity Market  
1.2 Fundamentals of Swing Options  
  1.2.1 An Example of a Swing Contract  
  1.2.2 Definition of a Swing Contract  

2 **The Least-Squares Monte Carlo Method**  
2.1 The LSM Algorithm  

3 **Fundamentals of Finite Difference Methods**  
3.1 The Crank-Nicolson Method  
3.2 The Theta Method  

4 **An Electricity Price Process**  
4.1 A Partial Differential Equation  

5 **Numerical Pricing with the Theta Scheme**  
5.1 Applying a Finite Difference Approach  
5.2 The Boundary and Initial Conditions  

6 **The Finite Volume Method**  
6.1 Applying the Finite Volume Method to the Pricing of Swing Options  

7 **Results**  
7.1 Theoretical Results  
7.2 Simulation Results  
  7.2.1 Finite Difference Scheme  
  7.2.2 The Finite Volume Scheme  

8 **Conclusion**
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Notations</td>
<td>55</td>
</tr>
<tr>
<td>Bibliography</td>
<td>57</td>
</tr>
<tr>
<td>Appendix I</td>
<td>61</td>
</tr>
<tr>
<td>Appendix II</td>
<td>65</td>
</tr>
<tr>
<td>Appendix III</td>
<td>67</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

In order to begin our examination of the energy market, we first have to take a brief look at the background and features of this particular market we are interested in, as well as the fundamentals of the type of option, i.e. the swing option, we are looking to price.

To examine the various methods used for pricing energy swing options so far, we will look at what has been done using the Least-Squares Monte Carlo method in Chapter 2. A brief version of a possible algorithm used will also be included in this chapter. As we are looking to price electricity swing options using numerical methods, the theory of Finite Difference approaches, in particular the Crank-Nicolson and Theta methods, will be discussed in Chapter 3.

In Chapter 4, we will talk about our choice of the price process used, and how it can be applied to the partial differential equation for the pricing of swing options. This leads us to Chapter 5 and Chapter 6, which will be devoted to the two numerical schemes we are investigating, i.e. the Theta scheme, as well as the Finite Volume scheme, when applied to the pricing of electricity swing options.

Lastly, Chapter 7 and Chapter 8 wraps up our paper with the results obtained from our numerical calibration will be discussed and compared with presently available results.
Chapter 1. Introduction

1.1 Background of the Electricity Market

With the liberalisation and the introduction of exchanges, the objectives of power generation industry have changed. In the traditional environment, firms provided energy for customers at a fixed price and the main consequence of the power market liberalisation is that the commodity now has a volatile price instead of a fixed rate.

There are many similarities and differences between the electricity market and the derivative market. Due to these differences between the electricity market and other financial markets, classical financial theories cannot be applied directly in the former market. Instead, modifications have to be made before they can be adapted. The introduction of competitive structures in the power industry is particularly challenging. The main difficulties are the physical characteristics of electrical power and the most essential factors include the following:

- Technology currently does not allow electricity, or electrical energy to be stored efficiently. It is essentially impossible to store the required electricity consumption of a large-sized factory, much less an entire nation. Electricity is therefore deemed as non-storable, thus a number of extensive consequences have to be considered. For example, the spot price is very sensitive to changes in the demand and supply of electricity. This implies high short-term volatility and the possible occurrence of spikes (which are comparatively large upward or downward movements of a price or value level in a short period).

- Unlike having a fixed number of shares over time in the share market, electricity can be produced, consumed and (hypothetically) stored. Based on this feature and microeconomic considerations, the price of electricity is expected to revert to production cost in the long term. This means that there is a mean-reverting effect for the spot price of electricity, i.e. volatility decreases in the long term. In other words, there exists a long-term equilibrium, otherwise known as the fair price, which is comparatively much less volatile than the spot price.

- Since hedging derivative contracts with the underlying asset or commodity requires storability, it therefore cannot be applied to electricity derivatives. In other words, the electricity market is incomplete. This means the existence of the risk neutral probability measure $\mathbb{Q}$ is not unique.
Transportation of electricity is very costly and access to power grids is essential for trading contracts that include physical delivery.

The generation and supply of electrical energy is determined by production companies. Therefore, decisions of these companies consider also their own internal portfolios and not only market situation and demand.

These features differentiate electricity from all other commodity markets. It is these uncertainties that necessitate an efficient management of production facilities and also the existence of financial contracts. Therefore energy derivatives are increasingly important instruments. Most popular in the electricity market is the swing option as this type of derivative contract allows flexibility in delivery with respect to both the timing and quantity of energy delivered.

1.2 Fundamentals of Swing Options

In general, an option is a financial derivative that represents a contract sold by one party (option writer) to another party (option holder). The contract offers the buyer the right, but not the obligation, to buy (call) or sell (put) a security or other financial asset at an agreed-upon price (the strike price) during a certain period of time or on a specific date (exercise date). In other words, call options give the holder the option to buy at a certain price, so the buyer would want the stock to go up, while put options give the holder the option to sell at a certain price, so the buyer would want the stock to go down.

There are a variety of options available in the market and they are broadly categorised as vanilla options or exotic options. A vanilla option is a normal option with no special or unusual features, i.e. a plain run-of-the-mill option, with a standard expiry and strike price. The two option styles are the European option and the American option, where the former is an option that can only be exercised at the end of its life, i.e. at its maturity, whilst the latter can be exercised anytime during its life. It is important to note that the names of the option styles have nothing to do with the geographical location at which they are traded.

An exotic option, on the other hand, differs from common American or Eu-
European options in terms of the underlying asset or the calculation of how or when the investor receives a certain payoff. These options are more complex than options that trade on an exchange, and generally traded over the counter (OTC). For example, chooser options, barrier options, Asian options, digital options and compound options are different types of exotic options, among others. In particular, path-dependent options are exotic options that are valued according to pre-determined price requirements for their respective underlying asset or commodity, such that the respective payoffs associated with these options are determined by the path of the underlying asset’s price.

Swing type derivatives are a broad class of path-dependent options that allow the holder to exercise a certain right multiple times over a specified period but only one at a time or per time interval. These options, mainly traded as OTC contracts at energy exchanges and are typically written on spot prices, offer flexibility with respect to timing and quantity, and can be seen as insurance for the option holder against excessive rises in electricity prices. In other words, the specifications of a swing contract typically include the time period, the total amount of energy that can be traded and the price per unit of the commodity, with the right to change (or swing, hence giving the option its name) the periodic fixed amount delivered to a new quantity with the restriction that this volume is kept between some pre-specified boundaries, and these (limited numbers of) swings have to be exercised at some specified time.

While swing contracts have been around for much longer than academic papers on their pricing methods have shown, they initially were solely agreements between producers and suppliers made with the purpose of providing the market suppliers with a bit of flexibility around the predetermined average commodity delivery requirement. In this way suppliers could, if only partially, hedge their exposure to volume risk arising from the complex patterns of consumption or demand and also incorporate the consideration of limited storage capacity. The deregulation of the energy market and recent soaring of commodity prices have greatly increased the need for derivatives managing the risk associated with spot prices, which now represent the new demand and supply situation.

Swing options can be seen as a portfolio of American options or a generalisation of Bermudan options (a type of option that can only be exercised on predetermined dates, usually every month), but the existence of a refractor period (during which no further right can be exercised and which is implemented to avoid the natural optimal strategy to exercise all the rights simultaneously) makes the pricing of these options tremendously less trivial.
Examining the two extreme possibilities of swing options, i.e. the one-swing (a single exercise right) and the full-swing (as many exercise rights as the overall contract period divided by the refracting period), which reduces the swing option to an ordinary American option or a combination of European options respectively, will exemplify the complexity of pricing methodology due to the various forms of swing options, and often the existence of a penalty function.

Despite the diversity of swing contract specifications, the difficulties one faces in pricing swing options are similar to those of valuing American type derivatives, i.e. the option holder has, at any time, the right but not the obligation to exercise the contract. Particularly, one also cannot assume that the holder will always exercise the contract in an optimal way to maximise expected profit but also according to their internal energy demands. Thus, it is tremendously difficult to derive explicit valuation formulae for swing contracts, hence numerical schemes are needed instead.

Mathematically, in spite of the variety of options, there are two preferred approaches in pricing swing contracts, namely Monte-Carlo modelling by the stochastic control theory and simulation as a multiple stopping time problem.

The goal of the former approach is to find the optimal consumption process for the underlying commodity and to use dynamic programming principles and techniques to compute numerical solutions. This approach is flexible and its main advantage is that it can be easily adapted to any stochastic model of its underlying. Notable techniques for this approach include the extension of the binomial/trinomial method to the forest of trees method [18], modelling as an impulse control problem and ideas of duality theory to derive upper and lower bounds of option prices. The latter approach uses the theory of the Snell envelopes to determine the optimal boundaries and prove the existence of a sequence of optimal exercise strategies.

1.2.1 An Example of a Swing Contract

Party ABC agrees to deliver 50 000 MW of power per day to XYZ for the month of September (the nominated amount). We assume that the price to be paid for power is fixed at $45/MWh. Party XYZ has the option to change this nominated daily amount for a limited number of times. A change in the
quantity may be necessary due to fluctuations in demand, weather changes, 
particular spot price expectations, high costs, amongst other factors.

At the start of each day, party XYZ has the right (but not the obligation) to 
decrease the consumption to 30 000 MW for that day alone, and at the 
same fixed price $45/MWh, but he may exercise this right for a maximum 
of 10 times over the entire month of September.

Additionally, party XYZ is required to purchase at least 900 000 MW (the 
minimal amount) in total over the month of September. In which case, if the 
requirement is not met, party XYZ must pay some form of penalty at the 
expiration of the contract. For example, if XYZ buys less than 900 000 
MW in total, he will have to purchase this difference in quantity from ABC 
at the price $45/MWh, $45/MWh, \(K - S\), \(K = \max(K - S, 0)\).

To summarise, in this simple swing contract, there exists three compo-
nents, namely

1. **The forward component**
   The commitment to deliver (to buy) 50 000 MW of power per day for 
   September at a price of $45/MWh.

2. **The swing option component**
   The right but not the obligation to decrease amount to 30 000 MW per 
   day at $45/MWh, up to 10 times.

3. **The penalty component**
   If the party XYZ consumed less than 900 000 MW within the prespec-
   ified period, he must purchase from ABC this difference in quantity at 
   the price of \(K - S, 0\).

### 1.2.2 Definition of a Swing Contract

For the purpose of this paper, we will be using the following definition of a 
swing contract.

A **swing contract** is an agreement to buy or sell electricity over a given 
time period at a fixed price, with some constrained flexibility in the volume 
and timing. This contract has two components: a pure forward agreement
and a swing option.

The pure forward agreement is a set of forward contracts with different expiry dates \( t_j \in [T_1, T_2], j = 1, ..., N \). A forward contract is a cash market transaction in which delivery of the commodity is deferred until after the contract has been made. Although the delivery is made in the future, the price is determined on the initial trade date. Here, each forward contract \( F_j \) is based on a fixed amount of electricity \( b_j \) to be delivered over the period \((t_j, t_{j+1})\).

The second component introduces the flexibility of this contract. The owner of the contract has \( \nu \) exercise rights, with \( 0 \leq \nu \leq N \). A right can be exercised only at one of the discrete dates \( t_j \). At each expiry date \( t_j \) the holder has the option to purchase an excess amount (up-swings \( v^+ (t_j) \)) or decrease (down-swings \( v^- (t_j) \)).

For \( 1 \leq j \leq N \) we can define the decision indicator:
\[
\chi^+_j (t_j) = 1, \text{ if an up-swing at } t_j \text{ is performed}\]
\[
\chi^-_j (t_j) = 1, \text{ if a down-swing at } t_j \text{ is performed}\]
\[
\chi^+_j (t_j) = 0, \text{ otherwise}\]
\[
\chi^-_j (t_j) = 0, \text{ otherwise}\]

We can write \( 0 \leq \chi^+_j (t_j) + \chi^-_j (t_j) \leq 1, \forall 1 \leq j \leq N \).

The total number of swings, \( n^+_j (t_j) \) and \( n^-_j (t_j) \), are bounded, i.e. their values should not exceed \( n^+_j \) and \( n^-_j \) respectively:
\[
0 \leq n^+_j (t_j) = \sum_{k=1}^{j} \chi^+_j (t_k) \leq n^+_j,
\]
\[
0 \leq n^-_j (t_j) = \sum_{k=1}^{j} \chi^-_j (t_k) \leq n^-_j.
\]

The total volume delivered over the time period \([T_1, T_2]\) via swing contracts and the consumed amount per swing are typically restricted between bounds specified in the contract. A violation of these constraints is usually met with some penalty costs, which settled at expiration. A general penalty function \( \Phi(v) \) denotes the total penalty cost which should pay the holder of the swing contract at time \( T_2 \) for a total demand of \( v \) units over \([T_1, T_2]\).
\[
v = \sum_{j=1}^{N} [\chi^+_j (t_j) v^+_j (t_j) - \chi^-_j (t_j) v^-_j (t_j)].
\]
For example, a penalty function can be the following:

\[
\Phi(v) = \begin{cases} 
    \Phi(c_1), & \text{if } v \leq v^-, \\
    0, & \text{if } v^- < v < v^+, \\
    \Phi(c_2), & \text{if } v \geq v^+. 
\end{cases}
\]  

(1.1)

Following Wegner [21], swing contracts are very flexible concerning the contract specifications, but they also provide market participants with an important instrument to hedge their risk exposures due to unexpected movements in the market. Extreme events or scenarios are much more likely to occur in the electricity market than they do in the foreign exchange, interest rate or equity markets. Moreover, these extreme market states exist only for a very short period of time. Swing options, therefore, represent an additional opportunity for power producers to respond to sudden short-term demand. Thus, by prescribing the maximum number of up- and down-swings, as well as the penalty function, swing contracts reduce the uncertainty in power demand, commonly experienced by producers.
Chapter 2

The Least-Squares Monte Carlo Method

To date, Monte Carlo methods have been used significantly in the pricing of swing options. While the Monte Carlo valuation techniques are relatively straightforward, this approach allows for increasing complexity. In particular with the electricity swing options, using the Monte Carlo approach allows for a compounding in the uncertainty, i.e. simulation can accommodate uncertainties from more than one source, such as exchange rate, correlation between underlying sources of risk, etc.. The Monte Carlo approach also allows for models incorporating some form of stochastic volatility, i.e. the volatility of the underlying (electricity as our traded commodity) is time-dependent. While our focus of this thesis is not on the above-mentioned method, it is still necessary to understand how pricing of swing options is done using this method.

