Measuring the Risk-neutral Probability Distribution of Equity Index Options

Authors: Gustav Dackner, Linus Falk

External Supervisor: Filip Mörk
Internal Supervisor: Jörgen Blomvall
Examiner: Mathias Henningsson

June 18, 2019
Upphovsrätt
Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under 25 år från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår. Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns lösningar av teknisk och administrativ art. Upphovsmannens ideella rätt innefattar rätt att bli nämnad som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart. För ytterligare information om Linköping University Electronic Press se förlagets hemsida http://www.ep.liu.se/.

Copyright
The publishers will keep this document online on the Internet - or its possible replacement - for a period of 25 years starting from the date of publication barring exceptional circumstances. The online availability of the document implies permanent permission for anyone to read, to download, or to print out single copies for his/hers own use and to use it unchanged for non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional upon the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility. According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement. For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its www home page: http://www.ep.liu.se/.
Abstract

The focus of this master thesis is to develop a model that measures the risk-neutral probability distribution of the future value of a portfolio consisting of options on the S&P 500 index. The cornerstone of the model is an explicit and thorough construction of the local volatility surface. The parametric model of Coleman et al. (1998), with some modifications, is used, representing the local volatility surface through a bicubic spline. The local volatility surface is optimized to be consistent with market data on option prices, futures contracts and Overnight Index Swap (OIS) interest rates. Repricing of options is done through a finite difference method (FDM) approach presented by Andersen and Brotherton-Ratcliffe (1998), using the Crank-Nicholson scheme. An interior point solver is used to minimize the squared pricing error weighted by liquidity in each option. Fast and accurate gradients of the objective function are obtained using the Automatic Differentiation library Autograd.

The local volatility surface is constructed for multiple dates and the systematic changes are analyzed through Principal Component Analysis (PCA) of the logarithmic local variance. A stochastic process is assigned to the local volatility surface using the sensitivities towards systematic changes identified through the PCA. Using a Gaussian Kernel Density Estimator, the probability density function (PDF) of the future value of the portfolio is measured. The method requires simulated portfolio values, which are achieved through FDM pricing using simulations of the local volatility surface and the underlying index. The cumulative distribution function (CDF) is finally computed by integration of the PDF. To evaluate the measured probability distribution, a normal CDF inversion of 106 measured out-of-sample CDF values are compared to theoretical normal distribution quantiles with Q-Q plots.

The constructed local volatility surface is consistent with market prices to an extent where it is more accurate for more liquid options. It is in most cases realistic with respect to smoothness, but have an unexpectedly large offset from the at-the-money strike level in the skew structure. It is unstable from date to date and also significantly dependent on choice of parameters, limited by computational power, and input data. The unstable construction of the local volatility surface results in measurement noise that cause auto correlation in the principal components, which impairs their explanatory ability. The main result show that the shape of the probability distribution is measured accurately, but the standard deviation (or volatility) is overestimated.
Acknowledgements

We would like to thank our internal supervisor Jörgen Blomvall and external supervisor Filip Mörk of Kidbrooke Advisory AB (Kidbrooke) for their input, feedback, and support without which this thesis could not have been completed. Additional mentions are examiner Mathias Henningsson and Sanna Brandel of Kidbrooke who also provided valuable input throughout the thesis.
## Contents

1 **Introduction**  
   1.1 Purpose ................................................................. 1  
   1.2 Delimitations .......................................................... 1  

2 **Method Overview**  
   2.1 Constructing the Local Volatility Surface  
   2.2 Analyzing the Systematic Changes of the Local Volatility Surface  
   2.3 Estimating the Probability Distribution  
   2.4 Evaluating Model Performance ....................................... 2  

3 **Theoretical Background**  
   3.1 Volatility .............................................................. 4  
      3.1.1 Black Scholes and Implied Volatility  
      3.1.2 Stochastic Volatility  
      3.1.3 Local Volatility .................................................. 5  
   3.2 Pricing under Local Volatility ...................................... 5  
      3.2.1 Monte-Carlo Simulation  
      3.2.2 Finite Difference Methods ...................................... 6  
   3.3 Deterministic Risk-Free Interest Rate  
   3.4 Measuring Dividend Structure ...................................... 10  
      3.4.1 Implied Dividend Yield from Future Contracts  
      3.4.2 Implied Dividend Yield from Put and Call Options  
      3.4.3 Forward Dividend Structure .................................... 11  
   3.5 Construct a Local Volatility Surface  
      3.5.1 Interpolation Models  
      3.5.2 Parametric Models ................................................ 12  
      3.5.3 Non-parametric Models .......................................... 13  
   3.6 Spline Parameterization ............................................. 14  
      3.6.1 General Bicubic Interpolation  
      3.6.2 Bicubic Spline ................................................... 15  
   3.7 Interior-point Methods .............................................. 16  
   3.8 Automatic Differentiation .......................................... 17  
      3.8.1 Forward Mode ..................................................... 17  
      3.8.2 Reverse Mode ..................................................... 18  
      3.8.3 Operator Overloading ........................................... 19  
      3.8.4 Source Code Transformation .................................... 19  
   3.9 Stochastic Local Volatility ......................................... 19  
      3.9.1 Fundamental Theorem of Asset Pricing  
      3.9.2 Drift Conditions ................................................ 20  
   3.10 Principal Component Analysis ...................................... 21  
   3.11 Kernel Density Estimator .......................................... 22  
   3.12 Inverse Transform Method and Q-Q Plots  

4 **Method**  
   4.1 Constructing the Local Volatility Surface  
      4.1.1 The Optimization Problem ...................................... 24  
      4.1.2 Parameterization of the Local Volatility Surface  
      4.1.3 Option Pricing by a Finite Difference Method ............... 25  
      4.1.4 Optimization Solver ............................................. 27  
      4.1.5 Automatic Differentiation ...................................... 31  
   4.2 Analyzing the Systematic Changes of the Local Volatility Surface  
      4.2.1 Dynamics of the Local Volatility Surface  
      4.2.2 Principal Component Analysis .................................. 32  
      4.2.3 Drift Conditions ................................................ 35
4.3 Estimating the Probability Distribution ...................................................... 35
  4.3.1 Simulating the Portfolio Value ........................................................... 36
  4.3.2 Kernel Density Estimator ........................................................................ 36
4.4 Evaluating Model Performance ................................................................. 37
  4.4.1 The Portfolio ............................................................................................ 37
  4.4.2 Data ........................................................................................................ 37
  4.4.3 Inverse Transform Method and Q-Q Plots .............................................. 38

5 Results and Analysis ..................................................................................... 39
  5.1 Constructing the Local Volatility Surface ................................................... 39
    5.1.1 Skew and Term Structure ..................................................................... 40
    5.1.2 Pricing Error .......................................................................................... 42
  5.2 Analyzing the Systematic Changes of the Local Volatility Surface ............. 44
  5.3 Model Performance Measuring the Probability Distribution ..................... 50

6 Discussion ..................................................................................................... 53
  6.1 The Local Volatility Surface ....................................................................... 53
  6.2 Systematic Changes of the Local Volatility Surface ................................. 54
  6.3 Accuracy of the Measured Probability Distribution .................................... 55
    6.3.1 Auto Correlation in Principal Components ........................................ 55
    6.3.2 Market Consistent Volatility versus Realized Volatility ..................... 55
  6.4 Implementation and Practical Issues ............................................................ 56
  6.5 Time Limitation ........................................................................................... 56
  6.6 Inadequate Evaluation .................................................................................. 57
  6.7 Ethical Aspects ............................................................................................ 57
  6.8 Suggestions for Future Studies ..................................................................... 57
  6.9 Conclusions ................................................................................................. 58

Appendices ......................................................................................................... a

Appendix A Proof of the No-arbitrage Drift Conditions ..................................... a
### List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Overview of the method.</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>Uniformly spaced grid for finite difference methods.</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Non-uniformly spaced grid for finite difference methods.</td>
<td>27</td>
</tr>
<tr>
<td>4</td>
<td>Non-uniformly spaced grid for finite difference methods, with boundary conditions.</td>
<td>30</td>
</tr>
<tr>
<td>5</td>
<td>Local volatility for 2011-01-14, a &quot;good&quot; date.</td>
<td>39</td>
</tr>
<tr>
<td>6</td>
<td>Local volatility for 2011-01-20, a &quot;bad&quot; date.</td>
<td>40</td>
</tr>
<tr>
<td>7</td>
<td>Local volatility skews for different maturities, &quot;good&quot; vs. &quot;bad&quot; date.</td>
<td>41</td>
</tr>
<tr>
<td>8</td>
<td>Local volatility skew and strike knots placement at moneyness levels [0.6 0.93 1.26 1.6].</td>
<td>41</td>
</tr>
<tr>
<td>9</td>
<td>Term structure for different levels of moneyness, &quot;good&quot; vs. &quot;bad&quot; date.</td>
<td>42</td>
</tr>
<tr>
<td>10</td>
<td>Pricing error expressed as difference in implied volatility (observed IV - repricing IV), &quot;good&quot; vs. &quot;bad&quot; date.</td>
<td>43</td>
</tr>
<tr>
<td>11</td>
<td>Local volatility for 2011-01-24, i.e. two days after the &quot;bad&quot; date.</td>
<td>43</td>
</tr>
<tr>
<td>12</td>
<td>Pricing error expressed as difference in implied volatility (observed IV - repricing IV) 2011-01-24, i.e. two days after the &quot;bad&quot; date.</td>
<td>44</td>
</tr>
<tr>
<td>13</td>
<td>Plot of the first eigenvector as a surface.</td>
<td>46</td>
</tr>
<tr>
<td>14</td>
<td>Plot of the second eigenvector as a surface.</td>
<td>46</td>
</tr>
<tr>
<td>15</td>
<td>Plot of the third eigenvector as a surface.</td>
<td>47</td>
</tr>
<tr>
<td>16</td>
<td>Principal components 1-3 over the calibration period.</td>
<td>48</td>
</tr>
<tr>
<td>17</td>
<td>Principal components 4-7 over the calibration period.</td>
<td>48</td>
</tr>
<tr>
<td>18</td>
<td>Constructed local volatility surface for six adjacent dates.</td>
<td>50</td>
</tr>
<tr>
<td>19</td>
<td>Q-Q plots for the measured probability distribution for $\Delta t = \frac{1}{252}$ and $\Delta t = \frac{2}{252}$.</td>
<td>51</td>
</tr>
<tr>
<td>20</td>
<td>Q-Q plots for the measured probability distribution for $\Delta t = \frac{3}{252}$ and $\Delta t = \frac{4}{252}$.</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>Q-Q plots for the measured probability distribution for $\Delta t = \frac{5}{252}$.</td>
<td>52</td>
</tr>
<tr>
<td>22</td>
<td>Example of error in measurement.</td>
<td>55</td>
</tr>
</tbody>
</table>
List of Tables

1. Name of constituent lists for European call and put options. ........................................... 37
2. Name of constituent lists for future contracts. ................................................................. 37
3. Examples of pricing errors in USD instead of Implicit Volatility, compare these with Figure 10 on the left. ................................................................. 43
4. Fraction of explained variance for each principal component. ........................................... 44
5. The seven largest eigenvalues and corresponding eigenvectors. ........................................... 45
6. Principal components’ auto correlation for different lags ..................................................... 49
1 Introduction

In March 2010, the Swedish investment bank HQ Bank estimated the market risk of their portfolio of equity index options to 33 million SEK. Three months later, on the 28th of June, they closed their last positions with a total loss of 1.23 billion SEK. The collapse of HQ Bank was a result of systematic errors in valuating their positions and quantifying the risk. More specifically, they used unsatisfactory assumptions about the underlying index volatility when observable market prices weren’t available for options with a long time to maturity. Thus, they underestimated the market risk and their losses heavily. Even though no one was convicted in the trials following the collapse, one can’t rule out that fraudulent behaviour might have played a part as well. Nonetheless, this is an example of why it is important with sophisticated and realistic models for pricing and measuring risk associated with financial derivatives.

The most used tool for pricing options is the Black-Scholes model (Black and Scholes 1973), which assumes a constant volatility over time. It is well known and repeatedly shown that this assumption is unrealistic and leads to risk measures that are inconsistent with the market. To counteract this flawed model, implied volatility models have been used for some time by practitioners to model the risk. However, implied volatility models have been criticized as they do not account for the time variation of volatility (Alexander and Nogueira 2004). In order to model the dynamic movement of the volatility, models for stochastic volatility were introduced (see e.g. Heston 1993, Bates 1996). A weakness with such models is that they introduce an additional diffusion process making the market model incomplete, meaning that a derivative can’t be hedged using only the underlying asset, i.e. another derivative is needed. A complete market model is desirable since it provides the opportunity to find a unambiguously arbitrage-free price of each derivative. Dupire (1994) noted this issue and presented a market model that maintained the completeness property and accounted for the time variation of volatility, the local volatility model. Under local volatility, the volatility of the underlying asset is assumed to be a locally deterministic function of the asset price and time to maturity, which can be visualized as a local volatility surface.

Local volatility surfaces that are consistent with the market are easy to construct (Coleman et al. 1998), since the observable market data is finite and thus there typically exists an infinite number of solutions. However, constructing surfaces that are realistic is not straightforward and several methods have been suggested (see e.g. Derman et al. 1996a, Coleman et al. 1998, Lagnado and Osher 1997, Geng et al. 2014). Aside from complexity, another problem is that the local volatility surface is measured using the information that is known today and have to be reconstructed every day, as new information becomes available. This poses a problem if one might be interested in knowing what the local volatility surface will look like at some point in the future. For example, this is necessary in order to get a good estimation of the probability distribution for the future value of a trading portfolio consisting of options. A forward-looking local volatility model for pricing and risk management would undoubtedly have been useful for HQ Bank in 2010. Simply using the local volatility surface observed today, assuming that it will stay static in the future, will underestimate the forward volatility skew and result in faulty forward pricing, hedge ratios and risk metrics.

To predict how the surface will evolve in the future, the dynamics have to be examined, measured and modelled in a way that prohibits arbitrage and is consistent with observable market prices. Derman and Kani (1997) presented a general method for modelling the time variation of local volatility to maintain the no arbitrage property of the forward surface. Their no-arbitrage conditions are similar to those of Heath et al. (1992) for stochastic interest rate term structure. We propose a method to analyze independent factors that compose the systematic changes of the local volatility surface and incorporate them to Derman and Kani’s (1997) model.

1.1 Purpose

The purpose of this thesis is to measure the risk-neutral probability distribution for the future value of a portfolio consisting of equity index options, using Derman and Kani’s (1997) local volatility model.

1.2 Delimitations

The probability distribution will be measured for a portfolio consisting of S&P 500 index options. No other markets will be studied.


2 Method Overview

The method for achieving the purpose of this thesis, stated in Section 1.1, is divided into four main components, Constructing the Local Volatility Surface, Analyzing the Systematic Changes of the Local Volatility Surface, Estimating the Probability Distribution and Evaluating Model Performance. The method overview and how the different components relate to each other is illustrated in Figure 1 and described in more detail next.

2.1 Constructing the Local Volatility Surface

The first component aims to construct the local volatility surface for the S&P 500 index. It is constructed for historical dates with observable market prices of options, futures contracts and OIS interest rates as input data. We choose the parametric model of Coleman et al. (1998), which applies a bicubic spline to the local volatility surface and minimize the square pricing error with respect to observable option prices, but we add some additional constraints to the original model. The repricing of options is done with a Finite Difference Method (FDM) approach originally presented by Andersen and Brotherton-Ratcliffe (1998). We use an interior-point optimization solver to solve this large non-linear optimization problem and apply Automatic Differentiation (AD) to the FDM option pricing to obtain the needed gradients. The output from this component is the constructed local volatility surface for a set of historical dates.

2.2 Analyzing the Systematic Changes of the Local Volatility Surface

The constructed local volatility surface for historical dates is the input to the second component. This component aims to analyze systematic changes in the local volatility surface in order to assign a stochastic process that accurately and realistically describes its changes over time. We model the dynamics of the local volatility surface using Derman and Kani’s (1997) local volatility model. The independent factors that compose the local volatility surface’s systematic changes are identified and decreased to a suitable dimension using Principal Component Analysis, see e.g. Jolliffe (2011), of the logarithmic local variance. With the given independent factors, the dynamics are derived through comparison to Derman and Kani’s (1997) model. We stay at providing a theoretical description of how no-arbitrage conditions are derived and applied, but simplify by using a drift of zero in our model for the dynamics of the local volatility surface.

2.3 Estimating the Probability Distribution

The stochastic process for the local volatility surface works as input to the third component. This component aims to measure the probability density function (PDF) and the cumulative distribution function (CDF) for the future value of a portfolio consisting of equity index options. The local volatility surface, as well as the underlying index, is simulated and the portfolio is priced in accordance with the simulation result, using the FDM approach. The procedure is repeated to obtain a set of simulated future portfolio values. The PDF is then measured using a Gaussian kernel density estimator and the CDF by integration of the PDF. The output from the third component is a method for measuring the sought probability distribution.

2.4 Evaluating Model Performance

The last component takes the resulting model as input, as it aims to evaluate its performance. The performance is evaluated out-of-sample by using the Inverse Transform Method and studying Quantile-Quantile (Q-Q) plots.
Observable market prices of options, futures and OIS interest rates

1. Constructing the Local Volatility Surface
   - Spline parameterization
   - Minimizing square pricing error
   - Finite Difference Method
   - Interior-point solver
   - Automatic Differentiation

2. Analyzing the Systematic Changes of the Local Volatility Surface
   - Model dynamics, Derman and Kani (1997)
   - Principal Component Analysis
   - No-arbitrage conditions

3. Estimating the Probability Distribution
   - Simulation
   - Gaussian kernel density estimator
   - Integration

4. Evaluating Model Performance
   - Inverse Transform Method
   - Out-of-sample Q-Q plots

$PDF$ and $CDF$ of portfolio value

Figure 1: Overview of the method. Italic text represents input/output data between the method components.
3 Theoretical Background

This section presents the theoretical background necessary for choosing a proper method with respect to achieving the purpose of the thesis.

3.1 Volatility

Modelling volatility is central in the purpose of the thesis. In this section, theoretical background regarding volatility and some different ways of measuring volatility is presented. The volatility $\sigma$ of an asset $S$ is a measure of the uncertainty about the returns of the asset. It is defined as the standard deviation of the asset’s return over a one-year period (Hull 2013). It is an easy and intuitive way of expressing the risk associated with owning the asset. As it turns out, there are different ways of measuring volatility from the market. A couple of examples are implied volatility, stochastic volatility and local volatility.

3.1.1 Black Scholes and Implied Volatility

Black and Scholes’ (1973) popular model for pricing European style plain vanilla options assumes a market model where the underlying asset $S$ follows a Geometric Brownian Motion,

$$\frac{dS_t}{S_t} = \mu dt + \sigma dB_t,$$

where $\mu$ is the drift, $\sigma$ the constant volatility and $B_t$ the standard Weiner process. Modelling $S$ under the risk-neutral measure $Q$, with drift $\mu = r - q$, where $r$ is the constant risk-free interest rate and $q$ is the constant dividend yield, they showed that the price, $C_t$, at time $t$, of a European style call option with strike price $K$ and time of maturity $T$, is given by

$$C_t(S_t, r, q, K, T, \sigma) = S_t e^{-q(T-t)} N_0(d_1) - Ke^{-r(T-t)} N_0(d_2),$$

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

$$d_2 = d_1 - \sigma \sqrt{T-t},$$

where $N_0(x)$ is the standard normal cumulative probability distribution function.

According to the proposed model, there is an explicit price of an option that depends on the parameters $S_t$, $r$, $q$, $K$, $T$ and $\sigma$, of which $\sigma$ is the only parameter that cannot be observed in the market. In practice, traders usually work with implied volatilities, which are the volatilities that price options in accordance with (3.2) and observed option prices for various strikes and maturities (ibid.). Thus, the implied volatility, $\sigma_{imp}$, is determined by solving the inverse equation,

$$C_t(S_t, r, q, K, T, \sigma_{imp}) = C_{t}^{obs},$$

where $C_{t}^{obs}$ is the observed market price of a call option.

Prior to the market crash in 1987, equity options were priced with the assumption that the implied volatility of an underlying asset should be the same regardless of strike price or maturity. The market crash showed that Black and Scholes’ (1973) assumption of log-normally distributed asset prices was invalid, because it didn’t represent the extreme market movements that were possible. Since then, implied volatilities for equity and equity index options show a volatility skew, where the volatility is higher if derived from the price of a low strike price option (deep-out-of-the-money put or deep-in-the-money-call) than if derived from a high strike price option (ibid.).

3.1.2 Stochastic Volatility

Latané and Rendleman (1976) noted that Black and Scholes’ (1973) assumption of constant volatility is unreasonable since different implied volatilities are required for different strike prices and maturities to repricce options in line with observed market prices. An alternative method is to use stochastic volatility, such as the models presented by Heston (1993) or Bates (1996), to allow the volatility to vary stochastically. Bates’ (1996) Stochastic Volatility Jump Diffusion (SVJD) model was introduced to capture both the skewness apparent in the option’s data and the leptokurtic distributions of the underlying indexes. The model can be written under
the risk-neutral measure as
\[ \frac{dS_t}{S_t} = (r - q - \lambda \mu_j)dt + \sqrt{\nu_t}dW^1_t + J_t dN_t, \]  
(3.6)
\[ dv_t = \kappa(\eta - \nu_t)dt + \theta \sqrt{\nu_t}dW^2_t, \]  
(3.7)
\[ \text{cov}(dW^1_t, dW^2_t) = \rho dt, \]  
(3.8)
where \( \lambda \) is the jump frequency of the Poisson process \( N \), which is independent of the Wiener processes \( W^1 \) and \( W^2 \), \( \mu_j \) is the unconditional mean for the adapted process \( J_t \) with
\[ \ln(1 + J_t) \sim \mathcal{N} \left( \ln(1 + \mu_j) - \frac{\sigma^2_j}{2}, \sigma^2_j \right), \]  
(3.9)
\( \nu_0 \) is the starting variance, \( \kappa \) is the rate with which the variance reverts to the level \( \eta \), \( \theta \) is the volatility of volatility and \( \rho \) is the correlation between the Wiener processes. A disadvantage with stochastic volatility models is that the completeness of the market model is lost since additional diffusion processes are introduced. Completeness is of high value since it allows for arbitrage pricing and hedging (Dupire 1994).

### 3.1.3 Local Volatility

The local volatility, initially introduced by Dupire (ibid.), is a locally deterministic function, \( \sigma(K, t) \), depending on strike price, \( K \), and time, \( t \). Local volatility is instantaneous, which means that the volatility \( \sigma(K, t) \) is deterministic for the infinitesimal time step \( dt \), in contrast to implied volatility, which intuitively can be seen as the average volatility of the underlying asset from the current time to the maturity of the option (Derman et al. 1996b). The local volatility function is easier to fit to observable market prices of plain vanilla options than stochastic volatility models (Gatheral and Taleb 2011) and it keeps the market model complete. Using local volatility, the continuous process for the underlying asset can be written as
\[ \frac{dS_t}{S_t} = \mu dt + \sigma(S_t, t) dZ_t. \]  
(3.10)

Gyöngy (1986) showed, unrelated to financial valuation methodology, that for a general Geometric Brownian Motion with a process for stochastic volatility, there exists a locally deterministic volatility with identical probability distribution. The relationship between the local volatility \( \sigma(K, t) \) and the stochastic volatility \( \sigma_t = \sqrt{\nu_t} \) is
\[ \sigma^2(K, t) = \mathbb{E} \left[ \sigma^2_t | S_t = K \right]. \]  
(3.11)
Hence, the underlying asset price can be modelled under the risk-neutral measure, \( Q \), according to
\[ \frac{dS_t}{S_t} = (r - q)dt + \sigma(S_t, t) dZ^Q_t. \]  
(3.12)
Solving (3.12) gives
\[ S_T \approx S_t \exp \left( \left( r - q - \frac{\sigma^2(S_t, t)}{2} \right) (T - t) + \sigma(S_t, t) \sqrt{T-t} \xi_t \right), \quad \xi_t \sim \mathcal{N}(0, 1) \text{ i.i.d.}, \]  
(3.13)
if \( T - t \) is small (since local volatility is instantaneous).

### 3.2 Pricing under Local Volatility

When constructing a local volatility surface, one of the objectives is to make it consistent with observable option prices. Thus, one needs a method for pricing options under local volatility. In continuous time, assuming a continuous spectrum of available option prices for all maturities and strike prices, there are analytical solutions for pricing an option under local volatility. However, in practice numerical methods have to be implemented to price a specific option. (Andersen and Brotherton-Ratcliffe 1998)

For a complete market model, such as the local volatility model, a partial differential equation (PDE) that an option price needs to satisfy to prohibit arbitrage can be derived. Consider the price of a European style call
option, \( C_t = C(t, S_t) \), with the underlying asset \( S_t \) following the GBM in (3.10). Further, let \( q \) be the continuous dividend yield of \( S \). Assume that the accumulated dividend until time \( t \) follows the process
\[
dD_t = q S_t dt.
\]

Let the portfolio \( V_p \) consist of a short position of one option and \( x_s \) of the underlying asset. We get an expression for the portfolio value,
\[
V_p = -C_t + x_s S_t,
\]
and an expression for the wealth value,
\[
V_w = -C_t + x_s (S_t + D_t).
\]
The dynamics of \( V_w \) is
\[
dV_w = -dC_t + x_s (dS_t + dD_t).
\]
If \( dC_t \) is expanded using Itô’s Lemma and both \( dS_t \) and \( dD_t \) is substituted with (3.10) and (3.14) respectively, we get
\[
dV_w = -\left( \frac{\partial C_t}{\partial t} + \mu S_t \frac{\partial C_t}{\partial S_t} - x_s \right) - x_s q S_t + \frac{1}{2} \sigma^2 (S_t, t) S_t^2 \frac{\partial^2 C_t}{\partial S_t^2} \right) dt - \sigma (S_t, t) S_t \left( \frac{\partial C_t}{\partial S_t} - x_s \right) dZ_t.
\]

By choosing \( x_s = \frac{\partial C_t}{\partial S_t} \), the Weiner process is eliminated and the wealth is deterministic and risk-free. More commonly, this is known as Delta hedging. We have
\[
dV_w = - \left( \frac{\partial C_t}{\partial t} - q S_t \frac{\partial C_t}{\partial S_t} + \frac{1}{2} \sigma^2 (S_t, t) S_t^2 \frac{\partial^2 C_t}{\partial S_t^2} \right) dt.
\]
If the wealth is risk-free, then
\[
dV_w = r V_w dt
\]
must hold, where \( r \) is the risk-free interest rate, assumed constant for now. Using (3.20), (3.19) and (3.15) together gives
\[
\begin{align*}
\frac{\partial C_t}{\partial t} + (r - q) S_t \frac{\partial C_t}{\partial S_t} + \frac{1}{2} \sigma^2 (S_t, t) S_t^2 \frac{\partial^2 C_t}{\partial S_t^2} &= r C_t, \\
C_T &= \Phi(S_T).
\end{align*}
\]
The partial differential equation, (3.21), is an extension of the famous Black Scholes Merton Partial Differential Equation (BSM PDE), which was originally derived under constant volatility and without dividend yield. The PDE can be solved in different ways. Two popular approaches to price options are by Monte-Carlo simulation and Finite Difference Methods.

### 3.2.1 Monte-Carlo Simulation

By applying an extension of the Feynman-Kac formula to the PDE in (3.21), it follows that the option value gets the stochastic representation
\[
C_t = e^{-r(T-t)} E_t^Q [\Phi(S_T)],
\]
where
\[
dS_t = (r - q) S_t dt + \sigma (S_t, t) dZ_t^Q.
\]
This is more commonly known as risk-neutral valuation. \( E_t^Q [\Phi(S_T)] \) denotes the expectation of the payoff function at time \( T \) conditioned to the information known at time \( t \), where \( S \) is modelled under the risk-neutral probability measure \( Q \). The expectation can thus be approximated by Monte-Carlo simulation as
\[
E_t^Q [\Phi(S_T)] \approx \frac{1}{N_{sim}} \sum_{i=1}^{N_{sim}} \Phi(S_{T,i}),
\]
where \( N_{sim} \) is the number of simulated Monte-Carlo paths and \( S_{T,i} \) is the simulated value of \( S \) at time \( T \) according to path \( i \). The solution to the SDE (3.23) is
\[
S_{t+\Delta t} = S_t \exp \left( \left( r - q - \frac{\sigma^2 (S_t, t)}{2} \right) \Delta t + \sigma (S_t, t) \sqrt{\Delta t} \xi_t \right), \quad \xi_t \sim \mathcal{N}(0,1) \text{ i.i.d.},
\]
and Monte-Carlo paths for \( S \) can be generated accordingly. The time step \( \Delta t \) needs to be sufficiently small since the local volatility \( \sigma(S_t, t) \) is instantaneous.
3.2.2 Finite Difference Methods

Finite Difference Methods values an option by solving (3.21) for a set of plausible discretized values of the underlying asset and time defined as a grid (Hull 2013). The value of an option can be determined in the edges of the grid and then iteratively backwards in time through the rest of the grid by solving a tridiagonal equation system. The approach in Hull (ibid.) is intuitive and clear, however, using a backward recursive discretization scheme, one can only price one option per solution of the PDE. This thesis requires pricing of options over a large spectrum of strikes, and maturities for a large number of dates, making Hull’s (2013) approach impractical. Andersen and Brotherton-Ratcliffe (1998) present an approach where they first generalize the backwards induction to a compact matrix form and then use fundamental arguments to derive a discrete-time forward induction that is consistent with the backwards induction. In that way, options with all strikes and maturities can be priced in one “sweep”, decreasing computational time tremendously. The methodology is explained next.

We perform the substitutions $x_t = \ln S_t$ and $H(x_t, t) = C(t, S_t)$ in (3.21). With $H_t = H(x_t, t)$ the PDE becomes

$$\frac{\partial H_t}{\partial t} + b(x_t, t) \frac{\partial H_t}{\partial x_t} + \frac{1}{2} v(x_t, t) \frac{\partial^2 H_t}{\partial x_t^2} = rH_t, \quad (3.26)$$

where $b(x, t) = r - q - \frac{1}{2} v(x, t)$ and $v(x_t, t) = \sigma^2(S_t, t) = \sigma^2(e^{x_t}, t)$. The plane $(x, t)$ is divided into a uniformly spaced grid with $M + 2$ points along the $t$ axis and $N + 2$ along the $x$ axis, generating the points

$$x_i = x_0 + i\Delta x = x_0 + i\frac{x_{N+1} - x_0}{N+1}, \quad i = 0, \ldots, N + 1, \quad (3.27)$$

$$t_j = j\Delta t = j\frac{T}{M+1}, \quad j = 0, \ldots, M + 1. \quad (3.28)$$

The grid is illustrated in Figure 2.

![Logarithmic asset price (i) and Time (j)](Figure 2: Uniformly spaced grid for finite difference methods. Blue nodes represent the edges, where end conditions need to be set.)

On the edges of the grid, i.e. for $i = 0, N + 1$ and $j = 0, M + 1$, boundary conditions are either known or must be set. The values $x_0$ and $x_{N+1}$ should be set so that the grid captures the statistically significant $x$ space. This entails an assumption that the initial stock price $x_{ini} = \ln S_{ini}$ is contained in the grid. The authors point out that the equidistance is not by any means proven to be optimal. For any node $(x_i, t_j)$, with $i = 1, \ldots, N$ and
\(j = 0, \ldots, M\), i.e. not on the edge of the grid, difference approximations can be made for the partial derivatives in (3.26). Using the notation \(H_{i,j} = H(x_i, t_j)\), we get

\[
\begin{align*}
\frac{\partial H_t}{\partial t} &\approx \frac{H_{i,j+1} - H_{i,j}}{\Delta t} \\
\frac{\partial H_t}{\partial x_i} &\approx (1 - \Theta) \frac{H_{i+1,j} - H_{i-1,j}}{2\Delta x} + \Theta \frac{H_{i+1,j+1} - H_{i-1,j+1}}{2\Delta x} \\
\frac{\partial^2 H_t}{\partial x_i^2} &\approx (1 - \Theta) \frac{H_{i+1,j} - 2H_{i,j} + H_{i-1,j}}{(\Delta x)^2} + \Theta \frac{H_{i+1,j+1} - 2H_{i,j+1} + H_{i-1,j+1}}{(\Delta x)^2},
\end{align*}
\]

where \(\Theta \in \{0, 0.5, 1\}\) corresponds to using the \textit{implicit}, \textit{explicit} and \textit{Crank-Nicholson} approach respectively. Other values of \(\Theta\) are possible but barely used in practice. Substituting the difference approximations into (3.21) gives the recursive relation

\[
H_{i-1,j} \left( -\frac{1}{2} \alpha (1 - \Theta)(v_{i,j} - \Delta x b_{i,j}) \right) + H_{i,j} \left( 1 + r \Delta t + \alpha (1 - \Theta)v_{i,j} \right)
+ H_{i+1,j} \left( -\frac{1}{2} \alpha (1 - \Theta)(v_{i,j} + \Delta x b_{i,j}) \right)
= H_{i-1,j+1} \left( \frac{1}{2} \alpha \Theta(v_{i,j} - \Delta x b_{i,j}) \right) + H_{i,j+1} \left( 1 - \alpha \Theta v_{i,j} \right) + H_{i+1,j+1} \left( \frac{1}{2} \alpha \Theta(v_{i,j} + \Delta x b_{i,j}) \right),
\]

\(i = 1, \ldots, N, j = 0, \ldots, M,\) (3.32)

where \(\alpha = \Delta t/(\Delta x)^2\). This can be written more compactly in matrix form as

\[
[(1 + r \Delta t)I - (1 - \Theta)M_j]H_j = (\Theta M_j + I)H_{j+1} + B_j,
\]

\(j = 0, \ldots, M,\) (3.33)

where \(I\) is the \(N \times N\) identity matrix, \(H_j\) is a vector,

\[
H_j = \begin{bmatrix} H_{1,j} \\ H_{2,j} \\ \vdots \\ H_{N,j} \end{bmatrix},
\]

(3.34)

\(B_j\) is a vector containing the prescribed values of \(H_j\) along the upper and lower edges of the grid,

\[
B_j = \begin{bmatrix} l_{1,j}((1 - \Theta)H_{0,j} + \Theta H_{0,j+1}) \\ 0 \\ \vdots \\ 0 \\ u_{N,j}((1 - \Theta)H_{N+1,j} + \Theta H_{N+1,j+1}) \end{bmatrix},
\]

(3.35)

and \(M_j\) a tridiagonal matrix,

\[
M_j = \begin{bmatrix} c_{1,j} & u_{1,j} & 0 & 0 & 0 & \cdots & 0 \\ l_{2,j} & c_{2,j} & u_{1,j} & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & l_{N-1,j} & c_{N-1,j} & u_{N-1,j} \\ 0 & 0 & 0 & \cdots & l_{N,j} & c_{N,j} & \end{bmatrix},
\]

(3.36)

where

\[
c_{i,j} = -\alpha v_{i,j},
\]

(3.37)

\[
u_{i,j} = \frac{1}{2} \alpha (v_{i,j} + \Delta x b_{i,j}),
\]

(3.38)

\[
l_{i,j} = \frac{1}{2} \alpha (v_{i,j} - \Delta x b_{i,j}).
\]

(3.39)

If the matrices in (3.33) can be determined, the price of an option can be obtained by iteratively solving the system of linear equations \textit{backwards} from time of maturity \(T\), since the payout vector \(H_{N+1}\) is known. Andersen and Brotherton-Ratcliffe (1998) derive sufficient conditions for (3.33) to have a unique solution and
further states that most realistic grids will satisfy those conditions. To derive the forward equation consistently with the backwards scheme (3.33), a couple of new concepts need to be introduced.

Up until this point we have used constant risk-free interest rate \( r \) and dividend yield \( q \). From now on we consider deterministic risk-free short rate \( r(t) \), as explained in more detail in Section 3.3, and dividend yield \( q(t) \), as explained in Section 3.4. We also introduce the concept of Arrow-Debreu securities. Let \( A_{k,l}^{i,j} (l \geq j) \) denote the price at node \((x_i,t_j)\) of the Arrow-Debreu security that pays out 1$ at \( t_l \) if and only if node \((x_k,t_l)\) is reached. Also, define \( d_l^j \) as the discount factor with maturity \( t_j \) as observed at time \( t_l \). In terms of Arrow-Debreu prices, (3.33) becomes

\[
\left( \frac{d_j^0}{d_{j+1}^0} I - (1 - \Theta)M_j \right) A_{k,j}^{k,j+1} = (\Theta M_j + I) A_{k,j+1}^{k,j+1} + B_j^k, \quad j = 0, \ldots, M, \quad k = 1, \ldots, N,
\]

where

\[
A_{j+1}^{k,j+1} = \begin{bmatrix}
A_{1,j+1}^{k,j+1} \\
\vdots \\
A_{N,j}^{k,j+1}
\end{bmatrix}, \quad B_j^k = \begin{bmatrix}
l_{1,j} \left( (1 - \Theta)A_{0,j}^{k,j+1} + \Theta A_{0,j+1}^{k,j+1} \right) \\
0 \\
\vdots \\
0 \\
u_{N,j} \left( (1 - \Theta)A_{N+1,j}^{k,j+1} + \Theta A_{N+1,j+1}^{k,j+1} \right)
\end{bmatrix},
\]

and

\[
A_{j+1}^{k,j+1} = \begin{bmatrix}
0 \\
0 \\
\vdots \\
1 \\
\vdots \\
0
\end{bmatrix},
\]

with the 1 at the \( k \):th row, due to the definition of Arrow-Debreu securities. If the grid spans a sufficient part of the relevant \( x \) space, the effect of \( B_j^k \) is generally negligible and any reasonable assumption on the local boundary behaviour will suffice. Andersen and Brotherton-Ratliff (1998) assumes that both upper and lower boundaries are absorbing, i.e. that

\[
A_{0,j}^{k,j} = \begin{cases}
\frac{d_j^0}{d_{j+1}^0}, & \text{if } k = 0 \\
0, & \text{if } k = 1, \ldots, N + 1
\end{cases}, \quad j = 0, \ldots, M,
\]

\[
A_{N+1,j}^{k,j} = \begin{cases}
\frac{d_j^0}{d_{j+1}^0}, & \text{if } k = N + 1 \\
0, & \text{if } k = 0, \ldots, N
\end{cases}, \quad j = 0, \ldots, M,
\]

for \( l \geq j \), resulting in \( B_j^k = 0 \) for \( k = 1, \ldots, N \). With these assumptions they show that a recursive relation in the initial Arrow-Debreu prices can be derived.

\[
\left( A_{ini}^{j+1} \right)^T = -\frac{\Theta}{1 - \Theta} \left( A_{ini}^{j} \right)^T + \left( A_{ini}^{j} \right)^T \left( \frac{d_j^0}{d_{j+1}^0} I - (1 - \Theta)M_j \right)^{-1} \left( I - \frac{\Theta d_j^0}{d_{j+1}^0} + (1 - \Theta) \right) , \quad j = 0, \ldots, M
\]

is a forward recursive formula, where

\[
A_{ini}^j = \begin{bmatrix}
A_{ini,1,j}^{1,j} \\
A_{ini,2,j}^{1,j} \\
\vdots \\
A_{ini,N,j}^{1,j}
\end{bmatrix},
\]
and \( A^i_{ini} \) is the price at node \((x = x_{ini}, t = 0)\) of the Arrow-Debreu security that pays out 1$ at \(t_j\) if and only if node \((x_i, t_j)\) is reached. By definition

\[
A^0_{ini} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix},
\]

i.e. zero for all elements that aren’t corresponding to the actual asset price at \(t = 0\), and (3.45) can be solved in a forward manner. When prices for Arrow-Debreu securities have been decided for the grid in Figure 2, the price \(C^{i,j}\) of a call option with strike \(K = S_i = e^{x_i}\) and maturity \(t_j\) can be decided through

\[
C^{i,j} = \sum_{l=i+1}^{N+1} A^l_{ini}(S_l - S_i).
\]

Using (3.45) and (3.48), as opposed to solving (3.45) for each option, saves a lot of time if one wishes to price a large number of options with varying strike and maturity.

When it comes to the choice of \(\Theta\), Andersen and Brotherton-Ratcliffe (1998) states that the explicit approach \((\Theta = 1)\) is the computationally fastest procedure but that it has bad convergence properties. The implicit and Crank Nicholson approach \((\Theta = 0.5, 0)\) has better convergence properties and \(\theta = 0.5\) is generally recommended.