For Monte Carlo methods, the numerical procedure of early exercising is a challenge. The holder of the swing option must decide each time before maturity either to hold the option or to exercise the option. The least-squares Monte Carlo method (LSM) is a very popular method for pricing swing options, as proposed by Longstaff and Schwarz [12].

Using a finite set of simple basis functions, e.g. simple polynomial basis functions, the cash flow of early exercise can be fitted on a regression with the LSM. These simple basis functions can be considered proxy for the continuation value. The comparison between the approximated features cash-flows and that of immediate exercise gives us the optimal stopping rule.
The time horizon \([0, T]\) of the swing option can be discretised in the following way: the American option is represented like a strip of Bermudan options, i.e. Bermudan options with exercise dates \(0 = t_0 < t_1 < t_2 < ... < t_N = T\). We use in the sequel the idea of dynamic programming for a Bermudan option:

If the payoff from immediate exercise exceeds the continuation value, then exercise is performed at an exercise date \(t_k\). Then the value of the remaining part of the swing option (not exercised at \(t_k\)) can be written in the following form:

\[
F(\omega, t_k) = \mathbb{E}_\mathbb{Q} \left[ \sum_{j=k+1}^N e^{-r(t_j-t_k)} C(\omega, t_k, t_j, T|\mathcal{F}_{t_k}) \right],
\]

where \(F(\omega, t_k)\) is the expectation of the option payoff under the risk neutral measure \(\mathbb{Q}\), conditional on the information up to \(t_k\), where \(r\) is constant and \(C(\omega, t_k, t_j, T)\) is the path of cash flow. Then for a given \(\omega\), we start to step backwards in time from \(t_N\). Comparing \(F(\omega, t_{N-1})\) with the immediate payoff, \(P(S_{t_{N-1}})\), where \(S_{t_{N-1}}\) refers to the value of the underlying at time \(t_{N-1}\), we obtain the early exercise decision for time \(t_{N-1}\).

The continuation value, \(F(\omega, t_{N-1})\), has to be estimated, while \(P(S_{t_{N-1}})\) is known. Also, the conditional expectation can be approximated as a series of simple basis functions \(B_l\), \(l = 0, ..., L\) by

\[
\hat{F}(\omega, t_{N-1}) = \sum_{l=0}^L B_l \hat{B}_l(S_{t_{N-1}}) \tag{2.1}
\]

where \(B_l\) are coefficients, which are found by the regression of future cash-flows \(C(\omega, t_{N-1}, t_N, T)\) on the basis functions. It is important to note that the regression is done only over all paths that have a continuation value, in other words, options that are in-the-money at time \(t_{N-1}\), where \(\hat{F}(\omega, t_{N-1})\) is an estimator of the continuation value.

By comparing \(P(S_{t_{N-1}})\) with \(\hat{F}(\omega, t_{N-1})\) we obtain a new exercise decision. The new cash-flows at \(t_{N-1}\), \(C(\omega, t_{N-2}, t_{N-1}, T)\) is obtained from the maximum of \(P(S_{t_{N-1}})\) and \(\hat{F}(\omega, t_{N-1})\). Iteration steps are repeated until the first time level is reached. Note that for each path there exists only one
optimal stopping rule.

A cash flow at the time step $t_k$ is generated by early exercise in path $i$, all subsequent cash flows which occur in this path later than $t_k$ have to be removed. As a result, we obtain a Least-Squares Monte Carlo method, which uses an average over the cash flows from each path

$$V_{\text{LSM}}^{N_{\text{sim}}} = \sum_{i=0}^{N_{\text{sim}}} C_{\text{LSM}}(\omega_i),$$

where $N_{\text{sim}}$ is the number of simulations and $C_{\text{LSM}}$ the discounted cash flow.

Expectedly, the Least-Square Monte Carlo algorithm can be applied to swing options valuation. Additional dimensions are added to cash flows in order to calculate the number of exercise rights for up-swings and down-swings that were exercised, and also penalty functions, which depend on the total number of exercises.

Firstly, we consider a swing option with $n_{\uparrow}^+$ up-swings and $n_{\downarrow}^+$ down-swings.

The matrix of the cash flow is extended by the dimensions $n_{\uparrow}^+$, $n_{\downarrow}^+$, where $\Phi$ represents the penalty function of the swing contract, which comes into effect if some violations occur in the lifetime of the contract, i.e.

$$\Phi(n_{\uparrow}(t_N), n_{\downarrow}(t_N)) = \Phi(v).$$

Also, $n_{\uparrow} = n_{\uparrow}(t_{j-1})$ and $n_{\downarrow} = n_{\downarrow}(t_{j-1})$ are the numbers of up- and down-swings already exercised at a certain time step $t_{j-1}$, $S_{j(i)}$ is the price of the underlying at time $t_j$ of the path $\omega_i$, and $C_{j}^{n_{\uparrow},n_{\downarrow}}(i)$ is the cash flow with exercised rights $(n_{\uparrow}, n_{\downarrow})$.

### 2.1 The LSM Algorithm

1. The initialization time step $t_N$

   $$C_{N}^{n_{\uparrow},n_{\downarrow}}(i) = \max \left\{ v_{\uparrow}(S_N(i) - K_{\uparrow}) - \Phi^{n_{\uparrow}+1,n_{\downarrow}}, \ v_{\downarrow}(K_{\downarrow} - S_N(i)) - \Phi^{n_{\uparrow},n_{\down\downarrow}+1}, \ -\Phi^{n_{\uparrow},n_{\down\downarrow}}, \ -\Phi^{n_{\up\uparrow},n_{\down\down\downarrow}} \right\}.$$
Where \(0 \leq n_\downarrow \leq n_\downarrow^+\) holds
\[
C_{N_\downarrow}^{n_\downarrow, n_\downarrow^+} (i) = \max \left\{ v_\downarrow (K_\downarrow - S_N (i)) - \Phi_{n_\downarrow, n_\downarrow^+}^{+}, - \Phi_{n_\downarrow, n_\downarrow^+}^{-} \right\}.
\]

Where \(0 \leq n_\uparrow \leq n_\uparrow^+\) holds
\[
C_{N_\uparrow}^{n_\uparrow, n_\uparrow^+} (i) = \max \left\{ v_\uparrow (S_N (i) - K_\uparrow) - \Phi_{n_\uparrow, n_\uparrow^+}^{-}, - \Phi_{n_\uparrow, n_\uparrow^+}^{+} \right\}.
\]

2. The time step \(t_j, j = 1, \ldots, N - 1\).

Using the least square regression, coefficients \(b_l = b_{l_\uparrow, n_\downarrow} (j)\) from (2.1) are estimated by minimization of
\[
\left\| \sum_{m=j+1}^{N} e^{-r(t_m-t_j)} C (\omega_i, t_j, t_m, T) - \sum_{l=0}^{L} b_l B_l (S_j (i)) \right\|,
\]
where \(i = 1, \ldots, N_{\text{sim}}\).

The continuation value is equal to \(\text{Cont}_{j_\uparrow, n_\downarrow} (i) = \sum_{l=0}^{L} b_l B_l (S_j (i))\), then early exercise is performed if the following conditions are met.

The early exercise condition for the up-swings is
\[
v_\uparrow (S_j (i) - K_\uparrow) + \text{Cont}_{j_\uparrow, n_\downarrow} (i) > \text{Cont}_{j_\uparrow, n_\downarrow^+} (i)
\]
for \(0 \leq n_\uparrow < n_\uparrow^+, 0 \leq n_\downarrow \leq n_\downarrow^+\) and \(n_\uparrow + n_\downarrow < j\).

Correspondingly for down-swings, we have the following inequality
\[
v_\downarrow (K_\downarrow - S_j (i)) + \text{Cont}_{j_\downarrow, n_\downarrow} (i) > \text{Cont}_{j_\downarrow, n_\downarrow^+} (i)
\]
for \(0 \leq n_\uparrow < n_\uparrow^+, 0 \leq n_\downarrow \leq n_\downarrow^+\) and \(n_\uparrow + n_\downarrow < j\).

To summarize the conditions we have
\[
C_{j_\downarrow}^{n_\downarrow, n_\downarrow^+} = \begin{cases} 
    v_\uparrow (S_j (i) - K_\uparrow), & \text{if } v_\uparrow (S_j (i) - K_\uparrow) + \text{Cont}_{j_\uparrow, n_\downarrow}^{n_\uparrow+1, n_\downarrow} (i) > \text{Cont}_{j_\uparrow, n_\downarrow} (i), \\
    v_\downarrow (K_\downarrow - S_j (i)), & \text{if } v_\downarrow (K_\downarrow - S_j (i)) + \text{Cont}_{j_\downarrow, n_\downarrow}^{n_\uparrow+1, n_\downarrow} (i) > \text{Cont}_{j_\downarrow, n_\downarrow} (i), \\
    0, & \text{otherwise}.
\end{cases}
\]

3. The last step, obtaining \(t_0 = 0\).

By calculating the average value of the sums of rows \(C_{0_\downarrow, n_\downarrow} (i)\) after the final iteration, we then finally obtain the value of the swing option.
Chapter 3

Fundamentals of Finite Difference Methods

Introduction

There are a number of prominent features when applying a finite difference method to the pure heat equation

\[ \frac{\partial y}{\partial \tau} = \frac{\partial^2 y}{\partial x^2}, \quad x \in \mathbb{R}, \quad 0 \leq \tau \leq \frac{\sigma^2 T}{2}, \quad (3.1) \]

which is obtained through standard transformations \[14\] of the classical Black-Scholes equation.

While this Euler-type transformation, \( S = Ke^x \), is not applicable to our partial differential equation for pricing path-dependent options (and in the case of this thesis, swing-type options) but standard options instead, this allows us to simplify the explanation of the features of finite difference methods, which remain, fundamentally, the same.

1. The Difference Approximation

Using the Taylor’s series expansion, we know that every twice continuously differentiable function \( f \) can be expressed as

\[ f'(x) = \frac{f(x+h) - f(x)}{h} - \frac{h}{2} f''(\xi); \]

where \( \xi \) is some number between \( x \) and \( x+h \), and the precise position of \( \xi \) is usually unknown. By introducing a one-dimensional grid of dis-
crete points \( x_i \), i.e. \( \ldots < x_{i-1} < x_i < x_{i+1} < \ldots \), we can discretise some interval, \( I \subseteq \mathbb{R} \).

For a simple example, we choose an equidistant grid with a mesh size \( h := x_{i+1} - x_i \). Now, \( I \) is discretised. Also, we know that for \( f \in C^2 (I) \), \( f \) is bounded if \( I \) is bounded and the term \(-\frac{h}{2} f'' (\varsigma) \) can be written as \( O (h^p) \), which is the error term of order \( p \) (here \( p = 1 \)), for convenience. Rewriting our expression, we now have a one-sided forward difference quotient,

\[
f' (x_i) = \frac{f_{i+1} - f_i}{h} + O (h).
\]

For the partial derivatives of \( y (x, \tau) \), analogous expressions of the above expression holds, and quite obviously, it would include the discretisation in \( \tau \). It is suggested to therefore replace the neutral notation \( h \) with either \( \Delta x \) or \( \Delta \tau \) respectively. Error orders of \( p = 2 \) are obtained by either central differences, for e.g.

\[
f' (x_i) = \frac{f_{i+1} - f_{i-1}}{2h} + O (h^2) , \text{ for } f \in C^3
\]

\[
f'' (x_i) = \frac{f_{i+1} - 2f_i + f_{i-1}}{h^2} + O (h^2) , \text{ for } f \in C^4,
\]

or one-sided differences that can involve more terms than our original example, such as

\[
f' (x_i) = \frac{-f_{i+2} + 4f_{i+1} - 3f_i}{2h} + O (h^2) , \text{ for } f \in C^3.
\]

In which case, by rearranging the terms and indices gives us an example of an approximation backward differentiation formula (BDF),

\[
f_i \approx \frac{4}{3} f_{i+1} - \frac{1}{3} f_{i+2} - \frac{2}{3} h f' (x_i).
\]

An evident advantage of using equidistant grids is that algorithms are comparatively easy to implement and error terms can be easily derived using Taylor’s expansion.

2. The Discretisation of the Grid

In the previous subsection for the Difference-Approximation, we have seen an example in which \( x \) was discretised. While the spatial domain, the time interval or both can be discretised, if only one of the two independent variables is discretised, then a semi-discretisation consisting
of parallel lines is obtained. By way of illustration, a full discretisation leading to a two-dimensional grid is performed.

Let $\Delta x$ and $\Delta \tau$ be the mesh sizes of the discretisations of $x$ and $\tau$, where the step in $\tau$, $\Delta \tau := \frac{\tau_{\text{max}}}{\nu_{\text{max}}}$, for $\nu_{\text{max}} := \frac{1}{2} \sigma^2 T$ and $\nu_{\text{max}}$ is a suitably chosen integer. Spatial discretisation is more complicated as the infinite interval $-\infty < x < \infty$ has to be replaced by a finite interval $a \leq x \leq b$ where the end values, $a = x_{\text{min}} < 0$ and $b = x_{\text{max}} > 0$ must be appropriately chosen such that a sufficient quality of approximation is obtained for the corresponding values of $S_{\text{min}} = Ke^a$, $S_{\text{max}} = Ke^b$ and the interval $S_{\text{min}} \leq S \leq S_{\text{max}}$. The step length in $x$ is then defined by $\Delta x := \frac{(b-a)}{m}$, for a suitably chosen integer $m$.