### 3.3 Deterministic Risk-Free Interest Rate

When pricing derivatives, one needs to decide how to calculate the time-value of money or, in other words, how to discount future payoffs. It is popular to use a constant risk-free interest rate, see e.g. Black and Scholes (1973), which implies that a discount factor would be calculated as

\[
d(t, T) = \exp(-r(T - t)),
\]

where \(r\) is the continuously compounded risk-free interest rate, assumed to be constant. A more realistic approach would be to use a non-constant but deterministic interest rate, by measuring the forward interest rate. Then a discount factor is calculated as

\[
d(t, T) = \exp\left(-\int_t^T f(s)ds\right),
\]

where \(f(t)\) is the instantaneous forward rate at time \(t\). It follows that, if assumed deterministic, the continuously compounded risk-free rate for any spot time \(t\) with maturity \(\tilde{T}\), \(r_t(\tilde{T})\), can be written

\[
r_t(\tilde{T}) = \frac{1}{\tilde{T} - t} \int_t^{\tilde{T}} f(s)ds.
\]

The forward rate can be measured from observable market prices for interest rate derivatives such as zero coupon treasury bonds or Overnight Index Swaps. (Hull 2013)

### 3.4 Measuring Dividend Structure

In order to accurately price contingent claims on an equity index according to prevailing markets, one has to determine the dividend term structure. Even though the cash income from dividends on a stock generally varies week by week, it is usually assumed that the dividends provide a known yield rate. The forward dividend rate of an underlying equity index can be measured from observable market prices for contingent claims on the index, e.g. future contracts and options.
3.4.1 Implied Dividend Yield from Future Contracts

Let \( q_t(T) \) be the annualized dividend yield rate during the life of a future contract on an equity index \( S_t \) with maturity \( T \), then Hull (2013) shows that the future price at time \( t \) is

\[
F_t(T, S_t) = S_t e^{(r_t(T) - q_t(T))(T - t)},
\]

where \( r_t(T) \) is the continuously compounded risk-free interest rate from \( t \) to \( T \). Solving for \( q_t(T) \) gives

\[
q_t(T) = r_t(T) + \frac{\ln \left( \frac{S_t}{F_t(T, S_t)} \right)}{T - t}
\]

and the implied yearly dividend rate, which applies from \( t \) to \( T \), is obtained.

3.4.2 Implied Dividend Yield from Put and Call Options

The put-call parity for options on a stock paying a dividend yield rate, \( q_t \), can be expressed (ibid.) through

\[
C_t(T, S_t) + Ke^{-r_t(T)(T - t)} = P_t(T, S_t) + S_t e^{-q_t(T)(T - t)},
\]

where \( C_t \) and \( P_t \) are the prices at \( t \) of a call and put option on \( S_t \) with maturity at \( T \) and strike price \( K \). The equation can be rearranged to yield the dividend rate,

\[
q_t(T) = \frac{1}{T - t} \ln \left( \frac{C_t(T, S_t) - P_t(T, S_t) + Ke^{-r_t(T)(T - t)}}{S_t} \right).
\]

3.4.3 Forward Dividend Structure

With the implied dividend for different maturities obtained from futures contracts and/or ATM options, one can derive the continuously compounded forward dividend rate, \( q^f_{\tilde{T}}(T) \). The following relation gives \( q^f_{\tilde{T}}(T) \) at a future time \( \tilde{T} \),

\[
q_t(T)(T - t) = q_t(\tilde{T})(\tilde{T} - t) + q^f_{\tilde{T}}(T)(T - \tilde{T})
\]

\( \iff \)

\[
q^f_{\tilde{T}}(T) = \frac{q_t(T)(T - t) - q_t(\tilde{T})(\tilde{T} - t)}{T - \tilde{T}}
\]

given that \( t \leq \tilde{T} \leq T \).

3.5 Construct a Local Volatility Surface

As stated in the purpose, Derman and Kani’s (1997) model for local volatility will be used. That includes studying the time variation of the local volatility surface for historical dates. To do that, one obviously needs to construct the surface for said dates. The objective is to fit the local volatility surface to observed option prices and obtain a realistic result. Important to note is that is has been shown by for example Bakshi and Kapadia (2003) that the market typically experience a negative volatility risk premium, meaning that the realized volatility often is lower than the market consistent volatility. Nonetheless, a local volatility surface is said to be realistic if it has certain desired properties. These properties are for example the skew structure, term structure and smoothness (Geng et al. 2014). It is fairly easy to construct a local volatility surface that is consistent with observable option prices (Coleman et al. 1998), since the set of observable option prices is finite and thus there typically is an infinite number of solutions. However, constructing a local volatility surface that is realistic is not straightforward.

From Dupire’s (1994) fundamental result, the local volatility can be expressed as

\[
\sigma^2(K, T) = 2 \frac{\partial C}{\partial T} + q(T)C + K(r(T) - q(T)) \frac{\partial C}{\partial K} \frac{\partial C}{\partial K},
\]

which assumes continuous time and available option prices for an infinite continuum of strike prices and maturities, which obviously isn’t the case. To construct a local volatility surface in the real, discrete, world at any instant, the following information is needed (Derman et al. 1996b):
• the current value of the underlying asset
• the current risk-free yield curve
• information about future dividend
• current prices of liquid standard options for a range of strikes and maturities.

The methods proposed in the literature for constructing local volatility surfaces can be divided into interpolation, parametric and non-parametric models.

3.5.1 Interpolation Models

One approach is to use interpolation and extrapolation of observable option prices to obtain sufficient input to (3.58). This method was initially presented by Dupire (1994) himself and built upon by Derman and Kani who first used binomial trees (Derman and Kani 1994a) and later trinomial trees (Derman et al. 1996a) to construct implied trees with the interpolated option prices as input. The implied trees reprice options correctly. A drawback with the implied tree approach is that it can’t be guaranteed that the branching probabilities in the trees are non-negative. It can be fixed with heuristic rules that are applied to override nodes with illegal branching. According to Andersen and Brotherton-Ratcliffe (1998), those rules are not only unsatisfactory, but also result in loss of information that can easily lead to significant compounded pricing errors.

The method proposed by Andersen and Brotherton-Ratcliffe (ibid.) also relies on interpolation of observable option prices. Instead of using a tree based model, they propose a finite differences approach, described in this thesis in Section 3.2.2, where they let the local volatility in each node be solved to reprice options consistent with the interpolated prices. Testing the technique on S&P 500 options, they manage to reproduce the observable option prices with the resulting local volatility surface. However, the resulting surface has a somewhat peculiar look, demonstrated in Figure 4 in the article, making the reader question how realistic it is, considering the smoothness property. Andersen and Brotherton-Ratcliffe’s (1998) forward recursive approach using Arrow-Debreu securities is harder to implement but more computationally efficient and has better convergence properties compared to the implied trees of Derman et al. (1996a).

A more modern approach is to use the Stochastic Volatility Inspired parameterization (Gatheral and Jacquier 2012) for interpolation of observable option prices (or their implied volatility) and then in (3.58) express \( C \) and its partial derivatives in terms of implied volatility. It is our understanding that this approach is used in practice because it is intuitive and simple, like many other models that have proven flaws.

Regardless of how the interpolation and extrapolation of observable option prices is done, such methods are shown to be subject to artificial misinterpretation and stability issues. The resulting local volatility surfaces are very sensitive to how the input data is interpolated, see e.g Crepey (2002). Another problem with interpolation methods like those of Dupire (1994), Derman and Kani (1994b), Derman et al. (1996a) and Andersen and Brotherton-Ratcliffe (1998), as expressed by Coleman et al. (1998), is that the only objective is to fit the local volatility surface to observable option prices. This results in local volatility models that accurately reprice options but that aren’t realistic.

3.5.2 Parametric Models

Using parametric models, one lets the local volatility surface take the form of some parameterization and tries to minimize the pricing error in comparison to observable option prices. Doing so, one hopefully obtains a local volatility surface that accurately reprices options with respect to observable option prices and is realistic.

Coleman et al. (ibid.) suggested that the problem of constructing a local volatility surface can be regarded as a nonlinear inverse function approximation from a finite data set and formed a least square optimization problem. The formulation is as follows. Let \( C(S_0, r, q, K, T; \sigma(s, t)) \) denote the function used to price a plain vanilla European call option with underlying asset price \( S_0 \), dividend yield \( q \), strike price \( K \), maturity \( T \) and local volatility function \( \sigma(s, t) \) for the underlying asset. Further, assume that we are given \( m \) observable data-tuples, \( \{(P_{i}^{bid}, P_{i}^{ask}, K_{i}, T_{i})\}_{i=1}^{m} \), containing bid price \( P_{i}^{bid} \), ask price \( P_{i}^{ask} \), strike price \( K_{i} \), and maturity \( T_{i} \) for option \( i \). Let \( C_{i}^{obs} = \frac{P_{i}^{bid} - P_{i}^{ask}}{2} \). Coleman et al. (ibid.) motivate that a suitable parameterization of \( \sigma(s, t) \) has proven to be a bicubic spline with the number of spline knots, \( p \), not exceeding the number of observable option prices. Splines will be further discussed in Section 3.6, but for now, observe a general parameterization,

\[
\sigma(s, t) = P(s, t; \bar{\sigma}),
\]  

\((3.59)\)
where $P(s,t;\bar{\sigma})$ is a parameterization given the $p$-vector $\bar{\sigma}$ of volatility values in the knots. For simplicity, we define

$$C_i(P(s,t;\bar{\sigma})) = C(S_0, r, q, K_i, T_i; P(s,t;\bar{\sigma})).$$

(3.60)

The optimization problem can be written as

$$\min_{\bar{\sigma}} f(P(s,t;\bar{\sigma})) := \sum_{i=1}^{m} w_i \left(C_i(P(s,t;\bar{\sigma})) - C_i^{obs}\right)^2,$$

(3.61)

s.t $\bar{\sigma} \in X,$

(3.62)

where $X$ defines constraints for the volatility in the knots, for example forcing positive volatility. In other words, it is a minimization problem with respect to the volatility in the knots, $\bar{\sigma}$. The weights $w_i$ can be used to make sure that the surface is more or less accurate for certain observable option prices. Typically you would want to be more precise around ATM strikes for example. Coleman et al. (1998) used a finite difference approach, see for example Section 3.2.2, to evaluate (3.61) and a gradient-based optimization solver to find a (locally) optimal solution. The gradients were evaluated by numeric differentiation. They tested their model with S&P 500 option data and showed that they could reproduce somewhat accurate option prices and hedging ratios. The resulting local volatility surface depends on the choice of knots for the parameterization, as presented in figure 6 and 7 in the article. The surface is clearly more realistic, in terms of smoothness, than the result of Andersen and Brotherton-Ratcliffe (1998).

Coleman et al. (1998), as mentioned, claimed that the number of knots for the parameterization should be less than or equal to the number of data points. This reduces the degrees of freedom significantly. However, if more knots are used, they explain that a regularization term should be introduced and added to (3.61). This is done in the work of Jackson et al. (1998), who also used a spline representation of the local volatility surface. Jackson et al.’s (1998) model were tested on the FTSE-100 index. The resulting surface, illustrated in Figure 5 of the article, could be used to reprice options accurately within a decent pricing error tolerance.

Parametric models clearly provide surfaces with better smoothness properties than the interpolation models. However, one clear disadvantage is that a parameterization needs to be chosen, determining beforehand the shape of the surface. Another weakness pointed out by Coleman et al. (1998) and Geng et al. (2014) is that the choice of the number of knots and their placement has a significant impact on the result.

### 3.5.3 Non-parametric Models

The approach of non-parametric models for construction of local volatility surfaces aims to solve the inverse problem of fitting a surface to observable option prices without a predetermined parameterization. Instead, regularization is applied to a discretized local volatility surface and the only assumption regarding the shape is that it is smooth. The first attempt using this approach was made by Lagnado and Osher (1997) who used the first order derivatives of the local volatility for regularization. Geng et al. (2014) extended the approach by minimizing the second order derivatives instead, producing local volatility surfaces with better smoothness properties. The resulting surface, when using the same observable option prices as Coleman et al. (1998) and Andersen and Brotherton-Ratcliffe (1998) of S&P 500 index options, is concluded to have no spikes, lower pricing errors, and looks significantly more realistic in terms of smoothness. The result is illustrated in figure 15 in the article. However, in a test that used observable option prices from Eurostoxx 50 index options, the shape of the local volatility surface, illustrated in figures 17 and 20 in the article, is not as satisfactory by comparison. The local volatility surface has clear spikes for some maturities. The authors explain the issues by stating that the choice of discretization parameters in the finite difference approach for repricing the options has a significant impact on the result. Barkhagen and Blomvall (2015) developed a general optimization based framework that produced smooth and realistic surfaces consistent with market prices of S&P 500 options. They were also first in the field of measuring local volatility to prove that the model produced squared local variance stable over time, through time series analysis.

The general description of the non-parametric approach is to solve an optimization problem similar to (3.61), but with the modifications that a regularization term is added to the objective function and the design variables are the entire local volatility surface, expressed in a set of discretized points, $u$. The formulation is, using the
same notation as in (3.61),
\[
\min_u f(u) := \sum_{i=1}^{m} w_i \left( C_i(u) - C_i^{\text{obs}} \right)^2 + \lambda_h h(u),
\]
where \( h(u) \) is a function containing numerical approximations of different orders of partial derivatives of \( u \). The parameter \( \lambda_h \) is a scalar that determines how much impact the regularization should have on the objective function, i.e., how important the smoothness property is in relation to pricing error.

### 3.6 Spline Parameterization

As explained in Section 3.5.2, previously used methods for constructing local volatility surfaces involve parametric representation of the surface. The problem of fitting a parameterization to represent the local volatility surface can be solved using splines. By definition, a spline of order \( k \) is a function \( F(x) \) defined for all real \( x \) that satisfies the following properties,

- It is composed of polynomial arcs of degree at most \( k - 1 \).
- It is of class \( C^{k-2} \), i.e., \( F(x) \) has \( k - 2 \) continuous derivatives.

Consider the one-dimensional case of fitting an interpolating spline, \( F(x) \), on a set of \( n \) grid points with values \( f(x_i) \) where \( x_0 < x_1 \ldots < x_n \). Each subinterval, \([x_i, x_{i+1}]\), \( i = 0, \ldots, n - 2 \), satisfies
\[
F_i(x) = \sum_{p=0}^{k-1} c_{i,p} (x - x_i)^p,
\]
where \( F_i \) is the polynomial for subinterval \([x_i, x_{i+1}]\). The grid point interpolation gives,
\[
\begin{align*}
F_i(x_i) &= f(x_i), & i &= 0, 1, \ldots, n - 2 \\
F_i(x_{i+1}) &= f(x_{i+1}), & i &= 0, 1, \ldots, n - 2.
\end{align*}
\]

From the definition we also get that,
\[
\begin{align*}
F_i(x_{i+1}) &= F_{i+1}(x_{i+1}), & i &= 0, 1, \ldots, n - 3, \\
F_i'(x_{i+1}) &= F_{i+1}'(x_{i+1}), & i &= 0, 1, \ldots, n - 3, \\
F_i''(x_{i+1}) &= F_{i+1}''(x_{i+1}), & i &= 0, 1, \ldots, n - 3.
\end{align*}
\]

Depending on the user’s preference, different boundary conditions can be used for the end-points \( x_0 \) and \( x_n \). Coleman et al. (1998) use the natural spline conditions,
\[
F_0''(x_0) = F_n''(x_n) = 0,
\]
which is typical if the slope of \( f \) is unknown at the edges. However, "clamped" boundaries can be used to force the edges to a certain slope,
\[
\begin{align*}
F_0'(x_0) &= f'(x_0), \\
F_n'(x_n) &= f'(x_n).
\end{align*}
\]

Solving for all the constants, \( c_{i,p} \) where \( i = 0, 1, \ldots, n - 2 \) and \( p = 0, 1, \ldots, k - 1 \), an interpolating spline in one dimension is achieved and any \( x \) on the interval \([x_0, x_n]\) can be evaluated. In engineering, the cubic spline (\( k = 4 \)) is popular since it gives the smoothest twice continuously differentiable functions that match observed data points (ibid.).

#### 3.6.1 General Bicubic Interpolation

The bicubic interpolation method, performs the interpolation with cubic polynomials (\( k = 4 \)) in two dimensions at once. Let \( f \) be a function of two variables, \( k \) and \( \tau \), further let
\[
X_{i,j} = (k_i, \tau_j), \quad i = 1, 2, \ldots, n \quad \text{and} \quad j = 1, 2, \ldots, m,
\]

\[
14
\]
be the input vector representing the grid-points. If \( X \) is thought of as a rectangle, we can proceed to divide it into \((n-1)(m-1)\) subrectangles, \(R_{i,j}\), defined by

\[
R_{i,j} = \{(k,\tau)|k_i < k < k_{i+1}, \tau_j < \tau < \tau_{j+1}\}. \tag{3.74}
\]

The bicubic interpolation function \( F \) can then be modeled as

\[
F_{i,j}(k,\tau) = \sum_{p=0}^{3} \sum_{q=0}^{3} c_{i,j,p,q} (k - k_i)^p (\tau - \tau_j)^q,
\]

\[
(k,\tau) \in R_{i,j}, i = 1,2,...,n-1 \text{ and } j = 1,2,...,m-1.
\tag{3.76}
\]

The linear system of equations arising from the \( C^2 \)-continuity in the corners of each sub-rectangle \(R_{i,j}\) can be expressed in terms of a matrix equation. We denote the corners by \(f(0,0), f(0,1), f(1,0)\) and \(f(1,1)\). The coefficients \(c_{p,q}\) of \(F_{i,j}\) are put into a vector

\[
\alpha = [c_{0,0} \ c_{1,0} \ c_{2,0} \ c_{3,0} \ c_{0,1} \ c_{1,1} \ c_{2,1} \ c_{3,1} \ c_{0,2} \ c_{1,2} \ c_{2,2} \ c_{3,2} \ c_{0,3} \ c_{1,3} \ c_{2,3} \ c_{3,3}]^T. \tag{3.77}
\]

And from the gridpoint-values we construct

\[
\beta = [f(0,0) \ f(1,0) \ f(0,1) \ f(1,1) \ f_k(0,0) \ f_k(0,1) \ f_k(1,0) \ f_k(1,1) \ f_{\tau}(0,0) \ f_{\tau}(0,1) \ f_{\tau}(1,0) \ f_{\tau}(1,1) \ f_{k\tau}(0,0) \ f_{k\tau}(0,1) \ f_{k\tau}(1,0) \ f_{k\tau}(1,1)]^T, \tag{3.78}
\]

where \(f_k, f_\tau\) and \( f_{k\tau} \) denote the derivative of \(f\) with respect to \(k, \tau\) and the mixed partial derivative of the two respectively. Solving the matrix equation \(M\alpha = \beta\), with the transformation matrix

\[
M = \begin{bmatrix}
1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 0 & 0 & -2 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
2 & -2 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
-3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
9 & -9 & -9 & 9 & 6 & 3 & -6 & -3 & 6 & -6 & 3 & -3 & 4 & 2 & 2 & 1 \\
6 & -6 & -6 & 6 & -3 & -3 & 3 & 3 & 4 & 4 & -2 & -2 & -2 & -2 & -1 & -1 \\
2 & 0 & -2 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-6 & 6 & 6 & -6 & 6 & -4 & 4 & 4 & 2 & -3 & -3 & -3 & -3 & -2 & -2 & -1 & -1 \\
4 & -4 & -4 & 4 & 4 & 2 & 2 & 2 & -2 & -2 & -2 & -2 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
\tag{3.79}
\]

gives the coefficients for one sub-rectangle. The entire surface parameterization is obtained if the system is solved for all grid-points, patching all rectangles together.

### 3.6.2 Bicubic Spline

The bicubic spline is a special case of the generalized bicubic interpolation. Interpolation using cubic polynomials has been used in finance applications by, for example Hagan and West (2006), who solve the related problem of constructing interest rate yield curves. Extending this to the two-dimensional case of fitting a volatility surface, Coleman et al. (1998) use the bicubic spline. The method solves for a one dimensional cubic spline twice, where the first result is stored and interpolated on the second time. In practice, this means interpolating in time- or strike-space first, and using the resulting splines as input data in the second run of the procedure to obtain the bicubic spline representing the surface.

The advantages of using splines in particular for local volatility surfaces, as expressed by Coleman et al. (ibid.) and Jackson et al. (1998), can be summarized as the following points:

- It is simple and convenient to have a structure that is uniquely determined by a finite number of constant weights. A number that is user-configurable.
- The surface is guaranteed to be smooth, which is desirable for a local volatility surface to be realistic and for fast convergence on numerical procedures.
- It is a flexible and general parameterization compared to other non-spline representations.
3.7 Interior-point Methods

Constructing a local volatility surface often involves solving a non-linear optimization problem (see e.g. Coleman et al. 1998 or Geng et al. 2014). Non-linear optimization problems can be solved efficiently using interior-point methods, which solve the KKT-conditions by applying Newton’s method (Boyd and Vandenberghe 2004).