Figure 3.1: The Discretised Grid in detail, with mesh sizes $\Delta x$ and $\Delta \tau$

As illustrated in Figure 3.1, this discretisation defines a two-dimensional uniform grid, where this equidistant grid is defined in terms of $x$ and $\tau$, and not in terms of $S$ and $t$. Transforming the $(x, \tau)$-grid back to the original $(S, t)$-plane would give us a non-uniform grid with unequal distances of the grid lines $S = S_{\ell} = Ke^{x_{\ell}}$, where the transformed grid is increasingly dense, the closer to $S_{\text{min}}$, and this may not be advantageous to the approximations of $V(S, t)$.

Grid lines $x = x_i$ and $\tau = \tau_\nu$ can be indicated with their indices, as seen in Figure 3.2 and the points where they intersect are called nodes. Since solutions are only defined for the nodes after discretisation, which is a significant contrast to the continuously defined analytical solution to the PDE, an error arises and is defined by the difference between the discretised and the analytical solution. The error depends on the
choice of parameters $\nu_{\text{max}}, m, x_{\text{min}}, x_{\text{max}}$. Therein lies the difficulty, as we do not know which choices for the parameters would match that of a prespecified error tolerance.

Figure 3.2: The Stencil of the Scheme, $\nu$ numerates the time steps and $i$ the spatial steps

It is important to note that for the *Finite Volume scheme* (see Chapter 6) used in this thesis, a semi-discretisation technique is applied instead, i.e. discretisation is done only for the spatial derivatives but the time variable remains continuous. This technique is the so-called “Method of Lines” (MOL).

3. The Explicit Euler Method

By substituting the difference quotients for $y_{i,\nu} \approx y(x_i, \tau_{\nu})$,

$$
\frac{\partial y_{i,\nu}}{\partial \tau} = \frac{y_{i,\nu+1} - y_{i,\nu}}{\Delta \tau} + O(\Delta \tau),
$$

$$
\frac{\partial^2 y_{i,\nu}}{\partial x^2} = \frac{y_{i+1,\nu} - 2y_{i,\nu} + y_{i-1,\nu}}{\Delta x^2} + O(\Delta x^2),
$$

into equation (3.1) and discarding the error terms, we obtain

$$
\omega_{i,\nu+1} = \lambda \omega_{i-1,\nu} + (1 - 2\lambda) \omega_{i,\nu} + \lambda \omega_{i+1,\nu},
$$

(3.2)

where $\omega_{i,\nu} \approx y(x_i, \tau_{\nu})$, and $\lambda := \frac{\Delta \tau}{\Delta x^2}$ denotes the parabolic mesh ratio.

In other words, it is easily observed that an evaluation of our equation is organised by *time levels*. This means that all nodes with the index $\nu$ form the $\nu$-th time level. We can then calculate the values $\omega_{i,\nu+1}$ for all $i$ of the next time level $\nu+1$, for a fixed $\nu$, and then carry
on with the next time level. Since this results in an explicit formula for all $\omega_{i,\nu+1}$ ($i = 0, 1, \ldots, m$), this gives the method its name, i.e. the *explicit method*, or otherwise known as, the *forward-in-time-difference method* (FTD method).

To begin, for $\nu = 0$, the values of $\omega_{i,0}$ are given by the initial condition (i.e. the payoff condition)

$$\omega_{i,0} = y(x_i, 0), \quad 0 \leq i \leq m,$$

where $y$ represents the transformed value functions of a standard call or put option. Values of $\omega_{0,\nu}$ and $\omega_{m,\nu}$ for $1 \leq \nu \leq \nu_{\text{max}}$ are supplied by suitable boundary conditions, i.e. $\omega_{0,\nu} = \omega_{m,\nu} = 0$.

For theoretical investigations, it is generally useful to collect all interior values $\omega$ of the time level $\nu$ into a vector

$$\omega^{(\nu)} := (\omega_{1,\nu}, \ldots, \omega_{m-1,\nu})^T.$$

Next, a constant $(m-1) \times (m-1)$ tridiagonal\footnote{This is dependent on the scheme used and may differ accordingly. In our particular problem, a tridiagonal matrix was used.} matrix is introduced such that the explicit method expressed in its matrix-vector form is

$$\omega^{(\nu+1)} = A\omega^{(\nu)} \text{ for } \nu = 0, 1, 2, \ldots,$$

with

$$A := A_{\text{expl}} := \begin{pmatrix}
1 - 2\lambda & \lambda & 0 \cdots & 0 \\
\lambda & 1 - 2\lambda & \ddots & \vdots \\
0 & \ddots & \ddots & \ddots & \ddots \\
& \ddots & \ddots & \ddots & \ddots & \ddots \\
0 & \cdots & 0 & \lambda & 1 - 2\lambda
\end{pmatrix}.$$

When using a computer programme for numerical computations instead, it is generally preferred to use

$$\omega_{i,\nu+1} = \lambda \omega_{i-1,\nu} + (1 - 2\lambda) \omega_{i,\nu} + \lambda \omega_{i+1,\nu}.$$

4. Stability

We denote the computer-calculated vector by $\bar{\omega}^{(\nu)}$ and the error vectors by

$$\varepsilon^{(\nu)} := \bar{\omega}^{(\nu)} - \omega, \text{ for } \nu \geq 0.$$
The result generated by a computer can then be rewritten as
\[ \tilde{\omega}^{(\nu+1)} = A\tilde{\omega}^{(\nu)} + r^{(\nu+1)}, \]
where the vectors \( r^{(\nu+1)} \) sum up the rounding errors that occur during the computation of \( A\tilde{\omega}^{(\nu)} \).

The following two lemmas provide us with the requirements for a method to be stable.

**Lemma 1.** The condition \( \rho(A) < 1 \) holds, if and only if \( A^\nu z \to 0 \), for all \( z \) and \( \nu \to \infty \), i.e. \( \rho(A) < 1 \) when \( \lim_{\nu \to \infty} \{A^\nu\}_{ij} = 0 \), where \( \rho(A) \) refers to the spectral radius\(^2\) of \( A \),
\[ \rho(A) := \max_i |\mu_i^A|, \]
where \( \mu_1^A, \ldots, \mu_{m-1}^A \) are the eigenvalues of \( A \). The proof of this lemma can be found in the Appendix.

Following Lemma 1, the requirement for a stable behaviour is then \( |\mu_i^A| < 1 \) for all eigenvalues, where \( i = 1, \ldots, m-1 \) in this case. This means that we need the eigenvalues of \( A \) in order to check the criterion of Lemma 1. To do that, we then split the matrix \( A \) into
\[ A = I - \lambda \cdot \begin{pmatrix} 2 & -1 & 0 \\ -1 & \ddots & \ddots \\ \vdots & \ddots & \ddots & -1 \\ 0 & \ddots & \ddots & 2 \end{pmatrix} =: G. \]

The eigenvalues of the matrix \( G \) are what remain to be determined.

**Lemma 2.** (See [14]). Let \( G = \begin{pmatrix} \alpha & \beta & 0 \\ \gamma & \ddots & \ddots \\ \ddots & \ddots & \ddots & \beta \\ 0 & \ddots & \ddots & \gamma & \alpha \end{pmatrix} \) be an \( N^2 \)-matrix,

\(^2\)The spectral radius of a matrix or a bounded linear operator is the supremum among the absolute values of the elements in its spectrum, and is sometimes denoted by \( () \).
where the eigenvalues $\mu_k^G$ and eigenvectors $v^{(k)}$ of $G$ are

$$
\mu_k^G = \alpha + 2\beta \sqrt{\gamma \beta} \cos \frac{k\pi}{N+1}, \quad k = 1, \ldots, N, \text{ and } \\
v^{(k)} = \left(\sqrt{\frac{\gamma \beta}{N+1}} \sin \frac{k\pi}{N+1}, \left(\sqrt{\frac{\gamma \beta}{N+1}} \right)^2 \sin \frac{2k\pi}{N+1}, \ldots, \left(\sqrt{\frac{\gamma \beta}{N+1}} \right)^N \sin \frac{Nk\pi}{N+1}\right)^T.
$$

Proof of Lemma 2 can be found by substituting $\mu_k^G$ and $v^{(k)}$ into $Gv = \mu^G v$.

From Lemmas 1 and 2, we observe that $N = m - 1, \alpha = 2, \beta = \gamma = -1$, and hence we obtain the eigenvalues $\mu^G$ and consequently also the eigenvalues $\mu^A$ of $A$, i.e.

$$
\mu_k^G = 2 - 2 \cos \frac{k\pi}{m} = 4 \sin^2 \left(\frac{k\pi}{2m}\right), \\
\mu_k^A = 1 - 4\lambda \sin^2 \left(\frac{k\pi}{2m}\right).
$$

The stability requirement $|\mu_i^A| < 1$ can then be written as

$$
\left|1 - 4\lambda \sin^2 \left(\frac{k\pi}{2m}\right)\right| < 1, \quad k = 1, \ldots, m - 1. \tag{3.3}
$$

Since we know that it is necessary for $\lambda > 0$, then (3.3) can be rewritten as $\frac{1}{2} > \lambda \sin^2 \left(\frac{k\pi}{2m}\right)$.

As the largest value of the sine term is $\sin \left(\frac{(m-1)\pi}{2m}\right)$ and this term grows monotonically to 1 as $m$ increases, we can then summarise the stability condition as follows

For $\lambda \leq \frac{1}{2}$, the explicit method $\omega^{(\nu+1)} = A\omega^{(\nu)}$ is stable.

Since $\lambda = \frac{\Delta x^2}{\Delta t^2}$, this stability criterion can be extended to bounding the $\Delta t$ step size:

$$
0 < \Delta t \leq \frac{\Delta x^2}{2}. \tag{3.4}
$$

5. The Implicit Euler Method

In the introduction to the explicit method, the time derivative was approximated with a forward difference, i.e. “forward” with respect to
the \( \nu \)-th time level. For an implicit method, we approximate with a backward difference

\[
\frac{\partial y_{i,\nu}}{\partial \tau} = \frac{y_{i,\nu} - y_{i,\nu-1}}{\Delta \tau} + O(\Delta \tau),
\]

which gives us the following as the corresponding alternative to (3.2)

\[
- \lambda \omega_{i+1,\nu} + (2\lambda + 1) \omega_{i,\nu} - \lambda \omega_{i-1,\nu} = \omega_{i,\nu-1}, \quad i = 0, 1, \ldots, m. \tag{3.5}
\]

As we can observe, (3.5) relates the time level \( \nu \) to the time level \( \nu - 1 \), where only the value of \( \omega_{i,\nu-1} \) is known, and we are left to find the three unknown values of \( \omega \) on the left-hand side of the equation.

![Figure 3.3: The Stencil of the Backward Difference Scheme](image)

Figure 3.3 shows us an imagery of the above relation. Then for the system of linear equations,

\[
A \omega^{(\nu)} = \omega^{(\nu-1)} \text{ for } \nu = 1, \ldots, \nu_{\max},
\]

where \( A := A_{\text{impl}} := \begin{pmatrix} 2\lambda + 1 & -\lambda & \cdots & 0 \\ -\lambda & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots \\ 0 & \cdots & \cdots & 0 \end{pmatrix} \)

and the vector \( \omega^{(\nu)} \) is implicitly defined as the solution of the system.

Again, it is assumed that \( \omega_{0,\nu} = \omega_{m,\nu} = 0 \) and for each time level \( \nu \) a system of equations as such has to be solved. To distinguish this particular method from other implicit methods, it is more accurately called
the backward time centred space scheme (BTCS), or simply known as the fully implicit or backward-difference method. The BTCS is unconditionally stable for $\Delta \tau > 0$, as shown analogously for the explicit case.

### 3.1 The Crank-Nicolson Method

For the methods mentioned earlier in the chapter, discretisation of $\frac{\partial y}{\partial \tau}$ are of the order $O(\Delta \tau)$. However, it is preferable to use a method where the time discretisation of $\frac{\partial y}{\partial \tau}$ has a higher order $O(\Delta \tau^2)$ and stability exists unconditionally. We consider again the heat equation \((3.1)\).

Crank and Nicolson suggested to average the forward- and backward-difference methods, i.e.

\[
\frac{\omega_{i, \nu+1} - \omega_{i, \nu}}{\Delta \tau} = \frac{\omega_{i+1, \nu} - 2\omega_{i, \nu} + \omega_{i-1, \nu}}{\Delta x^2}
\]

and backward-difference for time level $\nu + 1$:

\[
\frac{\omega_{i, \nu+1} - \omega_{i, \nu}}{\Delta \tau} = \frac{\omega_{i+1, \nu+1} - 2\omega_{i, \nu+1} + \omega_{i-1, \nu+1}}{\Delta x^2}
\]

Addition of these two approaches yields

\[
\frac{\omega_{i, \nu+1} - \omega_{i, \nu}}{\Delta \tau} = \frac{(\omega_{i+1, \nu} - 2\omega_{i, \nu} + \omega_{i-1, \nu} + \omega_{i+1, \nu+1} - 2\omega_{i, \nu+1} + \omega_{i-1, \nu+1})}{2\Delta x^2},
\]

\[(3.6)\]

Figure \[3.4\] gives an imagery and the following theorem summarises the features of this efficient method.

**Theorem 1. (Crank-Nicolson)** \[\text{[14]}\]

Suppose for $y \in C^4$ and $y$ is sufficiently smooth. Then

1. The order of the method is $O(\Delta \tau^2) + O(\Delta x^2)$.
2. For each time level $\nu$, a linear system of a simple tridiagonal structure must be solved.
3. Stability holds for all $\Delta \tau > 0$. 
While suitable boundary conditions $\omega_{0,\nu} = \omega_{m,\nu} = 0$ are still lacking, an algorithm for the basic version of the Crank-Nicolson Method for the PDE (3.1) can be formulated.

Algorithm 1. (Crank-Nicolson)

Start: Choose suitable values for $m$, $\nu_{\text{max}}$; calculate $\Delta x$, $\Delta \tau$

$\omega_{i,0} = y(x_i, 0)$, for $0 \leq i \leq m$

Calculate the $LR$-decomposition of $A$.

Loop: for $\nu = 0, 1, \ldots, \nu_{\text{max}} - 1$

Calculate $c := B\omega_{i,\nu}$ as a preliminary step.

Solve $Ax = c$, for example, using $LR$-decomposition\(^3\), i.e.

solve $Lz = B\omega_{i,\nu}$ and $Rx = z$,

$\omega^{(\nu+1)} := x$

When suitable boundary conditions can be implemented, Algorithm 1 has to be modified to include the effects of these conditions.

3.2 The Theta Method

Another popular method in the finite difference approach is the Theta method. For the purpose of explaining the essentials of this method, consider now the simple equation

$$\frac{\partial u}{\partial t} = k \frac{\partial u}{\partial x}, \quad x \in \mathbb{R}, \quad \tau > 0.$$

\(^3\)An $LR$ decomposition of a matrix $A$ is the factorisation of $A$ into a product of a lower (left) triangular matrix $L$ and an upper (right) triangular matrix $R$, $A = LR$
The equation can be solved by using a simple forward centred space differencing scheme

\[ u_{j}^{n+1} = u_{j}^{n} + k\Delta \tau \frac{u_{j+1}^{n} - u_{j-1}^{n}}{2\Delta x}, \]

where \( n \) is the time level and \( j \) refers to the spatial grid points. This scheme has a first order accuracy in time and second-order accuracy in space.

Using the von Neumann stability analysis (otherwise known as the Fourier stability analysis), we obtain the following ansatz

\[ u_{j}^{n} = \xi^{n} e^{ikj\Delta x}. \]

Then we can find the amplification factor \( \xi \), where \( \xi = 1 + \alpha \sin k\Delta x \) and \( \alpha = \frac{k\Delta \tau}{\Delta x} \). This method is unconditionally unstable since \( |\xi|^2 > 1, \forall \alpha \).

We can solve our equation also by using a backward time differencing approach, which yields an implicit scheme

\[ u_{j}^{n+1} = u_{j}^{n} + k\Delta \tau \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x}. \]

This scheme also has a first-order accuracy in time and a second-order accuracy in space. Then the respective amplification factor \( \xi = \frac{1}{1 - i\alpha \sin k\Delta x} \).

If \( |\xi|^2 < 1 \), the scheme becomes unconditionally stable.

The Theta method is obtained by the weighted averaging of the two schemes

\[ u_{j}^{n+1} = u_{j}^{n} + \theta k\Delta \tau \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2\Delta x} + (1 - \theta)k\Delta \tau \frac{u_{j+1}^{n} - u_{j-1}^{n}}{2\Delta x}. \]

For \( \theta = 0 \), the method is fully explicit.
For \( \theta = 1 \), it is fully implicit.
For \( \theta = \frac{1}{2} \), it is the previously-mentioned Crank-Nicolson method.

The corresponding amplification factor of the Theta-method

\[ \xi = \frac{1 + (1 - \theta) i\alpha \sin (k\Delta x)}{1 - \theta i\alpha \sin (k\Delta x)}. \]

The stability of the Theta-scheme depends on the value of \( \theta \). For example, for \( \frac{1}{2} \leq \theta \leq 1 \), \( |\xi|^2 < 1 \) holds for any choice of \( \alpha \).
Chapter 3. Fundamentals of Finite Difference Methods
Chapter 4

An Electricity Price Process

In order to examine our choice of the electricity price process, we will focus on one numerical approach to swing option pricing here by applying a partial differential equation (PDE) approach, using a mean-reverting process. For example, the Ornstein-Uhlenbeck process (OU) is one such process.

Why the Ornstein-Uhlenbeck Process?

The electricity prices have the following properties:

1. Mean-reversion. This means that with the increasing time horizon, volatility decreases.

2. Seasonal effects of the electricity market. This is a direct consequence of the non-storability of the electricity.

3. Occasional spikes. Extreme changes in prices occur occasionally for a short period of time.

From these properties, there exist many different approaches for modelling this process, for instance, the so-called lognormal mean-reversion process is a popular choice. In our case, we have chosen the Ornstein-Uhlenbeck process, which is also used quite often. This process demonstrates the mean-reverting features and is well-suited for modelling electricity prices, as the equilibrium of supply and demand is essential for the price fixing. The other property of electricity prices, i.e. seasonality is also reflected. The greatest disadvantage of this model, however, is its inability to cover the effects of extreme spikes.
In this model, the spot price is assumed to be of the form
\[ S_t = G(t) e^{X_t}, \]
where \( G(t) \) is the seasonality deterministic factor and \( X_t \) represents the stochastic factor, i.e. the deviation from the deterministic equilibrium level, \( G(t) \).

The dynamics of the stochastic part of the process is given by
\[ dX_t = -\alpha X_t dt + \sigma(t) dW_t, \tag{4.1} \]
where \( \alpha \) is the mean-reversion speed, \( \alpha = \text{constant} \), \( \sigma(t) \) represents the time-dependent volatility parameter, \( W_t \) is a standard Brownian motion, and \( G \) is a continuously differentiable function.

We assume also that the price changes exponentially
\[ S_t = \exp \left( \ln G(t) + X_t \right), \]
Applying Itô’s lemma (see Appendix I) to the stock price and combining with equation (4.1), we obtain the process for the electricity price
\[ dS_t = \alpha \left( \rho(t) - \ln(S_t) \right) S_t dt + \sigma(t) S_t dW_t, \tag{4.2} \]
where \( \rho(t) = \frac{1}{\alpha} \left[ \frac{d\ln(G(t))}{dt} + \frac{1}{2} \sigma^2(t) \right] + \ln(G(t)) \) is the mean reversion level.

The time-dependence of the mean-reversion level and the volatility are utilised to incorporate the seasonality into the asset’s price evolution.

Substituting \( x_t = \ln(S_t) \) and applying again Itô’s lemma and the normal distribution of \( dW_s \), we obtain for the price process
\[ dx_t = \left( \alpha \left( \rho(t) - x_t \right) - \frac{1}{2} \sigma^2(t) \right) dt + \sigma(t) dW_t. \tag{4.3} \]
Integrating (4.3), we obtain
\[ x_t = x_{t_0} e^{-\alpha(t-t_0)} + \int_{t_0}^{t} e^{-\alpha(t-s)} \left( \alpha \rho(s) - \frac{1}{2} \sigma^2(s) \right) ds \]
\[ + \int_{t_0}^{t} e^{-\alpha(t-s)} \sigma(s) dW_s. \tag{4.4} \]
Using a similar substitution for the deterministic part, $G(t)$:

$$x_1(t) = \ln G(t),$$

after which, we can write the equation for $\rho$ as

$$\rho(t) = \frac{1}{\alpha} \left( \frac{dx_1(t)}{dt} + \frac{1}{2} \sigma^2(t) \right) + x_1(t).$$

Using equations (4.2) and (4.4) yields

$$x_t = x_1(t) + (r_{t_0} - x_1(t_0)) e^{-\alpha(t-t_0)} + \int_{t_0}^{t} e^{-\alpha(t-s)} \sigma(s) dW_s.$$

With substitutions for $x_t$ and $x_1(t)$, we obtain the solution for the electricity price process $S_t = e^{x_t}$:

$$S_t = G(t) \left( \frac{S_{t_0}}{G(t_0)} \right) e^{-\alpha(t-t_0)} \times \exp \left\{ \int_{t_0}^{t} e^{-\alpha(t-s)} \sigma(s) dW_s \right\}.$$

### 4.1 A Partial Differential Equation

For the price process shown by (4.2), we can derive the PDE that governs the price of a derivative security using the usual delta-hedged portfolio (i.e., a portfolio whose delta is set or kept at or as close to zero as possible)

$$\Pi(S, t) = V(s, t) + \Delta V_1(S, t).$$

Due to the fact that electricity cannot be stored nor be used for arbitrage (i.e., the market is incomplete), we can represent $V_1(S, t)$ just like another contingent claim with the same underlying $S$, expiring at least on the same day as the considered security or thereafter, and by using the standard delta-hedging procedure, the random component will be eliminated, $d\Pi = 0$ and yields the PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2(t) S^2 \frac{\partial^2 V}{\partial S^2} + \left[ \alpha (\rho(t) - \ln(S)) - \xi(s, t) \sigma(t) \right] S \frac{\partial V}{\partial S} - rV = 0,$$

where $\xi(s, t)$ is used to represent the market price of risk, and $r$ is the risk-free interest rate, $r = \text{constant}$. 
In the current approach, we assume $\xi(s, t) = 0$, then we obtain
\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2(t) S^2 \frac{\partial^2 V}{\partial S^2} + \left[ \alpha (\rho(t) - \ln(S)) \sigma(t) \right] S \frac{\partial V}{\partial S} - rV = 0.
\]

If we assume $\sigma$ and $G$ to be time-independent constants, then we have
\[
\rho = \frac{1}{2} \alpha \sigma^2 + \ln(G)
\]
and our PDE simplifies to
\[
\frac{\partial V}{\partial t} + \frac{1}{2} \sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + \left[ \alpha (\rho - \ln(S)) \right] S \frac{\partial V}{\partial S} - rV = 0.
\] (4.5)

For spatial discretisation, one-sided derivatives have to be used at the boundaries.
Chapter 5

Numerical Pricing with the Theta Scheme

5.1 Applying a Finite Difference Approach

From equation (4.5), the PDE can be used for swing option pricing with dynamic programming and by applying finite difference schemes.

Let the time interval be discretised by the uniform grid points \( t_j = j \Delta t, \ j = 0, \ldots, N \) and the spatial domain discretized by \( S_m = m \Delta S, \ m = 0, \ldots, M \).

In other words, we have the following representations for derivatives:

\[
\begin{align*}
\frac{\partial V_{m,j}}{\partial t} &= \frac{V_{m,j+1} - V_{m,j}}{\Delta t} + O(\Delta t), \\
\frac{\partial V_{m,j}}{\partial S} &= \frac{V_{m+1,j} - V_{m-1,j}}{2\Delta S} + O(\Delta S^2), \\
\frac{\partial^2 V_{m,j}}{\partial S^2} &= \frac{V_{m+1,j} - 2V_{m,j} + V_{m-1,j}}{\Delta S^2} + O(\Delta S^2).
\end{align*}
\]
For the price interval \([0, S_m]\) we use the \(\theta\)-scheme to solve (4.5)

\[
\frac{V_{m,j+1} - V_{m,j}}{\Delta t} = (1 - \theta) \left[ \frac{\sigma^2}{2} S_m^2 \frac{V_{m+1,j} - 2V_{m,j} + V_{m-1,j}}{\Delta S^2} \\
+ \alpha (\rho - \ln(S_m)) S_m \frac{V_{m+1,j} - V_{m-1,j} - rV_{m,j}}{2\Delta S} \right] \\
+ \theta \left[ \frac{\sigma^2}{2} S_m^2 \frac{V_{m+1,j+1} - 2V_{m,j+1} + V_{m-1,j+1}}{\Delta S^2} \\
+ \alpha (\rho - \ln(S_m)) S_m \frac{V_{m+1,j+1} - V_{m-1,j+1} - rV_{m,j+1}}{2\Delta S} \right],
\]

where \(V_{m,j}\) is the pointwise approximation of the option price

\[
V_{m,j} \approx V(S_m, t_j).
\]

For the following three values that the implicitness parameter \(\theta, \theta \in [0, 1]\), can take, the corresponding scheme is applied. Where

- \(\theta = 0\) – we have the fully explicit scheme,
- \(\theta = \frac{1}{2}\) – the Crank-Nicolson scheme, and
- \(\theta = 1\) – the fully implicit scheme.

Using the tridiagonal matrix \(D = \{D_m\} = \{(a_m, b_m, c_m)\} \in \mathbb{R}^{M+1, M+1}\) with

\[
a_m = \frac{\Delta t S_m^2}{2\Delta S^2} - \frac{\alpha (\rho - \ln(S_m)) \Delta t S_m}{2\Delta S}, \\
b_m = -\frac{\Delta t S_m^2}{\Delta S^2} - r\Delta t, \\
c_m = \frac{\Delta t S_m^2}{2\Delta S^2} + \frac{\alpha (\rho - \ln(S_m)) \Delta t S_m}{2\Delta S},
\]

where \(m = 1, 2, \ldots, M + 1\),

then rewriting the PDE as a linear system of equations using the \(\theta\)-scheme, we have

\[
(1 - \theta D_m) V_{m,j+1} = (1 + (1 - \theta) D_m) V_{m,j}.
\]

At \(S_0 = 0\), equation (4.5) reduces to \(\partial_t V = rV\) and no further modifications
Numerical Schemes for Pricing Electricity Swing Options

are needed. We apply also a vanishing gamma boundary condition at $S_M$, i.e.
$$\Gamma = \frac{\partial^2 V}{\partial S^2} \approx \frac{V_{M+1,j} - 2V_{M,j} + V_{M-1,j}}{\Delta S^2} = 0.$$ 
Therefore $V_{M+1,j} = 2V_{M,j} - V_{M-1,j}$, which is, in other words, a linear spatial extrapolation.

For $0 \leq \theta \leq 1$, the general $\theta$-scheme is unconditionally stable in the $L_2$ norm, provided $\theta \geq \frac{1}{2}$. The Crank-Nicolson scheme is thus on the boundary of unconditional stability and is known to be consistent of the second order in time and space (see [7]). The scheme is unconditionally stable in the $L_2$-norm for the $L_2$ initial data. However, it is only conditionally convergent in the maximum norm, i.e. if
$$\Delta \tau \leq \frac{2}{M^2 \sigma^2 + r} \quad \text{and} \quad \sigma^2 \geq r,$$
conditions which can be fulfilled by the choice of parameters used.

5.2 The Boundary and Initial Conditions

Non-trivial cases

If the penalty function $[1,1]$, is equal to zero, there exists some non-trivial limiting cases that can be utilised to check the numerical evaluation.