Assume that an optimization problem can be written on the form

$$\min_x f(x) \quad \text{s.t.} \quad Ax = a \quad Bx \leq b.$$  

Further, assume that $f$ is twice continuously differentiable. Introduce slack variables, $s$, and a logarithmic barrier with accuracy $\mu_t$ to get

$$\min_x f(x) - \mu_t 1^T \ln s \quad \text{s.t.} \quad Ax = a \quad Bx + s = b \quad s > 0.$$  

The Lagrange function is

$$\mathcal{L}(x, y_a, y_b) = f(x) - \mu_t 1^T \ln s + y_a^T (Ax - a) + y_b^T (Bx - b - s),$$  

where $y_a$ and $y_b$ are vectors containing dual variables for (3.84) and (3.85) respectively. This gives us the KKT-conditions,

$$\nabla_x f(x^*) + A^T y_a^* + B^T y_b^* = 0 \quad (3.88)$$

$$A x^* = a \quad (3.89)$$

$$B x^* + s^* = b \quad (3.90)$$

$$\text{diag}(y_a^*) s = \mu_t 1 \quad (3.91)$$

$$y_b^* \geq 0 \quad (3.92)$$

$$s^* > 0. \quad (3.93)$$

for optimal values $(x^*, y_a^*, y_b^*, s^*)$. Linearizing (3.88)-(3.91) around $(\bar{x}, \bar{y}_a, \bar{y}_b, \bar{s})^T$ gives

$$\nabla_x f(\bar{x}) + \nabla^2_x f(\bar{x}) \Delta x + A^T (\Delta y_a + \Delta y_b) + B^T (\Delta y_b) = 0 \quad (3.94)$$

$$A(\bar{x} + \Delta x) = a \quad (3.95)$$

$$B(\bar{x} + \Delta x) + \bar{s} + \Delta s = b \quad (3.96)$$

$$Y_b \Delta s + \bar{Y}_b \Delta y_b = \mu_t 1, \quad (3.97)$$

where the notation $\bar{Y}_b = \text{diag}(\bar{y}_b)$ is used. Finally this yields a linear system of equations,

$$\begin{bmatrix} \nabla^2_x f(\bar{x}) & AT & BT & 0 \\ A & 0 & 0 & 0 \\ B & 0 & 0 & I \\ 0 & 0 & \bar{S} & \bar{Y}_b \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y_a \\ \Delta y_b \\ \Delta s \end{bmatrix} = - \begin{bmatrix} \nabla_x f(\bar{x}) + A^T \bar{y}_a + B^T \bar{y}_b \\ A \bar{x} - a \\ B \bar{x} + \bar{s} - b \\ \bar{Y}_b - \mu_t 1 \end{bmatrix}, \quad (3.98)$$

that can be solved to get the step direction. A simplified algorithm for solving the optimization problem formulated in the form of (3.80)-(3.82) can thus be described as:

0. Choose a starting point $z^0 = (x, y_a, y_b, s)^T$ and initial value for $\mu_t$. Set $k = 0$.

1. Solve (3.98) for $(\bar{x}, \bar{y}_a, \bar{y}_b, \bar{s})^T = z^k$.

2. Set $z^{k+1} = z^k + \alpha (\Delta x, \Delta y_a, \Delta y_b, \Delta s)^T$, with appropriate step length $\alpha$.

3. Update barrier, $\mu_t = 0.1 s^T \bar{y}_b / m$. 

16
4. Check some stopping criteria, norm of gradient etc.

5. Set $k = k + 1$ and go to 1.

This is called a primal-dual interior point solver, which is very efficient for convex non-linear optimization, finding the global optimum fast (Blomvall 2018).

3.8 Automatic Differentiation

Automatic differentiation (AD) is a technique used to numerically evaluate partial derivatives of functions. It is useful in many fields, for example machine learning or gradient and hessian based optimization with objective functions that are difficult to differentiate (Baydin et al. 2015). That makes it a useful tool for the problem of constructing a local volatility surface, which is often formulated as an optimization problem with an objective function that is evaluated numerically, i.e. hard to differentiate. AD is built on the theory that all computer programs can be interpreted as a series of elementary operations and functions (addition, subtraction, multiplication, division, exp, log, sin, cos, etc.) on which the chain-rule can be applied. Let $f(x) = y_3(y_2(y_1(x)))$, then the derivative of $f$ with respect to $x$ can be calculated as

$$\frac{df}{dx} = \frac{df}{dy_3} \frac{dy_3}{dy_2} \frac{dy_2}{dy_1} \frac{dy_1}{dx}. \quad (3.99)$$

Because of this, Capriotti and Giles (2011) states that the derivative of any programming subroutine can be found through AD. Automatic differentiation comes in mainly two different settings, forward mode and reverse mode, both of which can be implemented (Baydin et al. 2015) either with Operator Overloading or by Source Code Transformation.

3.8.1 Forward Mode

An implication of the chain rule is that (3.99) can be written as

$$\frac{df}{dx} = \frac{df}{dy_3} \frac{dy_3}{dx}. \quad (3.100)$$

Forward (or tangent) mode AD computes directional derivatives by propagating the chain in (3.100) in a forward manner, meaning that the algorithm proceeds by expanding the rightmost derivative,

$$\frac{df}{dx} = \frac{df}{dy_3} \frac{dy_3}{dy_2} \frac{dy_2}{dy_1} \frac{dy_1}{dx} = \frac{df}{dy_3} \left( \frac{dy_3}{dy_2} \frac{dy_2}{dy_1} \frac{dy_1}{dx} \right). \quad (3.101)$$

If the functions depend on multiple input parameters of which the derivatives are sought, the amount of computations needed increase linearly since each traversal of the chain result in the derivative with respect to only one input. On the contrary, the forward mode AD only requires one traversal of the chain to obtain all output derivatives with respect to a specific input parameter, also known as a tangent, or a column of the Jacobian matrix. Forward mode Automatic Differentiation can be illustrated with a simple example.

Let $y(x_1, x_2) = (\sin(x_1) - x_1x_2)^2$. Denote by $w$ all variables and functions of the subroutine $y$. The calculation scheme can be written as

$$w_1 = x_1 \quad (3.102)$$

$$w_2 = x_2 \quad (3.103)$$

$$w_3 = w_1w_2 = x_1x_2 \quad (3.104)$$

$$w_4 = \sin(w_1) = \sin(x_1) \quad (3.105)$$

$$w_5 = w_4 - w_3 = \sin(x_1) - x_1x_2 \quad (3.106)$$

$$w_6 = w_5^2 = (\sin(x_1) - x_1x_2)^2. \quad (3.107)$$

Following (3.102) - (3.107) will give the function value of $y$, we call this a "forward sweep" of the algorithm. The methodology to obtain a tangent is quite intuitive as the derivative with respect to the parameter is calculated in each step of the sweep. To obtain the tangent of $x_1$, the following computations can be made alongside the
function value calculation (\(\dot{w}\) means that \(w\) is derived with respect to the target parameter),

\[
\begin{align*}
\dot{w}_1 &= 1 \quad \text{(seed)} \\
\dot{w}_2 &= 0 \\
\dot{w}_3 &= \dot{w}_1 w_2 + w_1 \dot{w}_2 = x_2 \\
\dot{w}_4 &= \cos(w_1) \dot{w}_1 = \cos(x_1) \\
\dot{w}_5 &= \dot{w}_4 - \dot{w}_3 = \cos(x_1) - x_2 \\
\dot{w}_6 &= 2w_5 \dot{w}_5 = 2(\sin(x_1) - x_1 x_2) (\cos(x_1) - x_2).
\end{align*}
\]

Analogously for \(x_2\) we get

\[
\begin{align*}
\dot{w}_1 &= 0 \\
\dot{w}_2 &= 1 \quad \text{(seed)} \\
\dot{w}_3 &= \dot{w}_1 w_2 + w_1 \dot{w}_2 = x_1 \\
\dot{w}_4 &= \cos(w_1) \dot{w}_1 = 0 \\
\dot{w}_5 &= \dot{w}_4 - \dot{w}_3 = -x_1 \\
\dot{w}_6 &= 2w_5 \dot{w}_5 = 2(\sin(x_1) - x_1 x_2) (-x_1).
\end{align*}
\]

### 3.8.2 Reverse Mode

Reverse (or adjoint) mode AD computes directional gradients by propagating the chain in the reversed order, using the fact that (3.99) can be written as

\[
\frac{df}{dx} = \frac{df}{dy_1} \frac{dy_1}{dx}.
\]

The chain can be expanded like

\[
\frac{df}{dx} = \frac{df}{dy_1} \frac{dy_1}{dx} = \left( \frac{df}{dy_2} \frac{dy_2}{dy_1} \right) \frac{dy_1}{dx} = \left( \frac{cf}{dy_3} \frac{dy_3}{dy_2} \right) \frac{dy_2}{dy_1} \frac{dy_1}{dx}.
\]

The reverse mode only has to traverse the chain once to obtain the entire gradient of a specific output (one row of the Jacobian matrix). Hence, when the number of input parameters is greater than the number of resulting outputs, the reverse mode AD is the most efficient, but the opposite case favours the forward mode (Antonov 2017). Intuitively, reverse mode Automatic Differentiation can be a bit trickier to follow, but an example is illustrated below.

Again, let \(y(x_1, x_2) = (\sin(x_1) - x_1 x_2)^2\) and consider the calculation scheme (3.102) - (3.107). As with the forward mode, a sweep is first run from top to bottom to obtain the function values along the subroutine, but the derivative is not calculated during this sweep. Instead, all the intermediate function values are saved to be used later in the "backward sweep". To compute the gradient in the backward sweep, the following formula is applied in a bottom-up direction,

\[
\tilde{w}_i = \sum_{j \in \pi_i} \frac{dw_{i}}{dw_{j}} \tilde{w}_{j},
\]

where \(\tilde{w}_i\) denotes the intermediate derivative of \(w_i\) with respect to all input parameters and \(\pi_i\) is \(w_i\)'s following operations (all expressions that holds \(w_i\)). Applying (3.110) to (3.102) - (3.107) gives,

\[
\begin{align*}
\tilde{w}_6 &= 1 \\
\tilde{w}_5 &= 2w_5 \tilde{w}_6 = 2w_5 \\
\tilde{w}_4 &= \tilde{w}_5 = 2w_5 \\
\tilde{w}_3 &= -\tilde{w}_5 = -2w_5 \\
\tilde{w}_2 &= w_1 \tilde{w}_3 = w_1 (-2w_5) = -2x_1 (\sin(x_1) - x_1 x_2) \\
\tilde{w}_1 &= \tilde{w}_2 \tilde{w}_3 + \cos(w_1) \tilde{w}_4 = w_2 (-2w_5) + \cos(w_1) 2w_5 \\
&= 2(\sin(x_1) - x_1 x_2) (\cos(x_1) - x_2).
\end{align*}
\]
3.8.3 Operator Overloading
Operator Overloading uses polymorphic features of a programming language to override basic algebraic operations and elementary functions to make them return both the function value and its derivative. This is done along with creating some data structure to hold both these values. Operator overloading is typically easy to implement (Merriënboer et al. 2018) and naturally supports all features of the host language. A drawback is that it can create large overhead if procedures that are data-dependent have to be retraced as a result of the introduced data-type. The implementation will of course depend on the language, but assuming $x$ and $y$ are instances of the new datatype, the idea is to make the call $x + y$ return both $x + y$ and $\dot{x} + \dot{y}$. Analogously, $x \cdot y$ would return both $x \cdot y$ and $\dot{xy} + x \cdot \dot{y}$ and so on for all operators needed.

3.8.4 Source Code Transformation
Source Code Transformation (SCT) is done by using some compiler or package/library to transform the original subroutine to a new version of it that also calculates the intermediate derivatives needed in the chain rule. SCT does not create any runtime overhead (ibid.) but can be harder to implement, and requires the AD-tool to explicitly support all of the used features of the host language. An advantage with SCT is that the generated code can be analyzed and further optimized by the user.

Normally, when we want to evaluate a function like $y = x_1 x_2 + x_2$, it will be handled by the computer as a subroutine of the form,

$$x_3 = x_1 \cdot x_2$$
$$y = x_3 + x_2.$$  (3.118)

Using SCT, the idea is to send the code of the subroutine into a function that will return new code, including the calculation of the derivative. Let’s say SCT is used in forward mode to calculate the tangent of $x_1$. A resulting block of code from the function can look like

$$x_3 = x_1 \cdot x_2$$  (3.120)
$$dx_3 = x_2 = (dx_1 \cdot x_2 + x_1 \cdot dx_2 = 1 \cdot x_2 + x_1 \cdot 0)$$  (3.121)
$$y = x_3 + x_2$$  (3.122)
$$dy = x_2 = (dx_3 + dx_2 = x_2 + 0),$$  (3.123)

where $dx$ is the derivative of $x$ with respect to $x_1$.

3.9 Stochastic Local Volatility
The effective theory, as it is named by Derman and Kani, relies on the assumption that local volatility is static, which they criticize. As shown by Gyöngy (1986), the local volatility is deterministic given all available information at a certain time. But as soon as new information becomes available, it changes. In order to model the stochastic dynamics of the local volatility, Derman and Kani (1997) introduce several independent brownian motions, $W_i, i = 0, ..., n$, with which the surface of local variance, $\sigma^2_{K,T}(t,S)$, is allowed to vary stochastically according to

$$d\sigma^2_{K,T}(t,S) = \alpha_{K,T}(t,S)dt + \sum_{i=0}^{n} \theta^i_{K,T}(t,S)dW^i_t, $$  (3.124)

where $W^0 = Z$ is the wiener process of the underlying asset, $S$. The coefficients $\theta^i, i = 0, ..., n$ are unrestricted aside from mild measurability and integrability conditions. The drift functions $\alpha_{K,T}(t,S)$ are bound to no-arbitrage conditions derived in (ibid.), Appendix D. The no-arbitrage conditions are needed in order to ensure that the process is consistent with an arbitrage-free market. To fully understand the meaning of the no-arbitrage conditions derived by Derman and Kani (ibid.), one must define what an arbitrage free market is and how one decides if a market is arbitrage free.

3.9.1 Fundamental Theorem of Asset Pricing
An arbitrage opportunity represents the limitless creation of wealth through risk-free profit. We can define arbitrage through a self-financing strategy $\varphi$. Let $V_t$ denote the value of $\varphi$ at time $t$, then $\varphi$ is an arbitrage opportunity on $[0,T]$ if

- $V_0 = 0$,
• $V_T \geq 0$, $\mathbb{P}$-almost surely,
• $\mathbb{P}(V_T > 0) > 0$,

where $\mathbb{P}$ is the probability measure. The fundamental theorem of asset pricing stated through Delbaen and Schachermayer (1994) says that "the existence of an equivalent martingale measure is essentially equivalent to the absence of arbitrage opportunities", which means that if and only if the market is arbitrage free under $\mathbb{P}$, we can find an equivalent martingale measure (EMM), say $\tilde{\mathbb{P}}$, under which the market is free of arbitrage. For $\tilde{\mathbb{P}}$ to be an EMM on the filtration $\mathcal{F}_T$, we must have that

$$\mathbb{P}(A) = 0 \iff \tilde{\mathbb{P}}(A) = 0, \quad \forall A \in \mathcal{F}$$

and

$$\frac{S_i(t)}{G(t)} \text{ are } \tilde{\mathbb{P}}\text{-martingales for all } i = 0, 1, 2, \ldots, N,$$

where $S_0(t), \ldots, S_N(t)$ are the price processes that make up the market model and $G$ is the numeraire under $\tilde{\mathbb{P}}$. A general process $X_t$ is called a $\mathbb{P}$-martingale (Williams 1991) relative to $\mathcal{F}$ if

$$X_t \text{ is adapted to } \mathcal{F},$$

$$\mathbb{E}^\mathbb{P}[|X_t|] < \infty, \quad \forall t,$$

$$\mathbb{E}^\mathbb{P}[X_{t+1}|\mathcal{F}_t] = X_t, \quad \text{almost surely}, \quad t \geq 0.$$  

### 3.9.2 Drift Conditions

With a clear definition of what defines an arbitrage-free market, the no-arbitrage conditions for the drift terms $\alpha_{K,T}(t,S)$ in (3.124) are derived by studying transition probabilities instead of option prices. Let $P_{K,T}(t,S) = p(t,S,T,K)$ denote the probability that the underlying asset will reach level $K$ at time $T$, given that it has value $S$ at $t$. Assuming that the local variance varies over time according to (3.124), the dynamics are given by

$$dP_{K,T}(t,S) = \left(\frac{\partial P_{K,T}(t,S)}{\partial t} + \mu S \frac{\partial P_{K,T}(t,S)}{\partial S} + \frac{1}{2} \sigma^2_{K,T}(t,S) S^2 \frac{\partial^2 P_{K,T}(t,S)}{\partial S^2}\right) dt + \sigma_{K,T}(t,S) S \frac{\partial P_{K,T}(t,S)}{\partial S} dW_t^0$$

$$+ \frac{1}{2} \int_t^T \int_0^T \int_0^\infty \frac{\partial^2 P_{K,T}(t,S)}{\partial \sigma^2_{K',T'}(t,S)} d\sigma_{K',T'}^2(t,S) dK' dT'$$

where $\frac{\partial P_{K,T}(t,S)}{\partial \sigma^2_{K',T'}(t,S)}$ denotes the functional derivative of $P_{K,T}(t,S)$ with respect to $\sigma^2_{K',T'}(t,S)$. The variables $K', K'', T', T''$ denote the integrating variables. Substituting (3.124) into (3.130), using some properties of the transition probabilities and rearranging some terms gives the expression

$$dP_{K,T}(t,S) = \sigma_{K,T}(t,S) S \frac{\partial P_{K,T}(t,S)}{\partial S} \left( dW_t^0 + \frac{\mu(t) - \delta + q}{\sigma_{K,T}(t,S)} dt + \sum_{i=0}^{n} \left( \int_t^T \int_0^\infty \frac{\partial P_{K,T}(t,S)}{\partial \sigma^2_{K',T'}(t,S)} \theta_{K',T'}^i(t,S) dK' dT' \right) dW_t^i \right)$$

$$+ \left( \int_t^T \int_0^\infty \frac{\partial P_{K,T}(t,S)}{\partial \sigma^2_{K',T'}(t,S)} \left( \alpha_{K',T'}(t,S) + \frac{1}{2} \sum_{i=0}^{n} \theta_{K',T'}^i(t,S) \left( \frac{1}{p(t,S,T',K') \partial K''} \int_0^\infty \theta_{K'',T''}^i(t,S) p(t,S,T'',K'') K''^2 \frac{\partial^2}{\partial K''^2} p(T'', K'', T', K') dK'' dT'' \right) dK' dT' \right) \right) dt.$$  

Under an equivalent martingale measure, $d\tilde{W}_t^0 = dW_t^0 + \frac{\mu(t) - \delta + q}{\sigma_{K,T}(t,S)} dt$, $d\tilde{W}_t^i = dW_t^i + \Pi_t^i dt$, $i = 1, \ldots, n$.  

---
where \( \Pi \) is the market price of risk associated with the risk factor \( W^i \), the underlying asset has to be a martingale for the market to be arbitrage free. That is equivalent to the underlying asset having a drift equal to zero, which is equivalent to the transition probabilities having a drift equal to zero. Therefore the drift in (3.131) has to be zero for the market to be arbitrage free. We see that the drift parameters \( \alpha_{K,T}(t,S) \) in (3.124) must satisfy the no-arbitrage conditions
\[
\alpha_{K,T}(t,S) = -\sum_{i=0}^{n} \theta_{i,K,T}(t,S) \left( \frac{1}{2} \frac{1}{p(t,S,T,K)} \times \int_{t}^{T} \int_{0}^{\infty} \theta_{K',T'}(t,S)p(t,S,T',K')K'^{2} \frac{\partial^{2}}{\partial K'^{2}} p(T',K',T,K) dK' dT' - \Pi \right). \tag{3.134}
\]
The resulting dynamic under the equivalent martingale measure is
\[
dP_{K,T}(t,S) = \sigma_{K,T}(t,S) \frac{\partial P_{K,T}(t,S)}{\partial S} d\tilde{W}_t + \sum_{i=0}^{n} \left( \int_{t}^{T} \int_{0}^{\infty} \delta P_{K,T}(t,S) \frac{\partial^{2}}{\partial \sigma_{K',T'}^{2}}(t,S) \theta_{i,K',T'}(t,S) dK' dT' \right) d\tilde{W}_t, \tag{3.135}
\]
and the process does not imply arbitrage. The conditions in (Derman and Kani 1997) were commented on by Carmona and Nadtochiy (2009), meaning that there is a missing term. In Appendix A we redo the original proof more thoroughly to arrive at (3.134), which is very similar to Derman and Kani’s (1997) conditions, although there is a factor 2 difference.

### 3.10 Principal Component Analysis

Derman and Kani’s (1997) model for the dynamics of the local volatility surface, see (3.124), involves identifying independent factors with respect to which the surface changes. Principal Component Analysis (PCA) is a technique for doing just that. PCA is a multivariate technique for analyzing a data set by transforming it into a composition of eigenvectors, \( q \), and eigenvalues, \( \lambda \) (Wold et al. 1987). In financial applications, the covariance matrix of a data set is often used since it is symmetric and positive semi definite and can thus be represented through the eigenvalue decomposition,
\[
C = Q\Lambda Q^{-1} = QAQ^T = \sum_{i=1}^{n} \lambda_i q_i q_i^T, \tag{3.136}
\]
where \( n \) is the dimension of the data set, \( q_i \) the eigenvectors, \( Q \) is the matrix \((q_1, \ldots, q_n)\), and \( \Lambda \) is a diagonal matrix of the eigenvalues \( \lambda_i \). Jolliffe (2011) states that the purpose of PCA is to reduce the dimensionality of the data to a smaller set that still retains and explains the variation in the original data. By studying the eigenvalues sorted in a decreasing order, the variance through each dimension is obtained and it is possible to determine each principal component’s contribution to the total variance. The first \( k \) components explains the fraction
\[
\sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{n} \lambda_i \tag{3.137}
\]
of the total variance and a desirable level can be attained by choosing an appropriate \( k \). If the correlation between the elements of the studied data is high, the first few principal components explains a lot of the variation and the data can be largely reduced resulting in a much simpler set to comprehend. You can then get a good estimation of the covariance matrix,
\[
C \approx Q_k \Lambda_k Q_k^T = \sum_{i=1}^{k} \lambda_i q_i q_i^T, \tag{3.138}
\]
where \( Q_k \) is the matrix \((q_1, \ldots, q_k)\) and \( \Lambda_k \) is a diagonal matrix of the first \( k \) eigenvalues.

If PCA is done for historical observations \( x_i \) of a stochastic vector \( X_t \), i.e an eigenvalue decomposition of the covariance matrix \( C \), one can calculate the principal component \( j \) at time \( t \), \( p_{j,t} \), as
\[
p_{j,t} = q_j^T x_t, \quad j = 1, \ldots, k \tag{3.139}
\]
or on matrix form

\[ p_t = Q_k^T x_t, \]  

where \( p_t \) is a vector containing the principal components at time \( t \). Then it is possible to form a simple and low-dimensional stochastic process by simulating the principal components, \( p_t \),

\[ x_t = x_{t-1} + a_{t-1} + Q_k p_t, \quad \text{Var}(p_t) = \Lambda_k, \]  

(3.141)

which can be simplified to

\[ x_t = x_{t-1} + a_{t-1} + Q_k \sqrt{\Lambda_k} \xi_t, \quad \text{Var}(\xi_t) = I, \quad \text{i.i.d.} \]  

(3.142)

where \( \sqrt{\Lambda_k} \) is a diagonal matrix with the square root of the first \( k \) eigenvalues on the diagonal. This holds since

\[ \text{Var}(x_t) = \sqrt{\Lambda_k} \text{Var}(p_t) (\sqrt{\Lambda_k})^T \approx C, \]  

(3.143)

which is a good approximation of the covariance matrix. A suitable distribution for \( \xi_t \) can be chosen, as long as \( \xi(p_t) = I \). A well known example is the case for PCA of the forward interest rate yield curve, where a dimension of \( k = 3 \) often explains enough of the variation to accurately describe how the curve changes over time.

It should be noted that if (3.142) is used it is assumed that the principal components have no auto correlation, since the simulation is done independently of previously observed values of the principal components. This is central in analyzing the suitability for simulating \( x_t \) by simulating principal components. The descriptive ability of a Principal Component Analysis may be significantly affected by auto correlation (Vanhatalo and Kulahci 2016).

### 3.11 Kernel Density Estimator

The main purpose of the thesis is to measure a probability distribution for the future value of a portfolio of equity index options. The probability distribution of financial data can rarely be accurately expressed through closed form solutions, but accurate measures are of high importance for many different applications, e.g. valuation and risk management. Kristensen and Shin (2012) suggest a way of measuring the probability density function \( f \) through a kernel density estimator. If \( n_{\text{sim}} \) i.i.d. simulated observations \( \{\tilde{V}_k\}_{k=1}^{n_{\text{sim}}} \) is drawn from a model, the kernel density of \( f \) can be estimated as,

\[ \hat{f}(\tilde{V}) = \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} K_h(\tilde{V} - \tilde{V}_k), \]  

(3.144)

where

\[ K_h(x) = \frac{K(x/h)}{h} \]  

(3.145)

is the scaled kernel function for the bandwidth \( h > 0 \) and kernel function \( K(\cdot) \). Barkhagen et al. (2016) use this technique to measure the unknown CDF of the portfolio with estimates of the kernel density for a series of simulated two-day periods. Let \( \hat{f}_i(x) \) be attained with a Gaussian kernel, then the scaled kernel function is given by

\[ K_h(\tilde{V} - \tilde{V}_k) = \frac{1}{\sqrt{2\pi h}} e^{-\frac{\tilde{V} - \tilde{V}_k)^2}{2h}} \]  

(3.146)

and the kernel density estimator is hence given by

\[ \hat{f}(\tilde{V}) = \frac{1}{\sqrt{2\pi h n_{\text{sim}}}} \sum_{k=1}^{n_{\text{sim}}} e^{-\frac{(\tilde{V} - \tilde{V}_k)^2}{2h}}. \]  

(3.147)

If the underlying density being measured is Gaussian, the bandwidth \( h \) has been proven (by e.g. Silverman 1986) to have an optimal value at

\[ h = \left( \frac{4\hat{\sigma}^5}{3 n_{\text{sim}}} \right)^{\frac{1}{5}} \approx 1.06 \hat{\sigma}(n_{\text{sim}})^{-\frac{1}{5}}, \]  

(3.148)
where $\hat{\sigma}$ is the sample standard deviation of the simulated values. When (3.147) is used to approximate the $t_d$-day kernel density function, the actual outcomes of the portfolio value $t_d$ days after the approximation, $V_{t+t_d}$, can be used to evaluate the CDF values, $\hat{F}$, by

$$\hat{F}(V_{t+t_d}) = \int_0^{V_{t+t_d}} f_t(x) dx = \int_{-\infty}^{V_{t+t_d}} \frac{1}{\sqrt{2\pi}hn_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} e^{-\frac{(x-\tilde{V}_k)^2}{2h^2}} dx = \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} \int_{-\infty}^{V_{t+t_d}} \frac{1}{\sqrt{2\pi}h} e^{-\frac{(x-\tilde{V}_k)^2}{2h^2}} dx$$

$$= \left. \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} \int_{-\infty}^{\frac{V_{t+t_d}}{h} - \frac{\tilde{V}_k}{h}} \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}} dx \right|_{\tilde{x} = \frac{x - \tilde{V}_k}{h}} = \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} \mathcal{N}_0,1 \left( \frac{V_{t+t_d} - \tilde{V}_k}{h} \right),$$

(3.149)

which is easily calculated using the standard normal distribution CDF, $\mathcal{N}_0,1(\cdot)$.

### 3.12 Inverse Transform Method and Q-Q Plots

To evaluate if a distribution for a stochastic variable, e.g. the value of a portfolio of options, is accurate by observing outcomes of said stochastic variable, one can use the Inverse Transform Method and visualize the result in a Quantile-Quantile plot (Q-Q plot). If $\hat{F}$ is the true CDF of $V_t$, then

$$\hat{F}(V_t) \sim U(0,1),$$

(3.150)

i.e. uniformly distributed on $[0,1]$, and therefore

$$Y_t = \mathcal{N}_{0,1}^{-1} (\hat{F}(V_t)) \sim \mathcal{N}(0,1),$$

(3.151)

where $\mathcal{N}_{0,1}^{-1}(\cdot)$ is the inverse CDF of the standard normal distribution, according to the Inverse Transform Method. This is useful since we can observe outcomes of $Y_t$ and compare them to the normal distribution quantiles through a Q-Q plot. If $\hat{F}$ is a good estimation of $V_t$’s true CDF, then the outcomes of $Y_t$ should be normally distributed and the quantiles should be similar to the ones of the standard normal distribution, i.e a straight line in the Q-Q plot with slope 1 crossing the origin.
4 Method
To fulfill the purpose of this thesis, four main problems have to be solved. Firstly, the local volatility surface have to be constructed for historical dates. The surface should be consistent with observable option prices and realistic. Secondly, systematic changes of said local volatility surface have to be analyzed and translated to a stochastic process that accurately describes how it changes over time. Thirdly, the option portfolio’s future value has to be evaluated multiple times, using simulations of the local volatility surface, implying a probability distribution that can be measured. Lastly, the model’s performance needs to be evaluated. It will be done through an out-of-sample evaluation method. In this section, a method for solving the four main problems is laid out. Additionally, some implementation issues and data related questions are clarified.

4.1 Constructing the Local Volatility Surface
The parametric approach of Coleman et al. (1998), with some modifications, is chosen for construction of the local volatility surface for historical dates. The method takes into account both the objectives of fitting the local volatility surface to observable option prices and producing a realistic surface in terms of smoothness. Even though a non-parametric approach would have some advantages against the chosen one, it is considered too extensive to implement given that it is just part of the study as a whole. This section describes how the local volatility surface will be constructed by:

- formulation of the optimization problem,
- parameterization of the local volatility surface,
- pricing of options,
- solving of the optimization problem,
- and automatic differentiation of the objective function.

4.1.1 The Optimization Problem
Assume that we are given \( m \) observable data-tuples, \( \{(P_{\text{bid}}^i, P_{\text{ask}}^i, K_i, T_i)\}_{i=1}^m \), containing bid price \( P_{\text{bid}}^i \), ask price \( P_{\text{ask}}^i \), strike price \( K_i \), and maturity \( T_i \) for European call option \( i \). Let \( C_i^{\text{obs}} = \frac{P_{\text{bid}}^i + P_{\text{ask}}^i}{2} \). We also define a parameterization of the local volatility surface,

\[
\sigma(s, t) = \mathcal{P}(s, t; \bar{\sigma}), \quad (4.1)
\]

which is unique for a given vector \( \bar{\sigma} \) of local volatility in the knots, \( \{s_i, t_j\}, \ i = 0, ..., N_p, \ j = 0, ..., M_p \), where \( N_p + 1 \) is the number of knots in the price-direction, and \( M_p + 1 \) in the time-direction. Lastly, we define

\[
C_i(\bar{\sigma}) = C(S_0, r, q, K_i, T_i; \mathcal{P}(s, t; \bar{\sigma})) \quad (4.2)
\]

as a function that prices a call option with strike price \( K_i \) and maturity \( T_i \) given that the underlying asset has the current value \( S_0 \), dividend yield \( q \), risk-free interest rate \( r \) and the parameterization of the local volatility surface \( \mathcal{P}(s, t; \bar{\sigma}) \). The problem of constructing the local volatility surface is formulated as

\[
\min_{\bar{\sigma}} f(\bar{\sigma}) := \sum_{i=1}^m w_i \left( C_i(\bar{\sigma}) - C_i^{\text{obs}} \right)^2, \quad (4.3)
\]

s.t

\[
0.05 \leq \mathcal{P}(s_j, t; \bar{\sigma}) \leq 1, \quad j = 0, 1, 2, 3. \quad (4.4)
\]

Coleman et al. (ibid.) use the constraint that the local volatility in the spline knots should be between 0 and 1. The other values, i.e. in between and outside the knots, on the local volatility surface are not bound to this constraint. As an attempt to improve Coleman et al. (ibid.)’s model, we introduce (4.4) to force all the local volatility values along each strike-knot level, for the entire \( t \)-axis to be in the interval \([0.05, 1]\). Ideally, all points on the surface should be positive, but we choose not to impose further constraints. The problem is already heavily constrained by the parameterization and we do not want to force a certain shape onto the local volatility surface more than we already do. The upper and lower bound are chosen intuitively to not yield unrealistic solutions. The lower bound was initially set to 0 but changed to 0.05 because of stability issues in the option pricing methodology. For volatilities too small or too large in relation to the drift of the underlying
asset, the pricing leads to numerical problems when using finite difference methods. However, since S&P 500 has a realized volatility around 0.15 it is a choice that should not exclude any desired solutions.

The weights \( \{w_i\}_{i=1}^m \) are chosen so that repricing is more accurate for the most liquid options (close to at-the-money strikes and short maturities). More specifically they are chosen as

\[
  w_i = \frac{1}{e^s + \frac{p_{\text{bid}} - p_{\text{ask}}}{C_i^{\text{bid}}}},
\]

where \( e^s = 0.01 \) is a constant chosen to avoid infinitely large weights. This means that the weights are reversely proportional to the relative spread in bid/ask-price, i.e. the option with the smallest spread in relation to its observed price is given the largest weight. The relationship is used since the most liquid options tend to have the smallest spread. And we want to give more weight to liquid options since they provide more information than illiquid options.

### 4.1.1.2 Parameterization of the Local Volatility Surface

The Bicubic Spline presented in Section 3.6 is chosen as parameterization method. Advantages of using spline parameterization was discussed in Section 3.6.2, where the smoothness, ease of use, and strong convergence property is considered highly attractive for the purpose of this thesis.

The local volatility is expressed as a function of strike price, \( K \), and time-to-maturity, \( \tau = T - t \), in years. We choose a set of strike knot levels \( K_i, i = 0, \ldots, N_p \) and maturity knot levels \( \tau_j, j = 0, \ldots, M_p \), the two dimensions can be discretized to a \((N_p + 1) \times (M_p + 1)\) mesh. In this thesis, we choose the mesh as \([0.650 : \frac{1}{3}S_0 : 1.650] \times [0, 0.25, 0.5, 1.5] \), i.e. \( N_p = M_p = 3 \), resulting in 16 knots, for representing the points in \( K \) and \( \tau \). The choices of edges and step lengths in the mesh are motivated by the fact that we want to have knots where there is available and reliable information, i.e. observable market prices and we need the knot values to be comparable over time, i.e as \( S_0 \) changes making different strikes of options relevant and traded. That will correspond to a certain range in moneyness \((K/S_0)\) and time to maturity. We also want more information from shorter time to maturity than longer which is why the knots are placed with smaller distance for time to maturity less than 0.5 years. The choice to use only 16 knots is purely made due to computational cost. The optimization solver has a hard time converging in a reasonable time using more knots.

The Bicubic Spline is achieved with the procedure presented in Section 3.6.2. Firstly, the interpolation is made in strike-space, obtaining one cubic spline for each level of \( \tau \) in the mesh, we call this a "sub-spline" from now on. From Section 3.6 we have that each sub-interval (the space between two adjacent knots in a sub-spline) fits a third degree polynomial. For 4 strike-knots, there will be 3 polynomials in each sub-spline. They can be written as

\[
  F_i(x) = c_{i,0} + c_{i,1}(x - x_i) + c_{i,2}(x - x_i)^2 + c_{i,3}(x - x_i)^3, \quad i = 0, \ldots, 2,
\]

where \( x_i \) is the lower bound on interval \( i \). In order to be able to evaluate the sub-spline at any \( x \), the \( 3 \times 4 \) coefficients \( c_{i,p}, i = 0, 1, 2, p = 0, 1, 2, 3 \) has to be solved for. Combining (4.6) with the continuity constraints (3.67), (3.68) and (3.69), we get a new set of equations,

\[
  c_{i,0} = \sigma(x_i), \quad \text{for } i = 0, 1, \ldots, n - 2,
\]

\[
  \sum_{p=0}^{k-1} c_{n-2,p}(x_{n-1} - x_{n-2})^p = \sigma(x_{n-1}), \quad \text{for } i = 0, 1, \ldots, n - 2,
\]

\[
  \sum_{p=0}^{k-1} (c_{i,p}(x_{i+1} - x_i)^p) - c_{i+1,0} = 0, \quad \text{for } i = 0, 1, \ldots, n - 3,
\]

\[
  \sum_{p=1}^{k-1} (pc_{i,p}(x_{i+1} - x_i)^{p-1}) - c_{i+1,1} = 0, \quad \text{for } i = 0, 1, \ldots, n - 3,
\]

\[
  \sum_{p=2}^{k-1} (p(p - 1)c_{i,p}(x_{i+1} - x_i)^{p-2}) - 2c_{i+1,2} = 0, \quad \text{for } i = 0, 1, \ldots, n - 3,
\]

where \( x_i \) is the lower bound on interval \( i \). In order to be able to evaluate the sub-spline at any \( x \), the \( 3 \times 4 \) coefficients \( c_{i,p}, i = 0, 1, 2, p = 0, 1, 2, 3 \) has to be solved for. Combining (4.6) with the continuity constraints (3.67), (3.68) and (3.69), we get a new set of equations,
where \( \sigma(x_i) \) is the volatility value at strike knot \( i \). Using the natural spline end conditions we get, by putting (3.70) into (3.64),

\[
2c_{0,2} = 0, \quad (4.12)
\]

\[
\sum_{p=2}^{k-1} p(p-1)c_{n-2,p}(x_{n-1} - x_{n-2})^{p-2} = 0. \quad (4.13)
\]

Then, (4.7) - (4.11), together with (4.12) and (4.13), form a matrix equation like

\[
Ac = \begin{bmatrix} b \\ \sigma \end{bmatrix}, \quad (4.14)
\]

where

\[
A = \begin{bmatrix}
1 & x_1 - x_0 & (x_1 - x_0)^2 & (x_1 - x_0)^3 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1 & 2(x_1 - x_0) & 3(x_1 - x_0)^2 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 6(x_1 - x_0) & 0 & 0 & -2 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & x_2 - x_1 & (x_2 - x_1)^2 & (x_2 - x_1)^3 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 2 & 6(x_2 - x_1) & 0 & -1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -2 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix},
\]

\[
c = \begin{bmatrix} c_{0,0} \\ c_{0,1} \\ c_{0,2} \\ c_{0,3} \\ \vdots \\ c_{3,0} \\ c_{3,1} \\ c_{3,2} \\ c_{3,3} \end{bmatrix}, \quad \sigma = \begin{bmatrix} \sigma(x_0) \\ \vdots \\ \sigma(x_3) \end{bmatrix}
\]

and \( b \) is a column-vector of \( 3 \times (N_p - 1) + 2 = 8 \) zeros. Using linear-algebra matrix operations we solve for \( c \) by

\[
c = A^{-1} \begin{bmatrix} b \\ \sigma \end{bmatrix}. \quad (4.15)
\]

The same procedure is done for all four levels of \( \tau_j, \; j = 0, \ldots, 3 \) in the mesh, which in our case gives 4 sub-splines. This means that for any \( K \), we can get 4 local volatility values \( \sigma(K, \tau_j), \; j = 0, \ldots, 3 \) by evaluating all sub-splines at that \( K \). Using these values, a new set of spline coefficients, \( c^K \), can be calculated putting the maturity-knots in \( A \) and using \( \sigma(K, \tau_j) \) as volatility at the knots.

Since we now have a way of getting a spline for any value of \( K \) (call this a \( K \)-spline, represented by \( c^K \)), we can obtain a parameterized local volatility, \( P(K, \tau; \sigma) \), for any combination of \( K \) and \( \tau \) by evaluating the \( K \)-spline at the desired \( \tau \). The evaluation is done by locating the sub-interval in which \( \tau \) lies and putting it into the polynomial.

Assume we are evaluating three different values of \( \tau \) at the same strike, further let these three be located in sub-intervals 0, 1 and 2 respectively. The evaluation can then be expressed as

\[
P(K, \tau; \sigma) = Bc^K, \quad (4.16)
\]

where

\[
B = \begin{bmatrix}
1 & \tau - \tau_0 & (\tau - \tau_0)^2 & (\tau - \tau_0)^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & \tau - \tau_1 & (\tau - \tau_1)^2 & (\tau - \tau_1)^3 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau - \tau_2 & (\tau - \tau_2)^2 & (\tau - \tau_2)^3 \\
\end{bmatrix}
\]
If a maturity cannot be located in the interval \([\tau_0, \tau_3]\), linear extrapolation is used with the derivatives at the end-knots as slopes. Suppose we evaluate two maturities which are lower than \(\tau_0\) and higher than \(\tau_3\) respectively, the \(B\)-matrix would then look like
\[
B = \begin{bmatrix}
1 & \tau - \tau_0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & \tau - \tau_2 & (\tau_3 - \tau_2)^2 + 2(\tau_3 - \tau_2)(\tau - \tau_3) & (\tau_3 - \tau_2)^3 + 3(\tau_3 - \tau_2)^2(\tau - \tau_3)
\end{bmatrix}.
\]

In our case we will be evaluating multiple \(\tau\)-values, which can be represented on matrix form once each sub-interval has been located.

### 4.1.3 Option Pricing by a Finite Difference Method

The objective function in (4.3) contains the function
\[
C_i(\bar{\sigma}) = C(S_0, r, q, K_i, T_i; (P(s, t; \bar{\sigma}))
\]
which, per definition, is a function that prices a European call option given the parameters, \(S_0\), \(r\), \(q\), \(K_i\), \(T_i\), \(\bar{\sigma}\) and the parameterization of the local volatility surface \(P(s, t; \bar{\sigma})\). The option pricing will be done through the finite differences approach presented by Andersen and Brotherton-Ratcliffe (1998). The choice is mainly motivated because it makes it possible to price a lot of options in little computational time, in comparison to for example simulation techniques. Additionally, it is a purely deterministic approach which is desireable when optimizing, since it doesn’t introduce additional errors from iteration to iteration that might cause convergence issues when solving the optimization problem.