1. One up-swing, one down-swing or one up- and down-swing, i.e.,

$$\begin{align*}
(n^+_1, n^+_1) &= (1, 0), \\
(n^+_1, n^+_i) &= (0, 1), \quad \text{or} \\
(n^+_1, n^+_1) &= (1, 1).
\end{align*}$$

The holder of the swing option has the same possibilities as the owner of a Bermudan call option, a Bermudan put option or the sum of both with the same strikes, amounts and exercise schedule. This is because a Bermudan option can only be exercised once, as is true for both kinds of swing rights in the special swing contracts under consideration.
Chapter 5. Numerical Pricing with the Theta Scheme

2. Many up-swings, down-swings or both as there are swing opportunities.

\[(n\uparrow, n\downarrow) = (\nu, 0), \quad (n\uparrow, n\downarrow) = (0, \nu), \text{ or} \quad (n\uparrow, n\downarrow) = (\nu, \nu).\]

The value of a swing contract is equivalent to a series of European call and put options with the expirations equal to the swing opportunities.

**Upper and lower bounds of the swing contract**

We derive a swing contract with an arbitrary number \(n\uparrow\) and \(n\downarrow\) of swing rights without a penalty payment.

1. The sum of \(n\uparrow\) Bermudan call options and \(n\downarrow\) Bermudan put options gives us the upper bound. The value of the Bermudan option bundle is always greater than or at least equal to the swing option’s value.

2. The sum of the \(n\uparrow\) most valuable European calls and \(n\downarrow\) most valuable European puts among the strip of the European options covering all exercise opportunities gives us the lower bound. The value of the European option collection is always less than or at most equal to the value of the swing option.

**Remark 1.** These bounds are valid only in the case if penalty payments do not exist.

**Initial conditions**

The following assumptions will provide us with a simplification of the general swing contract specifications:

- A constant incremental amount for the up- and down-swings

\[v\uparrow(t_n) \equiv v\uparrow \quad \text{and} \quad v\downarrow(t_n) \equiv v\downarrow.\]

- The total volume

\[v = n\uparrow(t_N) v\uparrow - n\downarrow(t_N) v\downarrow,\]

where \(n\uparrow(t_N)\) and \(n\downarrow(t_N)\) refer to the number of up- and down-swings taken at expiration of the contract.
• \(V_{m,j}(n_\uparrow(t_{j-1}), n_\downarrow(t_{j-1}))\) represents the value of the swing option with the underlying price \(S_m\), at swing opportunity \(t_j\), and \(n_\uparrow(t_{j-1})\) up-swings and \(n_\downarrow(t_{j-1})\) down-swings have already been exercised.

• The inequality \(n_\uparrow + n_\downarrow < j\) must hold since only a single swing right is allowed to be exercised at every opportunity.

• If an up-swing is exercised, the swing contract holder receives the volume \(v_\uparrow\) for a price of \(K_\uparrow(t_j)\) then \(v_\uparrow(S_j - K_\uparrow)\) is added to his balance sheet.

• After one up-swing right is exercised, then \(V_{m,j}(n_\uparrow(n_\uparrow + 1), n_\downarrow)\).

Correspondingly, after one down-swing right is exercised, then \(V_{m,j}(n_\uparrow, n_\downarrow + 1)\).

• Maximization of the option value

1. If neither the maximum number of up- nor down-swings is reached,
\[
V_{m,j}(n_\uparrow, n_\downarrow) = \max \left\{ V_{m,j}(n_\uparrow + 1, n_\downarrow) + v_\uparrow(S_m - K_\uparrow), \right.
\]
\[
V_{m,j}(n_\uparrow, n_\downarrow + 1) + v_\downarrow(K_\downarrow - S_m), \right.
\]
\[
V_{m,j}(n_\uparrow, n_\downarrow) \right\}
\]

2. If maximum number of up-swings is reached,
\[
V_{m,j}(n_\uparrow^+, n_\downarrow) = \max \left\{ V_{m,j}(n_\uparrow^+, n_\downarrow + 1) + v_\downarrow(K_\downarrow - S_m), \right.
\]
\[
V_{m,j}(n_\uparrow^+, n_\downarrow) \right\}
\]

3. If maximum number of down-swings is reached,
\[
V_{m,j}(n_\uparrow, n_\downarrow^+) = \max \left\{ V_{m,j}(n_\uparrow + 1, n_\downarrow^+) + v_\uparrow(S_m - K_\uparrow), \right.
\]
\[
V_{m,j}(n_\uparrow, n_\downarrow^+) \right\}
\]

In the case of penalty payments, the corresponding option values can be written as
\[
V_{m,N}(n_\uparrow, n_\downarrow) = \max \left\{ v_\uparrow(S_m - K_\uparrow) - \Phi^{n_\uparrow+1,n_\downarrow}, \right.
\]
\[
v_\downarrow(K_\downarrow - S_m) - \Phi^{n_\uparrow,n_\downarrow+1}, \right. - \Phi^{n_\uparrow+1,n_\down\uparrow} \right\}
\]
\[
V_{m,N}(n_\uparrow^+, n_\downarrow) = \max \left\{ v_\downarrow(K_\downarrow - S_m) - \Phi^{n_\uparrow^+,n_\down\uparrow+1}, \right. - \Phi^{n_\uparrow^+,n_\down\uparrow} \right\}
\]
\[
V_{m,N}(n_\uparrow, n_\downarrow^+) = \max \left\{ v_\uparrow(S_m - K_\uparrow) - \Phi^{n_\uparrow+1,n_\downarrow^+}, \right. - \Phi^{n_\uparrow+1,n_\downarrow^+} \right\}
\]

where $\Phi^{n^\uparrow, n^\downarrow} = \Phi (v^\uparrow n^\uparrow - v^\downarrow n^\downarrow)$. 
The Finite Volume Method

The Finite Volume method is a numerical method used for the representation and evaluation of solutions to partial differential equations. The values in this method are calculated, as with finite difference methods, with a discretised grid. In order to better understand this method, let us first consider the following as a simple example.

Let the interval $I = (0, X)$ be divided into $N$ sub-intervals. $I_i := (x_i, x_{i+1})$, where $i = 0, 1, \ldots, N-1$, such that $0 = x_0 < x_1 < \cdots < x_N = X$. For every $i = 0, 1, \ldots, N-1$, we denote $h_i = x_{i+1} - x_i$, where $h = \max_{0 \leq i \leq N-1} h_i$ and also let $x_{i-\frac{1}{2}} = \frac{x_{i-1} + x_i}{2}$ and $x_{i+\frac{1}{2}} = \frac{x_i + x_{i+1}}{2}$ for every $i = 1, 2, \ldots, N-1$. If instead, we define $x_{-\frac{1}{2}} = x_0$ and $x_{N+\frac{1}{2}} = x_N$, these mid-points will form a second partition of $(0, X)$. By integrating our PDE over these intervals will allow us to achieve a spatial discretisation.

Thereafter, to discretize this system, we let $t_i (i = 0, 1, \ldots, K)$ be a set of the partition points in $[0, T]$ satisfying $T = t_0 > t_1 > \cdots > t_K = 0$. Then, the two-level implicit time-stepping method can be applied with a splitting parameter $\theta \in \left[\frac{1}{2}, 1\right]$.

As mentioned in the chapter devoted to Finite Difference methods, when $\theta = \frac{1}{2}$, the time-stepping scheme becomes the CrankNicolson scheme and when $\theta = 1$ it is the backward Euler scheme. Both of these schemes are unconditionally stable and are of the second- and first-order accuracy respectively.
In other words, when our fully discretised system satisfies the discrete maximum principle, it would mean the discretization is monotonic. This guarantees that for a non-negative final condition, numerical solutions obtained by this method are non-negative, as option prices should be.

On a concluding note, we should bear in mind that the local approximations to the first and second partial derivatives, \( \frac{\partial u}{\partial x} \) and \( \frac{\partial^2 u}{\partial x^2} \), can be easily obtained and these two quantities, known respectively as the \( \Delta \) and \( \Gamma \) of an option, are important in practice. Particularly, the former is used for constructing portfolios that hedge against risk of price movements in the underlying asset (i.e. portfolios that are delta neutral), or otherwise also known as delta hedging.

### 6.1 Applying the Finite Volume Method to the Pricing of Swing Options

Beginning again with the PDE (4.5) we have for the valuation of electricity swing options, we first want to rewrite our equation in its conservative formulation:

\[
V_t + \frac{1}{2} \sigma^2 V_{ss} + \alpha (\rho - \ln S) SV_s - rV = 0. \tag{6.1}
\]

Since we know that \( V_{ss} = \partial_s (V_s) \) and also the following three relations

\[
\begin{align*}
    \partial_s (SV) &= SV_s + V, \\
    \partial_s (S^2 V_s) &= S^2 V_{ss} + 2 SV_s, \\
    \partial_s (S \ln S \cdot V) &= S \ln S \cdot V_s + (1 + \ln S) V.
\end{align*}
\]

Inserting these relations into (6.1) we obtain

\[
V_t + \frac{1}{2} \sigma^2 \left[ \partial_s (S^2 V_s) - 2 \partial_s (SV) + 2V \right] + \alpha \rho \left[ \partial_s (SV) - V \right] \\
- \alpha \left[ \partial_s (SV \ln S) - (1 + \ln S) V \right] - rV = 0
\]
Correspondingly,
\begin{equation*}
V_t + \frac{1}{2} \sigma^2 \partial_s (S^2 V_s) + (\alpha \rho - \sigma^2) \partial_s (SV) + (\sigma^2 - \alpha \rho - r) V
- \alpha \partial_s (SV \ln S) + \alpha (1 + \ln S) V = 0
\end{equation*}
and finally
\begin{equation*}
V_t = \alpha \partial_s (SV \ln S) - \frac{1}{2} \sigma^2 \partial_s (S^2 V_s) - (\alpha \rho - \sigma^2) \partial_s (SV)
- (\sigma^2 - \alpha \rho - r) V - \alpha (1 + \ln S) V,
\end{equation*}
which we can now proceed to integrate with respect to its spatial domain, so that we have a semi-discretisation with respect to space and continuous in time.

\begin{equation*}
\int_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}} V_t dS = \alpha \left[ SV \ln S \right]_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}}
- \frac{1}{2} \sigma^2 \left[ S^2 V_s \right]_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}}
- (\alpha \rho - \sigma^2) \left[ SV \right]_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}}
- (\sigma^2 - \alpha \rho - r) \Delta S_i V_i (t) - \alpha V_i \int_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}} (1 + \ln S) dS,
\end{equation*}
where \( V_i = V_i (S_i) \).

Now,
\begin{equation*}
\int_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}} (1 + \ln S) dS = \left[ S \ln S \right]_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}}
= \alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} V_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} V_{i-\frac{1}{2}} \right]
- \frac{1}{2} \sigma^2 \left[ S_{i+\frac{1}{2}}^2 D^0 V_{i+\frac{1}{2}} - S_{i-\frac{1}{2}}^2 D^0 V_{i-\frac{1}{2}} \right]
- (\alpha \rho - \sigma^2) \left[ S_{i+\frac{1}{2}} V_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} V_{i-\frac{1}{2}} \right]
- (\sigma^2 - \alpha \rho - r) \Delta S_i V_i (t)
- \alpha \left[ V S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - V S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right],
\end{equation*}
where \( V_{i+\frac{1}{2}} = \frac{V_{i+1} + V_i}{2} \) and \( V_{i-\frac{1}{2}} = \frac{V_{i-1} + V_i}{2} \).
To proceed from here, we first need to make some approximations for all our integrands to have the same discretised spatial indices

$$\int_{t_n}^{t_{n+1}} \int_{S_{i-\frac{1}{2}}}^{S_{i+\frac{1}{2}}} V_i \, dS \, dt = \int_{t_n}^{t_{n+1}} \Delta S \partial_i V_i(t) \, dt = \Delta S \left[ V_i^{n+1} - V_i^n \right]$$

This will give us the following approximated equation

$$\Delta S \left[ V_i^{n+1} - V_i^n \right] = \left\{ \alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} V_{i+\frac{1}{2}}^{n+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} V_{i-\frac{1}{2}}^{n+\frac{1}{2}} \right] 
- \frac{1}{2} \sigma^2 \left[ S_{i+\frac{1}{2}}^2 \Delta^{n+\frac{1}{2}} V_{i+\frac{1}{2}} - S_{i-\frac{1}{2}}^2 \Delta^{n+\frac{1}{2}} V_{i-\frac{1}{2}} \right] 
- (\alpha \rho - \sigma^2) \left[ S_{i+\frac{1}{2}} V_{i+\frac{1}{2}}^{n+\frac{1}{2}} - S_{i-\frac{1}{2}} V_{i-\frac{1}{2}}^{n+\frac{1}{2}} \right] 
- (\sigma^2 - \alpha \rho - r) \Delta S V_i^{n+\frac{1}{2}} 
- \alpha V_i^{n+\frac{1}{2}} \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right] \right\} \Delta t$$