We create a grid with discrete points for logarithmic asset price \(x_t = \ln S_t\) and time \(t\) with \(M + 2\) points along the \(t\) axis and \(N + 2\) along the \(x\) axis by defining
\[
\begin{align*}
  x_i, & \quad i = 0, \ldots, N + 1 \\
  t_j, & \quad j = 0, \ldots, M + 1.
\end{align*}
\]

The grid is illustrated in Figure 3. Although not visually illustrated, note that the discretization of the \(x\)- and \(t\)-axis defines a non-uniform grid.

![Non-uniformly spaced grid for finite difference methods.](image-url)
In $x$-space we choose to use a constant discretization size, $\Delta x$, and edges, $x_0$ and $x_{N+1}$, of the grid as dependent of the current price of the underlying asset $S_{ini} = e^{x_{ini}}$. We let
\[ x_0 = 0.5 \cdot x_{ini}, \]
\[ x_{N+1} = 2 \cdot x_{ini}, \]
\[ \Delta x = \frac{x_{N+1} - x_{ini}}{\frac{1}{4}(N+1)} = \frac{x_{ini} - x_0}{\frac{1}{4}(N+1)}, \]

where we choose $N$ so that $N + 1$ is evenly divisible by 3. This choice of discretization in $x$-space will prove convenient when setting the boundary conditions for the grid. We then get
\[ x_i = x_0 + i \cdot \Delta x, \quad i = 0, \ldots, N + 1, \]
which is also convenient since we can use the same notation as Andersen and Brotherton-Ratcliffe (1998).

Discretization in $t$-space, however, is not done with a constant size. We want to be more exact for short maturities than longer ones and ideally we should try to include the maturities that correspond to the observable option prices. We choose a discretization size of one day until maturity of the first maturing option, then weekly discretization for the next eleven months and monthly discretization for the next twelve months. In that way, we will be able to choose the time points, $t_j$, $j = 1, \ldots, M + 1$, to include the relevant option maturities, since all options in the data set will have a maturity corresponding to the third Friday of the month (after January 2016, before that the third Saturday), or the closest non-bank holiday. More formally, we choose
\[ \Delta t_j = \begin{cases} \frac{1}{365}, & t_j < T_1 \\ \frac{T_{n+1} - T_n}{4}, & T_1 \leq t_j < T_{12} \\ T_{n+1} - T_n, & t_j \geq T_{12} \end{cases}, \quad j = 0, \ldots, M, \]

where $T_n$ denotes the $n$:th maturity in the data set. In the grid we then have
\[ t_j = \sum_{k=0}^{j-1} \Delta t_j, \quad j = 1, \ldots, M + 1 \]
\[ t_0 = 0 \]
and we should be able to match the option maturities. We choose $N = 299$ for this experiment, resulting in 301 points in $x$-space. In $t$-space, the number of time points will depend on several things, including the time of the first maturity for the observable option price data, however, it will at most amount to 87 points ($31 \cdot 1 + 4 \cdot 11 + 1 \cdot 12$).

For computational efficiency, we choose to solve for Arrow-Debreu prices in the grid nodes instead of option prices. In that way, we can price a range of options with varying strike price and maturity through one iteration over the $M + 2$ grid “columns”. Let $A_{i,j,k}(l \geq j)$ denote the price at node $(x_i, t_j)$ of the Arrow-Debreu security that pays out 1$ at $t_l$ if and only if node $(x_k, t_l)$ is reached. It can be noted, as a consequence of the definition of the Arrow-Debreu securities, that if $d_j^k$ denotes the discount factor with maturity $t_j$ as observed at time $t_l$, then
\[ p(t_j, S_i, t_l, S_k) = A_{i,j,k}^{d_j^k} \]

is the transition probability, i.e the probability that the underlying asset price reaches level $S_k = e^{x_k}$ at time $t_l$, given that it is at level $S_i$ at time $t_j$.

The most relevant Arrow-Debreu prices are the ones that are observed in node $(x_{ini}, t_0)$, where $x_{ini} = e^{S_{ini}}$ and $S_{ini}$ is the current price of the underlying asset. In other words, we want to find the Arrow-Debreu prices observed at the date on which we want to price options. Let $A_{ini,j,k}$ denote just that, i.e the price of the Arrow-Debreu security at node $(x_{ini}, t_0)$ that pays out 1$ if the underlying asset reaches $S_i = e^{x_i}$ at time $t_j$. Further, we define the vector
\[ A_{ini}^j = \begin{bmatrix} A_{ini,1,j} \\ A_{ini,2,j} \\ \vdots \\ A_{ini,N,j} \end{bmatrix}, \]

\[ (4.28) \]
which can be seen as a column in the grid in Figure 3 but without the top and bottom element, i.e \( i = 0, N + 1 \). As Andersen and Brotherton-Ratcliffe (1998) showed, the recursive relationship

\[
(A_{\text{ini}}^{j+1})^T = - \frac{\Theta}{1 - \Theta} (A_{\text{ini}}^j)^T + (A_{\text{ini}}^j)^T \left( \frac{d_0^j}{d_{j+1}^0} I - (1 - \Theta) M_j \right)^{-1} \left( \frac{\Theta d_{j+1}^i}{I} + (1 - \Theta) \right), \quad j = 0, \ldots, M
\]

(4.29)

holds, where \( I \) is the \( N \times N \) identity matrix and \( M_j \) a tridiagonal matrix,

\[
M_j = \begin{bmatrix}
c_{1,j} & u_{1,j} & 0 & 0 & 0 & \cdots & 0 \\
l_{2,j} & c_{2,j} & u_{2,j} & 0 & 0 & \cdots & 0 \\
& \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
0 & 0 & 0 & \cdots & l_{N-1,j} & c_{N-1,j} & u_{N-1,j} \\
0 & 0 & 0 & \cdots & 0 & l_{N,j} & c_{N,j}
\end{bmatrix},
\]

(4.30)

where

\[
c_{i,j} = -\alpha_j v_{i,j},
\]

(4.31)

\[
u_{i,j} = \frac{1}{2} \alpha_j (v_{i,j} + \Delta x b_{i,j}),
\]

(4.32)

\[
l_{i,j} = \frac{1}{2} \alpha_j (v_{i,j} - \Delta x b_{i,j}),
\]

(4.33)

and

\[
\alpha_j = \frac{\Delta t_j}{(\Delta x)^2},
\]

(4.34)

\[
\Delta t_j = t_{j+1} - t_j,
\]

(4.35)

and finally

\[
b_{i,j} = r(t_j) - q(t_j) - \frac{1}{2} v_{i,j},
\]

(4.36)

\[
v_{i,j} = \sigma^2 (S_i, t_j) = (P(e^{\mu_i}, t_j; \bar{\sigma}))^2.
\]

(4.37)

The parameter \( \Theta \) is set to 0.5, implying that the Crank-Nicholson method is used. The choice is well motivated since it gives the best convergence properties among the available alternatives. The approach is in theory unconditionally stable but, due to numerical difficulties, in practice it is not. That is partly why we chose the lower boundary in (4.4).

With reasonable assumptions, i.e a grid that spans a sufficient part of the relevant \( x \)-space, boundary conditions are only needed for \( j = 0, \quad i = 1, \ldots, N \) (the nodes marked with red in Figure 4). Intuitively, if \( x_0 \) and \( x_{N+1} \) are chosen so that the underlying asset realistically will never reach such a high or low level before time \( t_{M+1} \), it should hold that

\[
A_{\text{ini}}^{0,j} = A_{\text{ini}}^{N+1,j} = 0, \quad j = 1, \ldots, M + 1.
\]

(4.38)

This is achieved by the choices in (4.20) and (4.21).

The boundary conditions for \( j = 0 \) (the red nodes in Figure 4) are already decided following the definition of Arrow Debreu securities and the fact that \( x_{\text{ini}} \) is known at \( t_0 \). We have

\[
A_{\text{ini}}^0 = \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1 \\
\vdots \\
0
\end{bmatrix},
\]

(4.39)

where the only element that is equal to 1 is \( A_{\text{ini}}^{k,0} \) for which \( x_k = x_{\text{ini}} \). This is the reason we chose to define the \( x \)-space discretization based on \( x_{\text{ini}} \).
Once \( A_{i,j}^{t_{ini}} \), \( i = 0, \ldots, N + 1 \), \( j = 0, \ldots, M + 1 \) have been obtained using (4.29) with boundary conditions (4.39), the option price \( C^{i,j} \) corresponding to a European call option with strike price \( K = S_i = e^{x_i} \) and maturity \( T = t_j \) can be calculated as the expectation

\[
C^{i,j} = \sum_{l=i+1}^{N+1} A_{l,j}^{t_{ini}} (S_l - S_i),
\]

(4.40)

which is why we wanted to match the discrete time points with the option maturities.

The risk-free interest rate \( r \) will be modelled as deterministic. An implementation for measuring deterministic instantaneous forward rates \( f_{t_0}(t) \), for time \( t \) as observed at \( t_0 \), from observable prices for Overnight Index Swap contracts has been provided by the internal supervisor Jörgen Blomvall. The implementation provides instantaneous forward rate yield curves for historical dates. For more information about the method used for measuring the yield curves, see Blomvall (2017). The curves are discretized with tenors of one day. Given the forward rate yield curve \( \{ f_{t_0}(t_k) \}_{t_k=t_0}^{t_{M+1}} \), the discount factors used in (4.29) can be calculated as

\[
d_j^l = \exp \left( -\frac{1}{365} \sum_{t_k=t_l}^{t_j} f_{t_0}(t_k) \right).
\]

(4.41)

The component \( r(t_j) \) in (4.36) is the continuous risk-free rate at time \( t_j \) with maturity at \( t_{j+1} \) in the FDM grid. In terms of the forward rates it can be calculated as

\[
r(t_j) = \frac{1}{365 \Delta t_j} \sum_{t_k=t_j}^{t_{j+1}} f_{t_0}(t_k).
\]

(4.42)

The dividend yield \( q \) will also be modelled as deterministic, measured from observable prices of futures contracts for maturities less than 3 months and from option prices using put call parity for longer maturities. The separate methods for short and longer maturities are needed since future contracts are only traded for certain maturities. The option market is less liquid than the futures market for shorter maturities, but the reversed relationship
typically holds for the longer maturities. Given the observable future price \( F_{t_0}(T, S_t) \) at time \( t_0 \) with maturity \( T \) and underlying asset \( S_t \), we have
\[
F_{t_0}(T, S_t) = S_{t_0} e^{(r_{t_0}(T) - q_{t_0}(T))(T-t_0)},
\]
which is equivalent to
\[
q_{t_0}(T) = r_{t_0}(T) + \frac{\ln \left( \frac{S_{t_0}}{F_{t_0}(T, S_t)} \right)}{T-t_0},
\]
where \( q_{t_0}(T) \) and \( r_{t_0}(T) \) are the deterministic dividend yield and risk-free interest rate respectively, as observed at time \( t_0 \) with maturity \( T \). Using the instantaneous forward rates we have
\[
r_{t_0}(T) = \frac{1}{365(T-t_0)} \sum_{t_k=t_0}^T f_{t_0}(t_k)
\]
and \( q_{t_0}(T) \) is calculated from (4.44) for maturities less than 3 months. Now, consider observable put and call option prices, \( C_{t_0}(T, S_t) \) and \( P_{t_0}(T, S_t) \), at time \( t_0 \) with maturity \( T \). Put-call parity is expressed as
\[
C_{t_0}(T, S_t) + Ke^{-r_{t_0}(T)(T-t_0)} = P_{t_0}(T, S_t) + S_{t_0}e^{-q_{t_0}(T)(T-t_0)},
\]
which yields
\[
q_{t_0}(T) = \frac{1}{T-t_0} \ln \left( \frac{C_{t_0}(T, S_t) - P_{t_0}(T, S_t) + Ke^{-r_{t_0}(T)(T-t_0)}}{S_{t_0}} \right),
\]
whereby \( q_{t_0}(T) \) is calculated for maturities longer than 3 months. The component \( q(t_j) \) in (4.36) is the continuous dividend yield of the underlying asset from \( t_j \) to \( t_{j+1} \), which means we have to calculate the implied forward dividend. If we calculate \( q_{t_0}(T) \) and \( q_{t_0}(T) \) from (4.44) or (4.47), depending on \( T \) and \( \bar{T} \), we have that
\[
q_{t_0}(T)(T-t_0) = q_{t_0}(T)(\bar{T}-t_0) + q_{t_0}(T)(T-\bar{T}), \quad \bar{T} \leq T,
\]
which gives
\[
q_{t_0}(T) = \frac{q_{t_0}(T)(T-t_0) - q_{t_0}(T)(\bar{T}-t_0)}{T-\bar{T}}
\]
If we have observable future contract and option prices for all maturities \( t_j \), \( j = 1, \ldots, M + 1 \), we can use (4.44), (4.47), and (4.49) to decide the dividend yield for all time points \( t_j \) in the FDM grid. This is not the case for us, which is why we make the assumption that the forward dividend yield is piece-wise constant for the intervals between the available future and option maturities.

4.1.4 Optimization Solver

The optimization problem (4.3) is a non-linear minimization problem with respect to the volatility in the parameterized surface. With information from the spline parameterization, we can rewrite the constraint (4.4) to be linear in the spline knots, \( \bar{\sigma} \). Using (4.15) in (4.16) we get that
\[
P(K_i; \tau; \bar{\sigma}) = BA^{-1} \left[ \begin{array}{c} b \\ \sigma(K_i; \tau) \end{array} \right], \quad i = 0, 1, 2, 3,
\]
where
\[
\sigma(K_i; \tau) = \left[ \begin{array}{c} \sigma(K_i; \tau_0) \\ \sigma(K_i; \tau_1) \\ \sigma(K_i; \tau_2) \\ \sigma(K_i; \tau_3) \end{array} \right].
\]
Since \( b \) is a vector of eight zeros, eight columns can be removed from the matrix \( BA^{-1} \). If we consider this reduced matrix, \( (BA^{-1})_{\text{red}} \), we can express all \( P(K_i; \tau; \bar{\sigma}) \), \( i = 0, 1, 2, 3 \), as
\[
\begin{bmatrix}
P(K_0; \tau; \bar{\sigma}) \\ P(K_1; \tau; \bar{\sigma}) \\ P(K_2; \tau; \bar{\sigma}) \\ P(K_3; \tau; \bar{\sigma})
\end{bmatrix} = A_{\text{const}} \bar{\sigma},
\]
where
\[ A_{\text{constr}} = \begin{bmatrix}
(BA^{-1})_{\text{red}} & 0 & 0 & 0 \\
0 & (BA^{-1})_{\text{red}} & 0 & 0 \\
0 & 0 & (BA^{-1})_{\text{red}} & 0 \\
0 & 0 & 0 & (BA^{-1})_{\text{red}} 
\end{bmatrix} \]  

(4.53)

The optimization problem in Section 4.1.1 can finally be written on the form,
\[
\min_{\bar{\sigma}} f(\bar{\sigma})  \\
\frac{\beta}{0.5} \leq A_{\text{constr}} \bar{\sigma} \leq 1,  
\]

(4.54)

(4.55)

Since the constraint (4.55) is linear, the problem is solved efficiently using an interior-point solver. The start solution for \( \bar{\sigma} \) is set to \( \bar{\sigma}_j = 0.15, \ j = 1, \ldots, p \), since that is around the empirical volatility of the S&P 500 index. When the optimization problem is solved for consecutive dates, the start solution for the first date will be 0.15 and after that the start solution for each date will be set to the resulting optimal solution of the prior date.

Since there are various implemented tools available for interior-point based methods, we do not implement our own solver, but instead use the package SciPy.optimize.minimize with the method set to "trust-constr", which solves the problem through an interior-point method when inequality constraints are introduced. However, solving the optimization problem using this method requires gradient and hessian functions for the objective function. Given that we can retrieve the gradient, the hessian can be approximated computationally fast using SciPy.optimize.HessianUpdateStrategy where we have chosen the BFGS strategy. The gradient however needs to be acquired somehow. This will be discussed next.

4.1.5 Automatic Differentiation

In Section 4.1.4, we clearly see that the derivative of the objective function is needed repeatedly in order to solve the optimization problem. This derivatives cannot be obtained analytically since, as presented in Section 4.1.3, the objective function is evaluated partly through the finite difference method. To find the derivatives, automatic differentiation will be applied to the evaluation. This is mostly motivated by the fact that AD produces accurate derivatives in a fast manner and Baydin et al. (2015) states that it is a powerful tool for gradient and hessian based optimization with objective functions that are difficult to differentiate.

The python library Autograd will be used to implement automatic differentiation. Autograd uses reverse mode AD through operator overloading. In the optimization problem in Section 4.1.4, the number of function inputs (16) is larger than the number of outputs (1), and thus the reverse mode should produce faster derivatives than the forward mode. SCT would be chosen over operator overloading in this thesis if it was available in any Python package, because the generated code can be traced, debugged and optimized easily if needed, which is a desired feature in this case since a large number of derivatives will be calculated. However, there aren’t any available AD packages in Python that uses SCT, which makes the choice for us. The Autograd.grad function will be called on the routine for evaluating the objective function according to 4.1.3 and the resulting callable gradient function will be used as input to the optimization solver.

4.2 Analyzing the Systematic Changes of the Local Volatility Surface

We analyze the systematic changes of the local volatility surface by using an interpretation of Derman and Kani’s (1997) model. The analysis is done for historical dates by performing 3 steps. Firstly, we modify and rewrite Derman and Kani’s (1997) initial expression for the dynamics of the local variance to the dynamics of the logarithmic local variance. Secondly, we perform a Principal Component Analysis to identify independent factors that affect the logarithmic local variance over time. Lastly we show how to apply a no-arbitrage drift conditions that will depend on said factors and transition probabilities.

4.2.1 Dynamics of the Local Volatility Surface

The initial expression for the dynamics of the local variance in Derman and Kani’s (1997) model is
\[ d\sigma_{K,T}^2(t, S) = \alpha_{K,T}(t, S)dt + \sum_{i=0}^{n} \theta_{K,T}^i(t, S)dW_t^i. \]  

(4.56)
However, using (4.56), it is possible to obtain negative local variance, which isn’t desirable since it is unrealistic and therefore has to be handled somehow. We propose two changes to (4.56). Firstly we look at relative changes instead of absolute. Secondly, we propose that $\theta^i(t, S)$ are constants rather than functions of $t$ and $S$. With these changes, we get

$$\frac{d\sigma^2_{K,T}(t, S)}{\sigma^2_{K,T}(t, S)} = \alpha_{K,T}(t, S)dt + \sum_{i=0}^{n} \theta_{K,T}^i dW^i_t. \quad (4.57)$$

By using logarithmic local variance, we can find an expression that is more suitable for Principal Component Analysis. Let

$$v_{K,T}(t, S) = \ln \sigma^2_{K,T}(t, S). \quad (4.58)$$

To find the dynamics of (4.58) we apply Ito’s Lemma,

$$dv_{K,T}(t, S) = \frac{\partial v_{K,T}(t, S)}{\partial t} dt + \frac{\partial v_{K,T}(t, S)}{\partial \sigma^2_{K,T}(t, S)} d\sigma^2_{K,T}(t, S) + \frac{1}{2} \frac{\partial^2 v_{K,T}(t, S)}{\partial (\sigma^2_{K,T}(t, S))^2} (d\sigma^2_{K,T}(t, S))^2. \quad (4.59)$$

Since $W^i, \ i = 0, \ldots, n$, are independent and it can be shown that, in probability, it holds that

$$\langle dt \rangle^2 = 0, \quad (4.60)$$

$$dtdW^i = 0, \quad (4.61)$$

$$dW^i dW^j = \begin{cases} 0, & i \neq j \quad \text{(independent)}, \\ dt, & i = j \end{cases} \quad (4.62)$$

(4.59) can be written

$$dv_{K,T}(t, S) = \frac{1}{\sigma^2_{K,T}(t, S)} \left( \alpha_{K,T}(t, S)\sigma^2_{K,T}(t, S)dt + \sum_{i=0}^{n} \theta_{K,T}^i dW^i_t \right)$$

$$+ \frac{1}{2} \left( \frac{1}{\sigma^2_{K,T}(t, S)} \right)^2 \sum_{i=0}^{n} (\theta_{K,T}^i)^2 (\sigma^2_{K,T}(t, S))^2 dt$$

$$= \left( \alpha_{K,T}(t, S) - \frac{1}{2} \sum_{i=0}^{n} (\theta_{K,T}^i)^2 \right) dt + \sum_{i=0}^{n} \theta_{K,T}^i dW^i_t. \quad (4.63)$$

Rewriting the SDE (4.63) to the form of a stochastic integral equation, we get

$$v_{K,T}(t+\Delta t, S) = v_{K,T}(t, S) + \int_{t}^{t+\Delta t} \left( \alpha_{K,T}(u, S) - \frac{1}{2} \sum_{i=0}^{n} (\theta_{K,T}^i)^2 \right) du + \sum_{i=0}^{n} \int_{t}^{t+\Delta t} \theta_{K,T}^i dW^i_u, \quad \Delta t \geq 0. \quad (4.64)$$

Assuming $\Delta t$ is small, we can see $\alpha_{K,T}(t, S)$ as a constant and, using a Weiner process property

$$W_{t+\Delta t} - W_t = \sqrt{\Delta t} \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, 1) \ i.i.d., \quad \Delta t \geq 0, \quad (4.65)$$

(4.64) can be written

$$v_{K,T}(t+\Delta t, S) = v_{K,T}(t, S) + \left( \alpha_{K,T}(t, S) - \frac{1}{2} \sum_{i=0}^{n} (\theta_{K,T}^i)^2 \right) \Delta t + \sum_{i=0}^{n} \theta_{K,T}^i \sqrt{\Delta t} \xi_{t,i}, \quad \Delta t \geq 0, \quad (4.66)$$

where $\xi_t \sim \mathcal{N}(0, I) \ i.i.d.$

### 4.2.2 Principal Component Analysis

We perform the PCA on the logarithmic local variance in the spline knots, i.e. $\tilde{\sigma}$, to analyze the systematic changes in the local volatility surface. An alternative would be to use the resulting spline knots to reconstruct a surface for a set of strike-maturity points and use those points. However, we know that the spline introduces
additional errors in the local volatility surface which is why we consider the spline knots as the most representative points. This gives us a range of strike-price, \( K \), and time-to-maturity, \( \tau \), which we denote by \( [K_0, K_{N_p}] \) and \( [\tau_0, \tau_{M_p}] \). The ranges are placed on the mesh \([0.6S_0 : \frac{1}{3}S_0 : 1.6S_0] \times [0, 0.25, 0.5, 1.5] \). We build a matrix representing the spline knot mesh,

\[
\begin{pmatrix}
\sigma_{K_0, \tau_0} & \sigma_{K_1, \tau_0} & \ldots & \sigma_{K_{N_p-1}, \tau_0} & \sigma_{K_{N_p}, \tau_0} \\
\sigma_{K_0, \tau_1} & \sigma_{K_1, \tau_1} & \ldots & \sigma_{K_{N_p-1}, \tau_1} & \sigma_{K_{N_p}, \tau_1} \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\sigma_{K_0, \tau_{M_p}} & \sigma_{K_1, \tau_{M_p}} & \ldots & \sigma_{K_{N_p-1}, \tau_{M_p}} & \sigma_{K_{N_p}, \tau_{M_p}}
\end{pmatrix}_{(t)} ,
\]

(4.67)

where the index \( t \) indicates that the matrix is specified for each date \( t \). We want to study the systematic changes of this matrix over time through PCA and, since the underlying asset price varies with time, we choose to look at a fixed range of moneyness determined from the strike range above and the asset price at the present time. We let

\[
M_j = \frac{K_j}{S_t}, j = 0, \ldots, N_p ,
\]

(4.68)

where \( S_t \) is the price of the underlying asset at time \( t \), define the range of moneyness that is used for all matrices at any time. Hence, we modify (4.67) to

\[
\begin{pmatrix}
\ln (\sigma^2_{M_0, \tau_0}) & \ln (\sigma^2_{M_0, \tau_0}) & \ldots & \ln (\sigma^2_{M_{N_p-1}, \tau_0}) & \ln (\sigma^2_{M_{N_p}, \tau_0}) \\
\ln (\sigma^2_{M_0, \tau_1}) & \ln (\sigma^2_{M_1, \tau_0}) & \ldots & \ln (\sigma^2_{M_{N_p-1}, \tau_1}) & \ln (\sigma^2_{M_{N_p}, \tau_1}) \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
\ln (\sigma^2_{M_0, \tau_{M_p}}) & \ln (\sigma^2_{M_1, \tau_{M_p}}) & \ldots & \ln (\sigma^2_{M_{N_p-1}, \tau_{M_p}}) & \ln (\sigma^2_{M_{N_p}, \tau_{M_p}})
\end{pmatrix}_{(t)} ,
\]

(4.69)

Denote a column vector of (4.69) by \( \text{col}_{i,t} \), \( i = 0, \ldots, N_p \). The matrix is vectorized to form the \((M_p+1)(N_p+1)\)-vectors

\[
v_t = \begin{bmatrix}
\text{col}_{0,t} \\
\text{col}_{1,t} \\
\vdots \\
\text{col}_{N_p-1,t} \\
\text{col}_{N_p,t}
\end{bmatrix}, \quad \text{and} \quad \Delta v_t = v_{t+\Delta t} - v_t .
\]

(4.70)

Using the fixed discretization of spline knots in moneyness and time to maturity, the same procedure can be done for the historical dates in the studied data-set. This results in \( n_{\text{hist}} \) vectors that are used as columns of a \((M_p+1)(N_p+1) \times n_{\text{hist}}\) matrix

\[
(\Delta v_{t-n_{\text{hist}}+1}, \Delta v_{t-n_{\text{hist}}+2}, \ldots, \Delta v_{t-1}, \Delta v_t) .
\]

(4.71)

This is the matrix that we want to study with PCA. We find the covariance matrix of (4.71) and denote it by \( C \), which can be expressed through its eigenvalues, \( \lambda_i \), and eigenvectors, \( q_i \), through the eigenvalue decomposition

\[
C = QAQ^{-1} = Q\Lambda Q^T = \sum_{i=1}^{n_c} \lambda_i q_i q_i^T .
\]

(4.72)

The dimension of \( C \) becomes \( n_c \times n_c \), where \( n_c = (M_p+1)(N_p+1) \). \( Q \) is the matrix \((q_1, \ldots, q_{n_c})\) and \( \Lambda \) is a diagonal matrix of the eigenvalues \( \lambda_i \). If we sort the eigenvalues from largest to smallest we find the new dimension, \( k_{\text{PCA}} \), of the data set by assigning a target value to the fraction

\[
\sum_{i=1}^{k_{\text{PCA}}} \lambda_i \bigg/ \sum_{i=1}^{n_c} \lambda_i.
\]

(4.73)
We choose a target fraction of 99%, which means that we can explain 99% of the variance with the first \( k_{\text{PCA}} \) principal components, sorted by highest to lowest eigenvalue. With this new trimmed dataset, we get
\[
C \approx Q_{k_{\text{PCA}}} \Lambda_{k_{\text{PCA}}} Q_{k_{\text{PCA}}}^T = \sum_{i=1}^{k_{\text{PCA}}} \lambda_i q_i q_i^T. \tag{4.74}
\]
With this result, we can now form a simple stochastic process, as presented in 3.10, for the vector \( v_t \) as
\[
v_{t+\Delta t} = v_t + a_t + Q_{k_{\text{PCA}}} \sqrt{\Lambda_{k_{\text{PCA}}}} \xi_t, \quad \text{Var}(\xi_t) = I, \tag{4.75}
\]
where \( \sqrt{\Lambda_{k_{\text{PCA}}}} \) is a diagonal matrix with the square root of the first \( k_{\text{PCA}} \) eigenvalues on the diagonal and \( I \) is the identity matrix. Using (4.74) we can express (4.75) as
\[
v_{t+\Delta t} = v_t + a_t + \sum_{i=1}^{k_{\text{PCA}}} \sqrt{\lambda_i} q_i \xi_{t,i}. \tag{4.76}
\]

Remember the dynamic of the logarithmic local variance (4.66). If the exact discretization scheme that we presented in this chapter is used and the matrix is again rearranged to form the vector corresponding to \( v_t \), we realize that \( \sqrt{\Lambda q_i} \) gives us the vector containing the \( \theta^i \)-values for all spline knots we want to simulate. This result is central, and allows us to simulate the logarithmic local variance by simulating the principal components, i.e. by using (4.76). However, in order for the simulation to infer no arbitrage, some conditions on the drift coefficients \( \alpha_{K,T}(t,S) \) should be applied. This will be discussed next.

### 4.2.3 Drift Conditions

In Appendix A, we derive a proof for the drift condition limiting the term \( \alpha_{K,T}(t,S) \) in (4.66). The condition is presented as (A.16) in the appendix, and is very similar to (4.79) below. Under the equivalent martingale measure, \( Q \),
\[
dW^0_t = \frac{\mu(t) - r(t) + q(t)}{\sigma_{K,T}(t,S)} dt,
\]
\[
dW^i_t = dW^0_t + \Pi^i dt, \quad i = 1, \ldots, n, \tag{4.78}
\]
where \( \Pi^i \) is the market price of risk associated with \( W^i \). We can choose \( \Pi^i \) to an arbitrary value and we choose \( \Pi = 0 \). We can see that the choice of \( \Pi^i \) has no effect on the resulting process if we substitute \( \alpha_{K,T}(t,S) \) into (4.63) under \( Q \). The drift condition under \( Q \) thus becomes
\[
\alpha_{K,T}(t,S) = -\frac{1}{2} \sum_{i=0}^{n} \theta^i_{K,T} \frac{1}{p(t,S,T,K)} \int_t^T \int_0^\infty \frac{\sigma_{K',T'}^2(t,S)\theta^i_{K',T'}p(t,S,T',K')K'^2}{\varphi_{K,T}(T',K',T,K)K'^2} dK'^2 dt'. \tag{4.79}
\]
The variable \( \sigma_{K',T'}^2(t,S) \) is known and, with the result from the PCA, we also know how to obtain the constants \( \theta^i_{K,T} \) with \( \sqrt{\lambda q_i} \).

However, in lack of time to finish this thesis, we stay at providing a theoretical answer to how the drift condition can be set. Instead, we choose to simplify and set \( \alpha_{K,T}(t,S) = 0 \).

### 4.3 Estimating the Probability Distribution

We measure, at time \( t \), the probability distribution for the stochastic variable \( V_{t+\Delta t} \), representing the future value of a portfolio of equity index options at time \( t + \Delta t \), i.e. time \( \Delta t \) into the future. This is done in two steps. Firstly we simulate the underlying index price and the local volatility surface from \( t \) to \( t + \Delta t \) a number of times, \( n_{\text{sim}} \). The simulations are done with available information at time \( t \), i.e. from a constructed local volatility surface from market prices at time \( t \). For each simulation, we price the options in the portfolio to get \( n_{\text{sim}} \) simulated portfolio values, \( \{V_k\}_{k=1}^{n_{\text{sim}}} \). Secondly, the simulated portfolio values are used as input to a kernel density estimator to calculate the cumulative distribution function (CDF) for an arbitrary value by integration.
4.3.1 Simulating the Portfolio Value

We choose a portfolio consisting of a set of call options \( \{C^p_i(t, S_i)\}_{i=1}^{N_{\text{port}}} \) with corresponding strike prices and maturities \( \{(K_i, T_i)\}_{i=1}^{N_{\text{port}}} \). The portfolio value at time \( t \) is defined as

\[
V_t = \sum_{i=1}^{N_{\text{port}}} h^p_i C^p_i(t, S_t),
\]

where \( h^p_i \) is the number of contracts held associated with option \( C^p_i \). Since the underlying asset is assumed to follow the process

\[
dS_t = (r(t) - q(t))S_t dt + \sigma(S_t, t) dZ^Q_t,
\]

we can use the solution

\[
S_{t+\Delta t} = S_t \exp \left( \left( r(t) - q(t) - \frac{\sigma^2(S_t, t)}{2} \right) \Delta t + \sigma(S_t, t) \sqrt{\Delta t} \xi_t \right),
\]

where we assume that \( \Delta t \) is small enough (e.g. less than a week) and \( \xi_t \sim N(0, 1) \) i.i.d. Then we extract the local volatility surface points from (4.83) as

\[
\sigma_{K,T}(t + \Delta t, S) = \exp \left( \sqrt{v_{K,T}(t + \Delta t, S)} \right).
\]

The simulated local volatility surface is expressed in spline knot values. We apply the previously used bicubic spline parameterization with the simulated spline knots. Once we have the simulated value of the underlying asset, \( S_{t+\Delta t} \), and local volatility surface, \( \sigma_{t+\Delta t}(K,T) \), we use the method described in Section 4.1.3 to price all options in the portfolio, \( \{C^p_i(t, S_i)\}_{i=1}^{N_{\text{port}}} \), and then calculate the portfolio value using (4.80).

\[
4.3.2 \text{ Kernel Density Estimator}
\]

The method described in Section 4.3.1 is repeated \( n_{\text{sim}} \) times to generate a set of simulated portfolio values at time \( t + \Delta t \), \( \{V_{t+\Delta t, i}\}_{i=1}^{n_{\text{sim}}} \). We use a kernel density estimator to calculate the PDF \( \hat{f}_t(V_{t+\Delta t}) \) at time \( t \), from the simulated portfolio values. We use a Gaussian kernel which gives the estimator

\[
\hat{f}_t(V_{t+\Delta t}) = \frac{1}{\sqrt{2\pi} h n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} e^{-(V_{t+\Delta t} - \tilde{V}_{t+\Delta t,k})^2 / (2h^2)},
\]

where \( h \) is the bandwidth. We choose the bandwidth as

\[
h = \left( \frac{4\hat{\sigma}^5}{3n_{\text{sim}}} \right)^{\frac{1}{5}} \approx 1.06\hat{\sigma}(n_{\text{sim}})^{-\frac{1}{5}},
\]

where \( \hat{\sigma} \) is the sample standard deviation of the simulated values. This choice of bandwidth is the optimal choice if the the actual underlying density is Gaussian (Silverman 1986). The actual underlying density is probably not Gaussian, but in lack of information we make this choice anyway, since it gives a convenient analytical solution to the CDF. We choose \( n_{\text{sim}} = 100 \). With an estimator for the PDF, we calculate an estimation of the CDF, \( \hat{F}_t(V_{t+\Delta t}) \), by the integration

\[
\hat{F}_t(V_{t+\Delta t}) = \int_{-\infty}^{V_{t+\Delta t}} \hat{f}_t(x) dx \approx \frac{1}{n_{\text{sim}}} \sum_{k=1}^{n_{\text{sim}}} N_{0,1} \left( \frac{V_{t+\Delta t} - \tilde{V}_{t+\Delta t,k}}{h} \right)
\]
4.4 Evaluating Model Performance

The model performance will be evaluated out-of-sample. The task is to measure the probability distribution for the value of a simple portfolio consisting of one equity index option time \( \Delta t \) from the current time. We evaluate for different values of \( \Delta t \), namely \( \Delta t = \frac{1}{252}, \frac{2}{252}, \frac{3}{252}, \frac{4}{252}, \frac{5}{252} \), representing 1, 2, 3, 4, and 5 trading days into the future respectively.

The local volatility model will be calibrated over the period 2011-01-01 to 2011-01-31, which we call the calibration period \( (T_c) \), and evaluated during the period 2011-02-01 to 2011-07-01, which we call the evaluation period \( (T_e) \). This results in 20 trading days for calibration and 106 trading days for evaluation. The choice of such short periods of calibration and evaluation is made simply due to computational time. The local volatility surface has to be constructed for 126 dates, which results in a runtime of approximately one week. The accuracy of the measured probability distribution will be assessed by using the Inverse Transform Method and Q-Q Plots.

4.4.1 The Portfolio

The model is tested for a simple portfolio consisting of one option that is chosen to be at-the-money and with a time to maturity of approximately 0.5 years. For each evaluation date the option is chosen to fit the description as good as possible so that the portfolio is as similar as possible from day to day. It is fairly easy to match the strike price to be approximately at-the-money but it is harder to find an option that has a maturity of exactly 0.5 years since there is only one maturity date per month on the observed S&P 500 options. Therefore, the at-the-money option with maturity closest to 0.5 years is chosen at each date in the evaluation period.

4.4.2 Data

The local volatility surface is constructed for the equity index S&P 500 using observable option prices for European calls and puts for the calibration and evaluation period. The maturities of the contracts are limited to two years, i.e. \( \tau_{\text{max}} = 2 \). The option data is accessed through Thomson Reuters Eikon Datastream using the constituent lists presented in Table 1 from The Options Price Reporting Authority (OPRA). The data set consists of monthly options that have a time of maturity at the third Friday every month after January 2016, before that the time of maturity is the third Saturday of the month. This implies that the subset of the data set used for the calibration period and evaluation period will have times of maturity at the third Saturday of the month. We use only monthly options. There are weekly options on the S&P 500 but we have not managed to retrieve any data for them.

Typically, in-the-money (ITM) options are not as liquid as out-of-the-money (OTM) options and ITM also has an intrinsic value. In this thesis, OTM put options \( (K < S) \) will be used with the put-call parity (see e.g. (3.54) and (4.47)) to get the corresponding ITM call \( (K < S) \) option prices. This fixes potential problems due to illiquidity and errors in intrinsic values. The constituent lists of futures contracts for the calibration and evaluation period are presented in Table 2 and will also be accessed through Thomson Reuters Eikon Datastream.

![Table 1: Name of constituent lists for European call and put options.](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAQAAAC1HAwCAAAAC0lEQVR42mNkYAAAAABJ cumIAAcagAAAABJRU5ErkJggg==)

![Table 2: Name of constituent lists for future contracts.](data:image/png;base64,iVBORw0KGgoAAAANSUhEUgAAAAEAAAABCAQAAAC1HAwCAAAAC0lEQVR42mNkYAAAAABJ cumIAAcagAAAABJRU5ErkJggg==)

The data needed to model the deterministic risk-free interest rate for both the calibration period and the evaluation period is retrieved from Thomson Reuters Eikon. We use quoted prices for Overnight Index Swaps (OIS) with RIC codes \( \text{USD1MOIS=ICAP} \) and \( \text{USD1YOIS=ICAP} \) etc. for monthly and yearly maturities respectively. The available maturities are 1-11M, 1Y, 15M, 18M, 21M, 2-12Y, 15Y, 20Y, 25Y, and 30Y. However, prior to
April 2012, there were only contracts quoted with maturities up to 2Y. This is not a problem since we limit the
time-to-maturity to 2Y for the constructed local volatility surface. Lastly, the price of the underlying index is
retrieved from Thomson Reuters using the RIC code .SPX and we use the closing price.

There is a problem with the historical data for the underlying index price and option prices. The option market
closes about 5 minutes before the closing of the index. This poses a problem since the time series are not
synchronized, meaning that our measured dividend structure might be faulty. We note that this issue exists and
that there is little to do about it, since historical data is not easily available for specific times during the day.
The problem does not have a significant impact however since the error in time is absorbed by the estimated
dividend structure.

4.4.3 Inverse Transform Method and Q-Q Plots

Let the observed value of the portfolio at time $t$ be denoted by

$$V^\text{obs}_t = \sum_{i=1}^{N_p} h_i^p C_i^\text{obs}(t, S_t), \quad (4.88)$$

where $\{C_i^\text{obs}(t, S_t)\}_{i=1}^{N_p}$ are the portfolio’s observable option prices at time $t$. Using the proposed method, we
calculate

$$U_t = \tilde{F}_t(V^\text{obs}_{t+\Delta t}) = \int_{0}^{V^\text{obs}_{t+\Delta t}} \tilde{f}_t(x)dx, \quad t \in T^e. \quad (4.89)$$

If $\tilde{F}_t$ is a good approximation of the CDF of $V_{t+\Delta t}$ for all $t \in T^e$, the Inverse Transform Method says

$$U_t \sim U(0, 1), \quad t \in T^e, \quad (4.90)$$

i.e. that $U_t$ is uniformly distributed on $[0, 1]$. Once again, by the Inverse Transform Method we have that

$$Y_t = N_0^{-1}(U_t) \sim N(0, 1), \quad t \in T^e. \quad (4.91)$$

We use this to compare observations $Y_t$, sorted in ascending order, to quantiles of the standard normal distri-
bution in a Q-Q Plot. If a straight line with slope 1 that crosses through the origin is observed in the Q-Q Plot,
we can say that the measured probability distribution accurately describes the true probability distribution. It
is also possible to identify flaws in the model, for example if it measures a wrongly skewed distribution or if the
tails are too fat.
5 Results and Analysis

In this section, the results that are relevant to answer the purpose of the thesis are presented. We show results from constructing the local volatility surface, the Principal Component Analysis and the measurement of the probability distribution. The text consists of a combined exposition and analysis of the results which lays ground for a deeper discussion in Section 6.

5.1 Constructing the Local Volatility Surface

As mentioned, when we construct the local volatility surface, the objective is to fit it to observable option prices and obtain a realistic surface. In this section we comment on the resulting local volatility surface’s property of being realistic, in terms of skew structure, term structure and smoothness, as well as how consistent it is with observable option prices.

The reconstructed local volatility surface for 2011-01-14 is showed in Figure 5. It is considered a "good date" because the optimization solver has converged to a solution that shows a general appearance that seems realistic. In Figure 6 the local volatility surface on 2011-01-20 is displayed. It is considered a "bad date" since the optimization solver converged to a solution that does not show an appearance that seems realistic. Both dates exhibit a smooth local volatility surface which is expected since we use spline parameterization. A desirable property since it means that the expected variance of the underlying asset is not significantly different for a slight difference in future time point or value of the underlying asset. If we remember (3.11), the interpretation of the local variance as the conditional expectation of the future variance, it strengthens the analysis that the surface should be smooth to be realistic.