Rearranging our equation, we obtain

$$\frac{\Delta S_i}{\Delta t} \left[ V_i^{n+1} - V_i^n \right] = \alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_{i+1}^n + V_{i+1}^{n+1} + V_{i}^{n}}{4} 
- S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \frac{V_{i}^{n+1} + V_i^n + V_i^{n+1} + V_{i-1}^{n}}{4} \right] 
- \frac{1}{2} \sigma^2 \left[ S_{i+\frac{1}{2}}^2 \frac{V_{i+1}^{n+\frac{1}{2}} - V_{i+\frac{1}{2}}^{n+\frac{1}{2}}}{h} - S_{i-\frac{1}{2}}^2 \frac{V_{i}^{n+\frac{1}{2}} - V_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{h} \right] 
- (\alpha \rho - \sigma^2) \left[ S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_{i+1}^n + V_{i+1}^{n+1} + V_{i}^{n}}{4} 
- S_{i-\frac{1}{2}} \frac{V_{i}^{n+1} + V_i^n + V_i^{n+1} + V_{i-1}^{n}}{4} \right] 
- (\sigma^2 - \alpha \rho - r) \Delta S_i \frac{V_i^{n+1} + V_i^n}{2} 
- \alpha \frac{V_i^{n+1} + V_i^n}{2} \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right]$$
and thus,
\[
\frac{\Delta S_i}{\Delta t} [V_i^{n+1} - V_i^n] = \alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \frac{V_{i-1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- \frac{\sigma^2}{2h} \left[ S_{i+\frac{1}{2}}^2 V_{i+1}^{n+\frac{1}{2}} - \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) V_i^{n+\frac{1}{2}} + S_{i-\frac{1}{2}}^2 V_{i-1}^{n+\frac{1}{2}} \right] \times 
\left( \alpha \rho - \sigma^2 \right) \left[ S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- S_{i-\frac{1}{2}} \frac{V_{i-1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} \right] 
- (\sigma^2 - \alpha \rho - r) \Delta S_i \frac{V_{i+1}^{n+1} + V_i^n}{2} 
- \alpha \frac{V_{i+1}^{n+1} + V_i^n}{2} \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right] \right] .
\]

We obtain
\[
\frac{\Delta S_i}{\Delta t} [V_i^{n+1} - V_i^n] = \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \frac{V_{i-1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- \frac{\sigma^2}{2h} \left[ S_{i+\frac{1}{2}}^2 V_{i+1}^{n+\frac{1}{2}} + V_i^n + V_{i}^{n+1} + V_i^n \right] 
- \frac{\sigma^2}{2h} \left[ S_{i-\frac{1}{2}}^2 V_{i-1}^{n+\frac{1}{2}} + V_i^n + V_{i}^{n+1} + V_i^n \right] 
- \frac{\sigma^2}{2h} \left[ S_{i+\frac{1}{2}}^2 V_{i+1}^{n+\frac{1}{2}} - \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) V_i^{n+\frac{1}{2}} + S_{i-\frac{1}{2}}^2 V_{i-1}^{n+\frac{1}{2}} \right] \times 
\left( \alpha \rho - \sigma^2 \right) \left[ S_{i+\frac{1}{2}} \frac{V_{i+1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} 
- S_{i-\frac{1}{2}} \frac{V_{i-1}^{n+1} + V_i^n + V_{i}^{n+1} + V_i^n}{4} \right] 
- (\sigma^2 - \alpha \rho - r) \Delta S_i \frac{V_{i+1}^{n+1} + V_i^n}{2} 
- \alpha \frac{V_{i+1}^{n+1} + V_i^n}{2} \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right] \right] .
\]

Multiplying by 4 and sorting with respect to time levels, we now obtain
\[
4 \frac{\Delta S_i}{\Delta t} V_i^{n+1} - \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} \left( V_{i+1}^{n+1} + V_i^{n+1} \right) 
+ \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \left( V_{i-1}^{n+1} + V_i^{n+1} \right) 
+ \frac{\sigma^2}{h} \left[ S_{i+\frac{1}{2}}^2 V_{i+1}^{n+1} - \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) V_i^{n+1} + S_{i-\frac{1}{2}}^2 V_{i-1}^{n+1} \right] 
+ (\sigma^2 - \alpha \rho - r) \Delta S_i \frac{2V_i^{n+1} + 2\alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right] V_i^n}{2}.
\]
\[ \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} (V_{i+1}^n + V_i^n) + 4 \frac{\Delta S_i}{\Delta t} V_i^n \]

\[ - \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} (V_i^n + V_{i-1}^n) \]

\[ - \frac{\sigma^2}{h} \left[ S_{i+\frac{1}{2}} V_{i+1}^n - \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) V_i^n + S_{i-\frac{1}{2}}^2 V_{i-1}^n \right] \]

\[ - 2 (\sigma^2 - \alpha \rho - r) \Delta S_i V_i^n - 2 \alpha \left[ S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right] V_i^n. \]

Hence,

\[ \left[ \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} + \frac{\sigma^2}{h} S_{i-\frac{1}{2}}^2 \right] V_{i-1}^{n+1} + \left[ 4 \frac{\Delta S_i}{\Delta t} \right] V_i^{n+1} \]

\[ - \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} + \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \]

\[ - \frac{\sigma^2}{h} \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) + 2 (\sigma^2 - \alpha \rho - r) \Delta S_i \]

\[ + 2 \alpha \left( S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right) V_i^{n+1} \]

\[ + \left[ - \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} + \frac{\sigma^2}{h} S_{i+\frac{1}{2}}^2 \right] V_{i+1}^{n+1} \]

\[ = - \left[ \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} + \frac{\sigma^2}{h} S_{i-\frac{1}{2}}^2 \right] V_i^n \]

\[ + \left[ 4 \frac{\Delta S_i}{\Delta t} + \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} \right. \]

\[ - \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \]

\[ + \frac{\sigma^2}{h} \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) - 2 (\sigma^2 - \alpha \rho - r) \Delta S_i \]

\[ - 2 \alpha \left( S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right) V_i^n \]

\[ + \left[ \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} - \frac{\sigma^2}{h} S_{i+\frac{1}{2}}^2 \right] V_{i+1}^n. \]

In other words, we can now rewrite the equations as a system \( A \vec{V}^{n+1} = B \vec{V}^n \), where \( \vec{V} = [V_0^{n+1}, V_1^{n+1}, ..., V_i^{n+1}]^T \).
Coefficients of the matrix $A$ are equal to

\[ a = \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \frac{\sigma^2}{h} S_{i-\frac{1}{2}} \right) S_{i+\frac{1}{2}} + \frac{\sigma^2}{h} S_{i-\frac{1}{2}} \]

\[ b = \left[ \frac{4 \Delta S_i}{\Delta t} - \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} + \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \right. 
\[ \left. + 2 \left( \sigma^2 - \alpha \rho - r \right) \Delta S_i + 2 \alpha \left( S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right) \right] \]

\[ c = \left( \sigma^2 - \alpha \rho - r \right) \Delta S_i 2V_{i+1} + 2 \alpha \left( S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right) \]

Coefficients of the matrix $B$ are equal to

\[ d = - \left[ \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} + \frac{\sigma^2}{h} S_{i-\frac{1}{2}} \right] \]

\[ e = \left[ \frac{4 \Delta S_i}{\Delta t} + \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} - \left( \alpha \ln S_{i-\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i-\frac{1}{2}} \right. 
\[ \left. + \frac{\sigma^2}{h} \left( S_{i+\frac{1}{2}}^2 + S_{i-\frac{1}{2}}^2 \right) - 2 \left( \sigma^2 - \alpha \rho - r \right) \Delta S_i \right. 
\[ \left. - 2 \alpha \left( S_{i+\frac{1}{2}} \ln S_{i+\frac{1}{2}} - S_{i-\frac{1}{2}} \ln S_{i-\frac{1}{2}} \right) \right] \]

\[ f = \left[ \left( \alpha \ln S_{i+\frac{1}{2}} - \alpha \rho + \sigma^2 \right) S_{i+\frac{1}{2}} - \frac{\sigma^2}{h} S_{i+\frac{1}{2}} \right] \]

Representing these relations in matrix-form, we obtain our matrices $A$ and $B$ in the following form

\[
A = \begin{pmatrix}
b_1 & c_1 \\a_2 & b_2 & c_2 \\
& \ddots & \ddots & \ddots \\
& & a_{m-2} & b_{m-2} & c_{m-2} \\
& & & a_{m-1} & b_{m-1} & c_{m-1}
\end{pmatrix},
\]

\[
B = \begin{pmatrix}
e_1 & f_1 \\d_2 & e_2 & f_2 \\
& \ddots & \ddots & \ddots \\
& & d_{m-2} & e_{m-2} & f_{m-2} \\
& & & d_{m-1} & e_{m-1} & f_{m-1}
\end{pmatrix}.
\]
Chapter 7

Results

In the previous chapters, we have provided a brief introduction to the three methods of pricing swing options, namely a Least-Squares Monte Carlo method, a Finite Difference method using the Theta and Crank-Nicolson approaches, as well a Finite Volume method. The theoretical and simulated results as well as the numerical comparison and analysis of these methods will be presented in this chapter.

We will first discuss the theoretical pros and cons of each of the above-mentioned methods as well as to compare them with each other. Thereafter, we compare the numerical results from the pricing of swing options.

7.1 Theoretical Results

We have discussed three numerical methods for pricing swing options, namely the Least-Squares Monte Carlo method, the finite difference method and the finite volume method. The Ornstein-Uhlenbeck process has been chosen as the pricing process model as it covers the main properties of the energy market, such as its mean-revision features and seasonal effects.

Two of these three methods have been implemented and numerical solutions are compared:

1. A Least-Squares Monte Carlo method (LSM)
Under the assumption that the optimal exercise strategy is chosen, this approach tackles the difficulty of evaluating the early exercise feature.

2. A **Finite Difference method by using the Theta and Crank-Nicolson approaches** (*FD*)
   These methods are based on the dynamic programming principles.

3. A **Finite Volume method** (*FV*)
   This method is based on the direct discretization of the conservation laws.

**Summarising the Advantages and Disadvantages of the methods**

1. A **Least-Squares Monte Carlo method**
   **Advantages:**
   (a) Easy implementation.
   (b) Applicable to any stochastic process.
   (c) Possibility of monitoring the numerical uncertainty of the early exercise feature.
   **Disadvantages:**
   (a) Slow convergence.

2. A **Finite Difference method**
   **Advantages:**
   (a) Easy implementation.
   (b) Fast convergence.
   (c) Easy computation of the *Greeks*.
   **Disadvantages:**
   (a) This method can not be applied to any stochastical process.
   (b) Boundary conditions are needed.

3. A **Finite Volume method**
   **Advantages:**
   (a) Method is conservative.
Numerical Schemes for Pricing Electricity Swing Options

(b) Easy formulation for unstructured meshes.

Disadvantages:

(a) The higher order differencing approximation greater than the second order are more difficult to develop in three dimensions.

Comparison of the different methods

1. LSM versus FD:
   
   (a) FD has faster convergence than LSM.
   
   (b) LSM can be applied to any stochastic price process but not FD.

2. FD versus FV:
   
   (a) The spatial discretization of the FV scheme is carried out directly in the physical space, therefore there are no problem with any transformation between coordinate systems as in the case of the finite difference method.
   
   (b) In comparison with FD, FV is very flexible and can be easily implemented on structured as well as on unstructured grids.

7.2 Simulation Results

7.2.1 Finite Difference Scheme

We will consider now the numerical results from the pricing of swing options by using a Finite-Difference scheme and then a comparison will be made with that obtained using the Least-Squares Monte Carlo Algorithm.

We first introduce some parameters, which had been used and calibrated by Wegner [21] and these are shown in Table 7.1.

The penalty function used is defined by

\[
\Phi(v) = \begin{cases} 
50, & \text{if } v \leq -1, \\
0, & \text{if } -1 < v \leq 2, \\
30(v - 2), & \text{if } v \leq -1, 
\end{cases}
\]
where \( \Phi(v) = \Phi(n^\uparrow - n^\downarrow) \).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean reversion rate, ( \alpha )</td>
<td>0.525</td>
</tr>
<tr>
<td>volatility, ( \sigma )</td>
<td>0.392</td>
</tr>
<tr>
<td>mean reversion level, ( \rho )</td>
<td>3.1783</td>
</tr>
<tr>
<td>( \bar{x} = \ln(G) )</td>
<td>3.032</td>
</tr>
<tr>
<td>time to maturity, ( T )</td>
<td>1</td>
</tr>
<tr>
<td>time horizon ([0, 1])</td>
<td></td>
</tr>
<tr>
<td>interest rate, ( r )</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 7.1: Common Parameters used and calibrated by Wegner [21]

Parameters used in the specification of possible swing options are shown in Table 7.2. The parameters specific to the Theta-scheme used are shown in Table 7.3.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>up-strike, ( K^\uparrow )</td>
<td>25</td>
</tr>
<tr>
<td>down-strike, ( K^\downarrow )</td>
<td>15</td>
</tr>
<tr>
<td>number of up-swings, ( n^\uparrow )</td>
<td>( \in {1, \ldots, 10} )</td>
</tr>
<tr>
<td>number of down-swings, ( n^\downarrow )</td>
<td>( \in {1, \ldots, 10} )</td>
</tr>
<tr>
<td>up-volume, ( v^\uparrow )</td>
<td>1</td>
</tr>
<tr>
<td>down-volume, ( v^\downarrow )</td>
<td>1</td>
</tr>
<tr>
<td>time-steps = exercise opportunities, ( \nu = N )</td>
<td>52 (weekly)</td>
</tr>
</tbody>
</table>

Table 7.2: The Option Specification Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \theta )</td>
<td>0.5</td>
</tr>
<tr>
<td>spot interval, ( S )</td>
<td>( \in [0, 40] )</td>
</tr>
<tr>
<td>number of spot grid points, ( M )</td>
<td>80</td>
</tr>
<tr>
<td>number of time steps, ( N )</td>
<td>52 (weekly)</td>
</tr>
</tbody>
</table>

Table 7.3: The Theta-scheme Parameters

With all the above-mentioned parameters, different combinations of these parameters used can be used to calibrate the respective value functions of
swing-type options with different specifications. To illustrate this, in the following figures, the value functions of a variety of swing-type options with varying number of up- and down-swings, as well as the existence of the penalty function will be shown as examples.

Figure 7.1(a) shows the value functions of swing-type options with a certain number up-swing rights, where \( n^+_\uparrow \in \{1, \ldots, 10\} \), no down-swing rights, \( n^+\downarrow = 0 \) and no penalty applied while Figure 7.1(b) shows the corresponding graph with a penalty applied to the same up- and down-swing rights for comparison.

Vice versa, swing-type options without any up-swing rights, \( n^+_\uparrow = 0 \), have a certain number down-swing rights, where \( n^+_\downarrow \in \{1, \ldots, 10\} \) and no penalty applied can also be simulated. A simulation of this value function as well as the corresponding graph with a penalty applied to the same up- and down-swing rights for comparison are simulated and these figures are shown in the Appendix II.

The value of swing options with both up- and down-swing rights in place, for example, \( n^+_\uparrow = 2 \times n^+_\downarrow \), where \( n^+_\downarrow \in \{1, \ldots, 10\} \) changes can also be simulatd. The following Figures 7.2(a) and 7.2(b) illustrate the option values with and without a penalty applied respectively. The graphs representing the reversed conditions where the swing options have more down-swing rights than up-swing rights, i.e. \( n^+_\downarrow \in \{1, \ldots, 10\} \), \( n^+_\uparrow = 2 \times n^+_\downarrow \) are presented in Appendix II.
Chapter 7. Results

(a) Without any penalty applied

(b) With a penalty applied

Figure 7.2: The swing-type option with more up-swing rights than down-swing rights, \( n_{\uparrow}^+ = 2 \times n_{\uparrow}^\downarrow \), where \( n_{\uparrow}^\downarrow \in \{1, ..., 10\} \)

We have some brief graphical representations of the value of different swing options using the Finite-Difference approach, the Figures 7.3(a), 7.3(b) and 7.3(c) from Wegner [21] on the following page illustrate the relative error of the two methods, the Finite-Difference approach and the Least-Squares Monte Carlo method, for different combinations of up- and down-swing opportunities of a given swing option. We see that the relative error decreases with increasing in-the-moneyness.

The parameters used in the simulation for the Least-Squares Monte Carlo method are represented in Table 7.4

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>spot interval, ( S )</td>
<td>( \in [0, 40] )</td>
</tr>
<tr>
<td>number of spot grid points, ( M )</td>
<td>20</td>
</tr>
<tr>
<td>number of simulations, ( N_{sim} )</td>
<td>20,000</td>
</tr>
</tbody>
</table>

Table 7.4: The Least-Squares Monte Carlo Method Parameters used for simulations presented in Wegner [21]
(a) Comparing valuation errors for swing options with only down-swing rights

(b) Comparing valuation errors for swing options with only up-swing rights

(c) Comparing valuation errors for swing options with equal number of up- and down-swing rights

Figure 7.3: Comparing Relative Valuation Errors between the two methods under the three above-stated cases.
7.2.2 The Finite Volume Scheme

Here we consider the simulated results from the pricing of swing options by using the Finite-Volume scheme.

We first present in Table 7.5 some parameters we have used in our calibration.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>mean reversion rate, $\alpha$</td>
<td>2.1</td>
</tr>
<tr>
<td>volatility, $\sigma$</td>
<td>0.01</td>
</tr>
<tr>
<td>mean reversion level, $\rho$</td>
<td>2.382</td>
</tr>
<tr>
<td>$\bar{x} = \ln(G)$</td>
<td>1.985</td>
</tr>
<tr>
<td>time to maturity, $T$</td>
<td>0.5</td>
</tr>
<tr>
<td>time horizon</td>
<td>[0, 1]</td>
</tr>
<tr>
<td>interest rate, $r$</td>
<td>0.05</td>
</tr>
</tbody>
</table>

Table 7.5: The Finite-Volume Scheme Parameters used in the calibration

By using the same penalty function defined for the Finite Difference scheme and all the above-mentioned parameters, different combinations of these parameters used can likewise be used to calibrate the respective value functions of swing-type options with different specifications. To illustrate this, in the following figures, the value functions of a variety of swing-type options with a varying number of up- and down-swings, as well as the existence of the penalty function will be shown as examples of the Finite Volume approach.

Figure 7.4(a) shows the value function of swing-type options with a certain number up-swing rights, where $n_i^+ \in \{1, \ldots, 10\}$, without any down-swing rights, $n_i^- = 0$ and no penalty applied. Figure 7.4(b) shows the corresponding graph with a penalty applied to the same up- and down-swing rights for comparison.

Similarly, the value functions of swing options with both up- and down-swing rights in place, for example, $n_i^+ = 2 \times n_i^+$, where $n_i^+ \in \{1, \ldots, 10\}$ is changed can be simulated. The following Figures 7.5(a) and 7.5(b) illustrate the option values with and without a penalty applied respectively.
Numerical Schemes for Pricing Electricity Swing Options

Figure 7.4: The swing-type option with up-swing rights, \( n_+ \in \{1, \ldots, 10\} \), without any down-swing rights, \( n_- = 0 \).

(a) Without any penalty applied

(b) With a penalty applied

Figure 7.5: The swing-type option with more up-swing rights than down-swing rights, \( n_+ = 2 \times n_- \), where \( n_- \in \{1, \ldots, 10\} \).

(a) Without any penalty applied

(b) With a penalty applied
Chapter 7. Results
Chapter 8

Conclusion

In our study, we examined in detail the features of the electricity market, as well as the fundamentals of energy derivatives, in particular, the swing options. We considered and implemented the most successful numerical methods to date for finding the optimal price of swing options. The methods used are the Least-Squares Monte Carlo method and the Finite Difference approach.

Innovation of our work is the use of the Finite Volume scheme to calculate the price of swing options. Previously, this method has not been used for this purpose, although it is a more accurate and stable method compared with the Finite Difference method and can be applied to any stochastic process. Both the Finite Difference and Finite Volume methods have been successfully programmed and presented.

In our work, we studied the Ornstein-Uhlenbeck process and its characteristics. Particular attention is paid to consideration of a differential equation which, given suitable initial and boundary conditions, describes the price of swing options. It is this equation that underlies the use of the Finite Difference and Finite Volume methods.

For other avenues of further research, we can implement the Least-Squares Monte Carlo method and compare its results with our programmed methods. It would also be very interesting for us to consider the question of finding optimal expiration strategies of swing options using the different methods.
Notations

\( \chi^\uparrow, \chi^\downarrow \) The decision indicator

\( n^\uparrow (t_j), n^\downarrow (t_j) \) The total number of swings

\( \Phi (v) \) The general penalty function

\( F (w, t_k) \) The expectation of the option payoff under the risk neutral measure \( Q \)

\( C (\omega, t_k, t_j, T) \) The cash flow

\( P (S_{t_N}) \) The immediate payoff

\( F (\omega, t_{N-1}) \) The continuation value

\( N_{sim} \) The number of simulations

\( n^\uparrow, n^\downarrow \) The number of up- and down-swings

\( \text{Cont}^n_{j} \) The continuation value

\( K \) The strike price

\( v \) The volume over the exercise dates

\( G (t) \) The seasonality deterministic factor

\( \alpha \) The mean-reversion speed

\( \sigma (t) \) The time-dependent volatility parameter

\( W_t \) The standard Brownian motion

\( \theta \) The implicit parameter

\( V^n_{i,j} \) The value of swing option
Bibliography


*Electricity Real Options Valuation.* Acta Physica Polonica B, 37, No. 11, 1001 – 1011.


Appendix I

I. Theorem 1 (See [14])
Suppose for \( y \in C^4 \) and \( y \) is smooth. Then

1. The order of the method is \( O(\Delta \tau^2) + O(\Delta x^2) \).
2. For each time level \( \nu \), a linear system of a simple tridiagonal structure must be solved.
3. Stability holds for all \( \Delta \tau > 0 \).

Proof of Theorem 1

1. Let \( \delta_x^2 \omega_i, \nu := \frac{w_{i+1, \nu} - 2w_{i, \nu} + w_{i-1, \nu}}{\Delta x^2} \) be the notation for the symmetric difference quotient of the second order for \( y_{xx} \).

Apply the operator \( \delta_x^2 \) to the exact solution \( y \). Then, by Taylor’s expansion for \( y \in C^4 \), it can be shown that

\[
\delta_x^2 y_{i, \nu} = \frac{\partial^2}{\partial x^2} y_{i, \nu} + \frac{\Delta x^2}{12} \frac{\partial^4}{\partial x^4} y_{i, \nu} + O(\Delta x^4).
\]

The local discretisation error, \( \varepsilon \), describes how well the exact solution \( y \) of equation (3.1) satisfies the difference scheme

\[
\varepsilon := \frac{y_{i, \nu+1} - y_{i, \nu}}{\Delta \tau} - \frac{1}{2} \left( \delta_x^2 y_{i, \nu} + \delta^2 \delta_x y_{i, \nu+1} \right) .
\]

Then, applying the same operator \( \delta_x^2 \) of (8.1), to the expansion of \( y_{i, \nu+1} \) at \( \tau_{\nu} \), and knowing that \( y_{\tau} = y_{xx} \) leads to

\[
\varepsilon = O(\Delta \tau^2) + O(\Delta x^2)
\]

61
(2) With \( \lambda := \frac{\Delta \nu}{\Delta x} \), Equation (3.6) is re-written as

\[
\begin{align*}
-\frac{\lambda}{2} \omega_{i-1, \nu+1} + (1 + \lambda) \omega_{i, \nu+1} - \frac{1}{2} \omega_{i+1, \nu+1} &= \frac{\lambda}{2} \omega_{i-1, \nu} + (1 + \lambda) \omega_{i, \nu} + \frac{1}{2} \omega_{i+1, \nu} \\
\end{align*}
\]

(8.2)

The values of the new time level, \( \nu + 1 \) are implicitly given by the system of equations (8.2). For the simplest boundary conditions, \( \omega_{0, \nu} = \omega_{m, \nu} = 0 \), (8.2) is a system of \( m-1 \) equations, i.e. (8.2) can be re-written as

\[
A \omega^{(\nu+1)} = B \omega^{(\nu)} \tag{8.3}
\]

where

\[
A := \begin{pmatrix}
1 + \lambda & -\frac{\lambda}{2} & 0 \\
-\frac{\lambda}{2} & \ddots & \ddots \\
0 & \ddots & -\frac{\lambda}{2} \\
1 - \lambda & \frac{\lambda}{2} & 0
\end{pmatrix},
\]

\[
B := \begin{pmatrix}
\frac{\lambda}{2} & \ddots & \ddots \\
\ddots & \ddots & \ddots \\
0 & \frac{\lambda}{2} & 1 - \lambda
\end{pmatrix}.
\]

The eigenvalues of \( A \) are real and lie between 1 and \( 1 + 2\lambda \). This rules out a zero eigenvalue, and so \( A \) must be non-singular and the solution of (8.3) is uniquely defined.

(3) The matrices, \( A \) and \( B \), as shown above, can be re-written in terms of a constant tridiagonal matrix,

\[
A = I + \frac{\lambda}{2} G, \quad G := \begin{pmatrix}
2 & -1 & 0 \\
-1 & \ddots & \ddots \\
0 & \ddots & -1
\end{pmatrix}, \quad B = I - \frac{\lambda}{2} G.
\]

Therefore, Equation (8.3) reads as

\[
(2I + \lambda G) \omega^{(\nu+1)} = (2I - \lambda G) \omega^{(\nu)} = (4I - 2I - \lambda G) \omega^{(\nu)} = (4I - C) \omega^{(\nu)},
\]

62
which leads to the formally explicit iteration
\[ \omega^{(\nu+1)} = (4C^{-1} - I) \omega^{(\nu)}. \] (8.4)

II. Itô’s Lemma

Let \( F(S_t, t) \) be a twice differentiable function, and random process, \( dS_t = \sigma_t dW_t + \gamma_t dt \).

Then by Itô’s Lemma

\[
dF_t = \frac{\partial F}{\partial S_t} dS_t + \frac{\partial F}{\partial t} dt + \frac{1}{2} \cdot \frac{\partial^2 F}{\partial S_t^2} \sigma_t^2 dt
\]

\[
\Rightarrow dF_t = \left[ \frac{\partial F}{\partial S_t} \gamma_t + \frac{\partial F}{\partial t} + \frac{1}{2} \cdot \frac{\partial^2 F}{\partial S_t^2} \sigma_t^2 \right] dt + \frac{\partial F}{\partial S_t} \sigma_t dW_t.
\]
Appendix II

Simulated Results

Figure 8.1: The swing-type option without any up-swing rights, $n^+_\uparrow = 0$, down-swing rights $n^+_\downarrow \in \{1, \ldots, 10\}$.

Figure 8.2: The swing-type option with more down-swing rights than up-swing rights, $n^+_\downarrow = 2 \times n^+_\uparrow$, where $n^+_\uparrow \in \{1, \ldots, 10\}$.
Appendix III

MATLAB Programming Code

A. Finite Difference Scheme

```matlab
function price = Swing(S0,K, r, T, v, G, alpha, Smax, dS, dt, U, u, D, d, delta)
clear all

% parameters (Thesis Wilhelm, p.111)
S0=200;
K=100; % strike
r=0.05; % iterest rate
T=10; % final time
sigma=0.3; % volatility= sigma
G=1; % seasonality parameter
alpha=.4; % daily mean reversion speed

% discretization parameters
Smax= 200; % max. spatial interval.
dS = 0.5; % spatial step
dt= 0.001; % time step
%dt=0.01;
theta= 0.5;

Up = 200; % max. number up-swings
u= 105; % number of already exercised up-swings
Down= 240; % max. number down-swings
d= 150; % number of already exercised down-swings
deltaper = 5; % consumption of electricity per up-swing
deltamn = -3; % consumption of electricity per down-swing
```
div = 0.03; % dividend yield

Up = 6; % max. number up-swings
u = 2; % number of already exercised up-swings
Down = 4; % max. number down-swings
d = 0; % number of already exercised down-swings
deltaup = 1; % consumption of electricity per up-swing
deltadn = 0.5; % consumption of electricity per down-swing
div = 0;

% Spatial Grid
M = round(Smax/dS); % number of spatial grid points
disp(['M = ',num2str(M)])
dS = Smax/M;
disp(['dS = ',num2str(dS)])
disp(' ')

% temporal Grid
N = round(T/dt);
disp(['N = ',num2str(N)])
dt = T/N;
disp(['dt = ',num2str(dt)])

matval = zeros(M+1,N+1); % to store V(S,t)
vetS = linspace(dS, Smax+dS, M+1); % spatial vector with M+1 points

veti = 0:N; % vector of time indices
vetj = 1:M+1; % vector of space indices
x = log(vetS); % standard Euler transformation in space

rho1 = 10; % value for penalty
rho = 1/(2*alpha)*sigma^2 + log(G);

% coefficients of system matrix M1, M2
a = (dt*(vetj.^2)*sigma^2)/2 - alpha*((rho-log(vetS))*dt.*vetj)/2;
b = -(dt*(vetj.^2)*sigma^2) - r*dt;
c = (dt*(vetj.^2)*sigma^2)/2 + alpha*((rho-log(vetS))*dt.*vetj)/2;
%% test: without rho/log term
% a = (dt*(vetj.^2)*sigma^2)/2 - alpha*dt*vetj/2;
% b = -(dt*(vetj.^2)*sigma^2) - r*dt;
% c = (dt*(vetj.^2)*sigma^2)/2 + alpha*dt*vetj/2;

% for testing
% a = (dt*sigma^2)/2/dS^2*ones(1,M+1);
% b = -(dt*sigma^2)/dS^2*ones(1,M+1);
% c = (dt*sigma^2)/2/dS^2*ones(1,M+1);

% D = diag(a(3:M),-1)+diag(b(2:M))+ diag(c(2:M-1),1);
% D = spdiags([a(2:M+1) 0]’ b’ [0 c(1:M)]’, -1:1, M+1, M+1);
D = diag(a(2:M+1),-1)+ diag(b) + diag(c(1:M),1);
M1 = eye(M+1,M+1) - theta *D;

% Change M1 according to boundary conditions (first/last row)
% matval(1,:) = D*K*exp( -r* dt );
% matval(M+1,:) = Up-K*exp(-r*dt);
M1(1,1) = 1; M1(1,2)=0;
M1(M+1,M) = 0; M1(M+1,M+1)=1;

% also for right hand side
M2 = eye(M+1,M+1) + (1-theta) *D;
M2(1,1) = 0; M2(1,2)=0;
M2(M+1,M) = 0; M2(M+1,M+1)=0;

% vector for inhom BCs
inhomBC=zeros(M+1,1);
inhomBC(1) = Down*K*exp( -r* dt );
inhomBC(M+1) = Up*(exp(x(M+1)-dt) - K*exp(-r*dt));
% for testing
% inhomBC(1) = 0;
% inhomBC(M+1) = 0;

% Initial conditions
if (0 <= u < Up) && (0 <= d < Down)
    v1 = max(u*deltaup*(vetS-K), d*deltadn*(K-vetS)) - rho1;
    v2 = max((u+1)*deltaup*(vetS-K), d*deltadn*(K-vetS)) - rho1;
    v3 = max(u*deltaup*(vetS-K),(d+1)*deltadn*(K-vetS)) - rho1;
    m1 = max(v1,v2);
    m2 = max(v3,0);
    matval(:,1) = max(m1, m2);
    disp(['max 1 = ',num2str(max(matval(:,1)))])
elseif (u == Up) && (0 <= d < Down),
    matval(:,1)= max(max(Up*deltaup*(vetS-K), ...
    d*deltadn*(K-vetS)) - rho1,...
    max(Up*deltaup*(vetS-K),...
    (d+1)*deltadn*(K-vetS)) - rho1,0);
elseif (0 <= u < Up) && (d == Down),
    matval(:,1)= max(max(u*deltaup*(vetS-K),...
    Down*deltadn*(K-vetS)) - rho1,...
    max((u+1)*deltaup*(vetS-K), Down*deltadn*(K-vetS)) - rho1,0);
elseif (u == Up) && (d == Down),
    matval(:,1)= max(max(Up*deltaup*(vetS-K),...
    d*deltadn*(K-vetS)) - rho1, 0);
end;

% Scaling = diag(1./vetj);
% M1=Scaling*M1;M2=Scaling*M2;inhomBC=Scaling*inhomBC;

% solve system
% LU decomposition
[ L, U ] = lu(M1);

% solution
for j=1:N
    % old=matval(1:M+1,j);
    % time-dependent BCs
    inhomBC(1) = Down*K*exp( -r* dt*j );
    % inhomBC(M+1) = Up*(exp(x(M+1)-dt*j) - K*exp(-r*dt*j));
    % inhomBC(M+1) = Up*(Smax*exp(-div*dt*j) - K*exp(-r*dt*j));
    % linear extrapolation
    % inhomBC(1) = 2* matval(2,j)-matval(3,j);
    % inhomBC(M+1) = 2* matval(M,j)-matval(M-1,j) ;
matval(1:M+1,j+1) = U (L (M2*matval(1:M+1,j) + inhomBC));

% new = U (L (M2*old + inhomBC));
% new = M1 (M2*old + inhomBC);
% matval(1:M+1,j+1) = new;

end
for i=2:1001
price2(i) = interp1(vetS, matval(:,i), S0);
end
figure(1); clf; hold on;
plot(veti(:,30:1001), price2(:,30:1001))
save('price2')

B. Finite Volume Scheme

function price = Swing(S0,K, r, T, v, G, alpha, Smax, dS, dt, U, u, D, d, delta)
clear all

%S0=200;
K=100; % strike
r=0.05; % interest rate
T=0.005; % final time
sigma=0.3; % volatility = sigma
G=1; % seasonality parameter
alpha=.4; % daily mean reversion speed

Up = 3; % max. number up-swings
u=2; % number of already exercised up-swings
Down=6; % max. number down-swings
d=2; % number of already exercised down-swings

c deltaup = 1; % consumption of electricity per up-swing
deltadn = 0.5; % consumption of electricity per down-swing
div =0;

%S0=200;
K=100;
Kup=99; % up strike
Kdown=49; % down strike
r=0.05; % interest rate
T=0.5; % final time
sigma=0.1; % volatility=sigma
G=7; % seasonality parameter
alpha=2.1; % daily mean reversion speed

% discretization parameters
Smax= 200; % max. spatial interval.
dS = 7; % spatial step
dt= 5; % time step
%dt=0.01;
%theta= 0.5;

Up = 6; % max. number up-swings
u= 6; % number of already exercised up-swings
Down=4 ; % max. number down-swings
d= 3; % number of already exercised down-swings

deltaup = 2; % consumption of electricity per up-swing
deltadn = 1; % consumption of electricity per down-swing

% spatial Grid
M= round(Smax/dS); % number of spatial grid points
disp([’M =’, num2str(M)])
dS = Smax/M;
disp([’dS =’, num2str(dS)])
disp(‘ ’)

% temporal Grid
N = round(T/dt);
disp([’N =’, num2str(N)])
dt = T/N;
disp([’dt =’, num2str(dt)])

matval = zeros(M+1,N+1); % to store V(S,t)
vetS = linspace(dS, Smax, M+1); % spatial vector with M+1 points

veti = 0:N; % vector of time indices
vetj = 1:M+1; % vector of space indices

x = log(vetS); % standard Euler transformation in space

rho1 = 2; % value for penalty
rho = 1/(2*alpha)*sigma^2 + log(G);

% coefficients of system matrix M1, M2

b = 4*(dt)/(dS)...
- (alpha*log(vetS+dS/2)-alpha*rho+sigma^2).* (vetS+dS/2)...
+ (alpha*log(vetS-dS/2)-alpha*rho+sigma^2).* (vetS-dS/2)...
- sigma^2/dS*( (vetS+dS/2).^2+(vetS-dS/2).^2)...
+ 2*(sigma^2-alpha*rho-r)*dS...
+ 2*alpha*( (vetS+dS/2).* log(vetS+dS/2) - (vetS-dS/2).* log(vetS-dS/2) ) ;

a = (alpha*log(vetS-dS/2)-alpha*rho+sigma^2).* (vetS-dS/2)...
+ sigma^2/dS*(vetS-dS/2).^2;

c = -(alpha*log(vetS+dS/2) - alpha*rho + sigma^2).* (vetS+dS/2)...
+ sigma^2/dS*(vetS+dS/2).^2;

%%% test: without rho/log term
% a = (dt*(vetj.*2)*sigma^2)/2 - alpha*dt*vetj/2;
% b = -(dt*(vetj.*2)*sigma^2) - r*dt;
% c = (dt*(vetj.*2)*sigma^2)/2 + alpha*dt*vetj/2;

% for testing
% a = (dt*sigma^2)/2/dS^2*ones(1,M+1);
% b = -(dt*sigma^2)/dS^2*ones(1,M+1);
% c = (dt*sigma^2)/2/dS^2*ones(1,M+1);

%D = diag(a(3:M),-1)+diag(b(2:M))+ diag(c(2:M-1),1);
%D = spdiags([a(2:M+1)],[b(1:M+1)]',-1 : 1, M + 1, M + 1);
M1 = diag(a(2:M+1),-1)+ diag(b(1:M+1)) + diag(c(1:M),1);
%M1 = eye(M+1,M+1) - theta *D;

% Change system matrix M1 according to boundary conditions (first/last
%matval(1,:) = D*K*exp(-r*dt);
%matval(M+1,:) = Up-K*exp(-r*dt);
M1(1,1) = 1; M1(1,2)=0;
M1(M+1,M) = 0; M1(M+1,M+1)=1;

bb = 4*(dt)/(dS)...
  + (alpha*log(vetS+dS/2)-alpha*rho+sigma^2)*(vetS+dS/2)...
  - (alpha*log(vetS-dS/2)-alpha*rho+sigma^2)*(vetS-dS/2)...
  + sigma^2*dS*( (vetS+dS/2)^2+(vetS-dS/2)^2)...
  - 2*(sigma^2-alpha*rho)*dS...
  - 2*alpha*((vetS+dS/2).*log(vetS+dS/2)-(vetS-dS/2).*log(vetS-dS/2));

aa = -(alpha*log(vetS-dS/2)-alpha*rho+sigma^2)*(vetS-dS/2)...
  + sigma^2*dS*(vetS-dS/2)^2;

cc = (alpha*log(vetS+dS/2) -alpha*rho +sigma^2)*(vetS+dS/2)...
    - sigma^2*dS*(vetS+dS/2)^2;

M2 = diag(aa(2:M+1),-1)+ diag(bb(1:M+1)) +diag(cc(1:M),1);

% also for right hand side
%M2 = eye(M+1,M+1) + (1-theta) *D;
M2(1,1) = 0; M2(1,2)=0;
M2(M+1,M) = 0; M2(M+1,M+1)=0;

% vector for inhom BCs
inhomBC=zeros(M+1,1);
inhomBC(1) = Down*K*exp(-r*dt);
inhomBC(M+1) = Up*(exp(x(M+1)-dt) - K*exp(-r*dt));
% for testing
%inhomBC(1) = 0;
%inhomBC(M+1) = 0;

% Initial conditions
if (0 <= u < Up) && (0 <= d < Down)
    v1 = max(u*deltap*(vetS-Kup), d*deltadn*(Kdown-vetS))-

74
\[ \begin{align*}
\rho_1; \\
v_2 &= \max((u+1) \cdot \text{deltaup} \cdot (\text{vetS-Kup}), d \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1; \\
v_3 &= \max(u \cdot \text{deltaup} \cdot (\text{vetS-Kup}), (d+1) \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1; \\
m_1 &= \max(v_1, v_2); \\
m_2 &= \max(v_3, 0); \\
\text{matval}(:,1) &= \max(m_1, m_2); \\
\text{disp}([\max_1 = ', \text{num2str}(\max(\text{matval}(:,1)))])
\end{align*} \]

elseif (u == Up) && (0 \neq d \neq Down),
\text{matval}(:,1) = \max(\max(\text{Up} \cdot \text{deltaup} \cdot (\text{vetS-Kup}), ... \\
\text{d} \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1,... \\
\max(\text{Up} \cdot \text{deltaup} \cdot (\text{vetS-Kup}), ... \\
(d+1) \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1,0); \\
\text{elseif (0 \neq u \neq Up) \&\& (d == Down),} \\
\text{matval}(:,1) = \max(\max(u \cdot \text{deltaup} \cdot (\text{vetS-Kup}), ... \\
\text{Down} \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1,... \\
\max((u+1) \cdot \text{deltaup} \cdot (\text{vetS-Kup}), \text{Down} \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1,0); \\
\text{elseif (u == Up) \&\& (d == Down),} \\
\text{matval}(:,1) = \max(\max(\text{Up} \cdot \text{deltaup} \cdot (\text{vetS-Kup}), ... \\
\text{d} \cdot \text{deltadn} \cdot (\text{Kdown-vetS})) - \rho_1, 0); \\
\text{end;}
\%
\text{Scaling} = \text{diag}(1./\text{vetj}); \\
\text{M1} = \text{Scaling} \cdot \text{M1}; \text{M2} = \text{Scaling} \cdot \text{M2}; \text{inhomBC} = \text{Scaling} \cdot \text{inhomBC}; \\
\%
\text{solve system} \\
[L, U] = \text{lu}(\text{M1}); \% \text{LU decomposition}
\%
\text{solution}
\%
\text{for j=1:N}
\%
\text{old=matval(1:M+1,j);} \\
\text{inhomBC}(1) = \text{Down} \cdot \text{K} \cdot \exp(-r \cdot dt \cdot j); \\
\%
\text{inhomBC(M+1) = Up}(\exp(x(M+1)-dt \cdot j) - K \cdot \exp(-r \cdot dt \cdot j));
inhomBC(M+1) = U^p(Smax*exp(-div*dt*j) - K*exp(-r*dt*j));

% linear extrapolation
%inhomBC(1) = 2* matval(2,j)-matval(3,j);
%inhomBC(M+1) = 2* matval(M,j)-matval(M-1,j) ;

matval(1:M+1,j+1) = U(L(M2*matval(1:M+1,j) + inhomBC));
%new=U(L(M2*old + inhomBC));
%new=M1(M2*old + inhomBC);
%matval(1:M+1,j+1)=new;

end

%for i=2:N+1
%price2(i) = interp1(vetS, matval(:,i), S0);
%end
figure(1);clf;hold on;
%plot(veti(2:N+1 ), price2(2:N+1))
plot(vetS, matval(1:M+1,N+1))

%save('price2')