Figure 5: Local volatility for 2011-01-14, a "good" date.
5.1.1 Skew and Term Structure

One reason that makes the surface on the good date more realistic is noted if we study the skew structure for different times to maturity, as displayed in Figure 7, for the two dates. The general appearance of a local volatility surface should be so that the volatility smile is flatter as time to maturity increases. On the left, i.e. on the good date, we see that as time to maturity is close to 0 the smile is steeper than for 0.25 years and it gradually becomes flatter up until the maximum time to maturity of 1 year. On the right however, i.e. for the bad date, we see a similar steep smile for the shortest time to maturity but an irregular pattern for the rest of the lines. For 0.25 years the smile is flatter for high strikes than 0.5 years and 1 year, which is somewhat unrealistic, once again by the interpretation of (3.11).

The good date shows a clear skew structure around a strike level which is expected, but not the at-the-money strike level which is somewhat surprising. There do often exist a certain offset when looking at the implied volatility in the market for equity index options, which can give an implication that there might be for local volatility as well. However, the offset shown here seems too large and somewhat unrealistic. Additionally, the local volatility at-the-money is around 0.15-0.20 which can be deemed reasonable, but it is not quite as reasonable that the local volatility approaches zero for higher moneyness to later increase again. In some cases we obtain local volatility values for short time to maturity that are negative, which is unrealistic as well. We are convinced that this is a result of the spline knot placement. In Figure 8 we illustrate the local volatility skew of the good date with marked levels for the strike knots as well as the at-the-money strike level. The choice of spline knot placement is made with inspiration from the choice of Coleman et al. (1998). Given the nature of spline interpolation, in hindsight, it might be better to place the strike knots symmetrically around at-the-money or alternatively place one knot at-the-money. With our choice it might be hard for the optimization solver to force the at-the-money volatility level to be correct and at the same time making it the minimum.
G. Dackner & L. Falk

RESULTS AND ANALYSIS

Figure 7: Local volatility skews for different maturities, "good" vs. "bad" date.

Figure 8: Local volatility skew and strike knots placement at moneyness levels [0.6 0.93 1.26 1.6].

In Figure 9 we compare the term structure for different levels of moneyness between the two dates. It is not entirely straightforward to analyze term structure isolated since it is affected by and can intervene with a realistic flattening forward skew. One significant difference is that on the right, the term structure for most moneyness levels do not flatten out in the end which they do on the left. It is more realistic with a flattening term structure since the expected volatility for longer maturities shouldn’t vary a lot.
RESULTS AND ANALYSIS

5.1.2 Pricing Error

In Figure 10 we illustrate the pricing error for the two dates. It is hard to illustrate the pricing error in other ways than through difference in implied volatility since the price of an option varies significantly from the strike prices on the lower end of the spectrum to those on the upper end. If we study the plot on the left first, we see that the prices along the dotted line representing the current index level, or at-the-money strike, the errors seem to be smaller than for strikes further out. This is a result of the choice of weights in the optimization problem, where we have chosen them so that repricing should be more accurate for more liquid options, which amount to options close to at-the-money and short maturities. However, comparing the different levels of time to maturity, it seems like pricing error is larger for shorter times than longer. This is not entirely true since the relation between option prices and implied volatility (i.e. the greek Vega) is very small for extremely short maturities and extreme strikes. This means that a small difference in pricing error will result in a large difference in implied volatility. That is also why there are some extreme differences spotted in the edges of the plot. We provide some examples of pricing difference in Table 3 to show what such large differences in implied volatility for strike prices far from at-the-money translates to in USD. The last row in Table 3 however shows an example where the model has admitted a significant pricing error in both implied volatility and USD. That is an example of noise in the market data, where the ask-bid spread was very large and not representative of a realistic option price. Thus, that specific observation received a smaller weight in the optimization problem. Looking at the pricing error for the good date, we can note that the majority of the pricing errors are positive, meaning that our model underprices the options. This indicates that it is problematic with spline parameterization and optimization, i.e. it is hard to find a solution that reprices options precisely when a predetermined shape of the local volatility surface is assigned.

Now if we observe the right plot instead we see that the pattern is somewhat different. Firstly, all options with time to maturity around 0.4 years with strike price less than at-the-money have been overpriced instead of underpriced. Secondly, there are more extreme implied volatility errors for the first time to maturity. That might be because the date 2011-01-20 is the third Thursday in January, which means that the first maturity is two days from that date. This entails that the model seems to have problems with handling options that have extremely short time to maturity. With this insight in mind, we show a plot of the local volatility surface three trading days after the bad date, i.e. 2011-01-24, where the first occurring time to maturity is approximately one month, in Figure 11, and corresponding pricing error in Figure 12. The surface has returned to a more realistic appearance similar to that on the good date. However, the pricing error have another pattern that differs from the previously shown dates. On 2011-01-24 options with the two longest times to maturity tend to be overpriced. The model seems to be inconsistent with over -or underpricing options with similar strike price and time to maturity from day to day.
Figure 10: Pricing error expressed as difference in implied volatility (observed IV - repricing IV), "good" vs. "bad" date.

Table 3: Examples of pricing errors in USD instead of Implicit Volatility, compare these with Figure 10 on the left.

<table>
<thead>
<tr>
<th>Strike($)</th>
<th>Maturity(yrs)</th>
<th>Observed price ($)</th>
<th>Calculated price($)</th>
<th>IV-diff</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>0.67</td>
<td>1176.82</td>
<td>1176.18</td>
<td>0.9754</td>
<td>0.74</td>
</tr>
<tr>
<td>200</td>
<td>0.42</td>
<td>1082.68</td>
<td>1082.66</td>
<td>0.50</td>
<td>0.99</td>
</tr>
<tr>
<td>650</td>
<td>0.02</td>
<td>642.78</td>
<td>642.72</td>
<td>0.49</td>
<td>0.99</td>
</tr>
<tr>
<td>1000</td>
<td>0.92</td>
<td>272.64</td>
<td>298.87</td>
<td>-0.28</td>
<td>0.07</td>
</tr>
</tbody>
</table>

Figure 11: Local volatility for 2011-01-24, i.e. two days after the "bad" date.
5.2 Analyzing the Systematic Changes of the Local Volatility Surface

The Principal Component Analysis (PCA) resulted in a set of eigenvectors and eigenvalues that could be used to select the number of principal components needed to explain a certain fraction of the variance. In Table 4 we present the explained fraction for each of the sixteen principal components and the cumulative explained variance for each choice of the variable $k_{\text{PCA}}$ in (4.73). To reach the target ratio of explained variance we thereby chose $k_{\text{PCA}} = 7$, which resulted in an explained variance of 99.72%. We did not expect to need 7 components, since there are examples from doing PCA on interest rate yield curves where 3 principal components are enough to explain over 99% of the variance. We believe that the reason for a larger number is because the PCA was conducted on a surface, i.e. two dimensional data, as opposed to the one dimensional data of an interest rate yield curve.

<table>
<thead>
<tr>
<th>Principal Component</th>
<th>Explained Variance</th>
<th>Accumulated Explained Variance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>62.99%</td>
<td>62.99%</td>
</tr>
<tr>
<td>2</td>
<td>14.02%</td>
<td>77.00%</td>
</tr>
<tr>
<td>3</td>
<td>9.43%</td>
<td>86.43%</td>
</tr>
<tr>
<td>4</td>
<td>5.32%</td>
<td>91.75%</td>
</tr>
<tr>
<td>5</td>
<td>3.90%</td>
<td>95.65%</td>
</tr>
<tr>
<td>6</td>
<td>3.11%</td>
<td>98.76%</td>
</tr>
<tr>
<td>7</td>
<td>0.95%</td>
<td>99.72%</td>
</tr>
<tr>
<td>8</td>
<td>0.16%</td>
<td>99.88%</td>
</tr>
<tr>
<td>9</td>
<td>0.05%</td>
<td>99.93%</td>
</tr>
<tr>
<td>10</td>
<td>0.03%</td>
<td>99.96%</td>
</tr>
<tr>
<td>11</td>
<td>0.02%</td>
<td>99.98%</td>
</tr>
<tr>
<td>12</td>
<td>0.01%</td>
<td>99.99%</td>
</tr>
<tr>
<td>13</td>
<td>0.01%</td>
<td>100.00%</td>
</tr>
<tr>
<td>14</td>
<td>0.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>15</td>
<td>0.00%</td>
<td>100.00%</td>
</tr>
<tr>
<td>16</td>
<td>0.00%</td>
<td>100.00%</td>
</tr>
</tbody>
</table>

In Table 5 we present the seven largest eigenvalues and their corresponding eigenvector. The interpretation of the eigenvectors are sensitivities against corresponding principal component (i.e. systematic change) for the point on the logarithmic local variance grid that the specific index in the vector represents. If we look back at how the PCA is performed, especially looking at (4.69) and (4.70), the interpretation becomes clearer. It
is hard to draw any specific conclusion regarding the direct systematic changes of the local volatility surface since the PCA is done on logarithmic local variance, but there are some things to note by just looking at the eigenvectors.

Table 5: The seven largest eigenvalues and corresponding eigenvectors.

<table>
<thead>
<tr>
<th>Eigenvalues</th>
<th>11.5665</th>
<th>2.5739</th>
<th>1.7311</th>
<th>0.9767</th>
<th>0.7170</th>
<th>0.5708</th>
<th>0.1753</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvectors</td>
<td>-0.0083</td>
<td>0.0099</td>
<td>-0.0236</td>
<td>-0.0096</td>
<td>0.0096</td>
<td>0.0146</td>
<td>-0.1291</td>
</tr>
<tr>
<td></td>
<td>-0.0079</td>
<td>0.1212</td>
<td>0.0602</td>
<td>-0.0564</td>
<td>-0.2141</td>
<td>-0.3837</td>
<td>0.6171</td>
</tr>
<tr>
<td></td>
<td>-0.0487</td>
<td>0.0265</td>
<td>0.1439</td>
<td>0.1194</td>
<td>-0.0945</td>
<td>-0.1669</td>
<td>0.1186</td>
</tr>
<tr>
<td></td>
<td>0.8203</td>
<td>0.0615</td>
<td>0.1921</td>
<td>-0.1765</td>
<td>-0.4608</td>
<td>-0.0680</td>
<td>-0.1885</td>
</tr>
<tr>
<td></td>
<td>-0.0114</td>
<td>-0.0054</td>
<td>-0.0116</td>
<td>-0.0057</td>
<td>-0.0036</td>
<td>0.0247</td>
<td>-0.0211</td>
</tr>
<tr>
<td></td>
<td>0.0152</td>
<td>0.0111</td>
<td>0.0068</td>
<td>0.0061</td>
<td>-0.0104</td>
<td>-0.0174</td>
<td>0.1182</td>
</tr>
<tr>
<td></td>
<td>-0.0142</td>
<td>0.0257</td>
<td>0.0276</td>
<td>0.0020</td>
<td>-0.0065</td>
<td>0.0148</td>
<td>-0.0350</td>
</tr>
<tr>
<td></td>
<td>0.5121</td>
<td>-0.2458</td>
<td>0.0347</td>
<td>0.1948</td>
<td>0.6053</td>
<td>0.3347</td>
<td>0.3586</td>
</tr>
<tr>
<td></td>
<td>0.0005</td>
<td>0.0015</td>
<td>0.0028</td>
<td>-0.0008</td>
<td>0.0028</td>
<td>-0.0004</td>
<td>-0.0010</td>
</tr>
<tr>
<td></td>
<td>0.0849</td>
<td>0.0071</td>
<td>-0.0716</td>
<td>0.1144</td>
<td>0.0378</td>
<td>-0.3598</td>
<td>0.4213</td>
</tr>
<tr>
<td></td>
<td>0.0105</td>
<td>0.0037</td>
<td>-0.1811</td>
<td>-0.1255</td>
<td>-0.1008</td>
<td>-0.0041</td>
<td>0.2890</td>
</tr>
<tr>
<td></td>
<td>0.0601</td>
<td>0.6879</td>
<td>0.2963</td>
<td>-0.2730</td>
<td>0.5181</td>
<td>-0.2526</td>
<td>-0.1222</td>
</tr>
<tr>
<td></td>
<td>-0.0005</td>
<td>-0.0005</td>
<td>-0.0004</td>
<td>-0.0002</td>
<td>-0.0006</td>
<td>-0.0002</td>
<td>0.0009</td>
</tr>
<tr>
<td></td>
<td>0.1850</td>
<td>-0.1963</td>
<td>-0.4647</td>
<td>0.2651</td>
<td>0.2288</td>
<td>-0.6462</td>
<td>-0.3500</td>
</tr>
<tr>
<td></td>
<td>0.0874</td>
<td>0.1554</td>
<td>-0.6999</td>
<td>-0.5818</td>
<td>0.0378</td>
<td>0.1708</td>
<td>0.0967</td>
</tr>
<tr>
<td></td>
<td>-0.0947</td>
<td>-0.6193</td>
<td>0.3232</td>
<td>-0.6334</td>
<td>0.1801</td>
<td>-0.2586</td>
<td>-0.0304</td>
</tr>
</tbody>
</table>

The first eigenvector (furthest to the left) corresponds to the principal component explaining 62.99% of the total variance in the surface. We see that element 4 and 8 are significantly larger than the rest of the elements. That means that the logarithmic variance on the two spline knot volatilities representing moneyness 0.6 and 0.93 and time to maturity 1.5 years are much more sensitive to the first principal component than the rest of the knot points. In Figure 13 we illustrate the sensitivities towards the first principal component for each logarithmic local variance point representing a spline knot. We believe the reason that the spline knot values for maturities 1.5 years are subject to large variation is simply due to the input data. The observable options in the available data for the calibration period all have a time to maturity less than or equal to 1 year. With no information in the range 1-1.5 years to maturity the knot values for 1.5 years are subject to find significantly different values from date to date when optimizing the surface. The same phenomenon can actually be seen through eigenvector 2-3 as well, as seen in Figure 14 and Figure 15. It seems as if the knot values for maturity 1.5 years are generally more sensitive than the shorter maturities, both negative and positive sensitivity that is.

For all four spline knots with maturity 0, i.e. the shortest, the sensitivities are relatively close to zero for the first three principal components. That, in symbiosis with previous observations, would imply that the skew structure for longer time to maturity is more prone to vary from day to day than the skew structure for short time to maturity.
Figure 13: Plot of the first eigenvector as a surface.

Figure 14: Plot of the second eigenvector as a surface.
Figure 15: Plot of the third eigenvector as a surface.

In Figures 16 and 17 we illustrate the principal components’ values over the calibration period. They are all notably "jumping" between different levels, especially PC1. That means that there has been significant variation in the input data, i.e. the local volatility surface. The "jumping" property implies that the auto correlation is significant, i.e. far from 0. This can be seen in Table 6 where we show the auto correlation of each principal component for different lags.
Figure 16: Principal components 1-3 over the calibration period.

Figure 17: Principal components 4-7 over the calibration period.
Table 6: Principal components’ auto correlation for different lags

<table>
<thead>
<tr>
<th>Lag</th>
<th>PC1</th>
<th>PC2</th>
<th>PC3</th>
<th>PC4</th>
<th>PC5</th>
<th>PC6</th>
<th>PC7</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0.05</td>
<td>0.10</td>
<td>0.20</td>
<td>-0.12</td>
<td>-0.01</td>
<td>0.26</td>
<td>0.24</td>
</tr>
<tr>
<td>2</td>
<td>-0.09</td>
<td>0.16</td>
<td>-0.05</td>
<td>-0.39</td>
<td>-0.07</td>
<td>0.21</td>
<td>0.19</td>
</tr>
<tr>
<td>3</td>
<td>-0.02</td>
<td>-0.04</td>
<td>0.27</td>
<td>0.01</td>
<td>0.11</td>
<td>-0.21</td>
<td>0.19</td>
</tr>
<tr>
<td>4</td>
<td>0.35</td>
<td>-0.08</td>
<td>0.10</td>
<td>-0.07</td>
<td>-0.05</td>
<td>-0.27</td>
<td>-0.09</td>
</tr>
<tr>
<td>5</td>
<td>0.10</td>
<td>-0.34</td>
<td>0.03</td>
<td>0.37</td>
<td>-0.23</td>
<td>-0.35</td>
<td>-0.03</td>
</tr>
<tr>
<td>6</td>
<td>-0.22</td>
<td>0.05</td>
<td>0.07</td>
<td>-0.02</td>
<td>-0.12</td>
<td>-0.08</td>
<td>0.03</td>
</tr>
<tr>
<td>7</td>
<td>0.06</td>
<td>-0.04</td>
<td>0.02</td>
<td>-0.26</td>
<td>-0.06</td>
<td>0.13</td>
<td>-0.30</td>
</tr>
<tr>
<td>8</td>
<td>0.11</td>
<td>-0.03</td>
<td>0.07</td>
<td>-0.03</td>
<td>0.00</td>
<td>0.17</td>
<td>-0.18</td>
</tr>
<tr>
<td>9</td>
<td>-0.31</td>
<td>-0.08</td>
<td>-0.20</td>
<td>-0.04</td>
<td>0.04</td>
<td>0.15</td>
<td>-0.36</td>
</tr>
<tr>
<td>10</td>
<td>-0.28</td>
<td>0.04</td>
<td>-0.31</td>
<td>0.29</td>
<td>-0.18</td>
<td>0.14</td>
<td>-0.25</td>
</tr>
<tr>
<td>11</td>
<td>0.00</td>
<td>-0.15</td>
<td>0.00</td>
<td>-0.08</td>
<td>0.27</td>
<td>0.00</td>
<td>-0.08</td>
</tr>
<tr>
<td>12</td>
<td>0.10</td>
<td>-0.17</td>
<td>-0.10</td>
<td>-0.31</td>
<td>0.09</td>
<td>-0.08</td>
<td>-0.11</td>
</tr>
<tr>
<td>13</td>
<td>-0.06</td>
<td>-0.05</td>
<td>-0.25</td>
<td>0.19</td>
<td>-0.06</td>
<td>-0.24</td>
<td>-0.09</td>
</tr>
<tr>
<td>14</td>
<td>-0.31</td>
<td>-0.07</td>
<td>0.00</td>
<td>-0.06</td>
<td>0.01</td>
<td>-0.01</td>
<td>0.09</td>
</tr>
<tr>
<td>15</td>
<td>0.09</td>
<td>0.07</td>
<td>-0.19</td>
<td>-0.05</td>
<td>0.11</td>
<td>-0.21</td>
<td>-0.05</td>
</tr>
<tr>
<td>16</td>
<td>-0.04</td>
<td>0.04</td>
<td>-0.18</td>
<td>0.06</td>
<td>-0.13</td>
<td>-0.05</td>
<td>0.09</td>
</tr>
<tr>
<td>17</td>
<td>0.04</td>
<td>0.04</td>
<td>0.01</td>
<td>-0.05</td>
<td>-0.06</td>
<td>-0.03</td>
<td>0.20</td>
</tr>
<tr>
<td>18</td>
<td>-0.10</td>
<td>0.04</td>
<td>0.01</td>
<td>0.03</td>
<td>-0.04</td>
<td>0.01</td>
<td>0.05</td>
</tr>
<tr>
<td>19</td>
<td>0.05</td>
<td>0.00</td>
<td>0.00</td>
<td>0.02</td>
<td>-0.12</td>
<td>-0.02</td>
<td>-0.02</td>
</tr>
</tbody>
</table>

In Figure 18 we present the local volatility surface from 6 adjacent days of the 20 days that were analyzed with PCA. The previously mentioned variability in the local volatility surface is clearly apparent. We see that the skew structure for longer time to maturity varies from day to day significantly more than the skew structure for short time to maturity.
5.3 Model Performance Measuring the Probability Distribution

The probability distribution of the future value of a portfolio was measured and the results show that the model could accurately capture the true distribution, although with an overestimated volatility. Figures 19, 20 and 21 show Q-Q plots of the measured distribution for $\Delta t = \frac{1}{252}, \frac{2}{252}, \frac{3}{252}, \frac{4}{252}, \frac{5}{252}$, i.e. ranging from 1 to 5 trading days from the current date. The vertical axis show "Normal data quantiles" which are calculated by normal inverse transformation of the sorted out-of-sample measured CDF values. They are compared to the theoretical
normal quantiles on the horizontal axis. The simulations for the CDF measurements are done with new random numbers each run.

Figure 19: Q-Q plots for the measured probability distribution for $\Delta t = \frac{1}{252}$ and $\Delta t = \frac{2}{252}$.

Figure 20: Q-Q plots for the measured probability distribution for $\Delta t = \frac{3}{252}$ and $\Delta t = \frac{4}{252}$. 
Figure 21: Q-Q plots for the measured probability distribution for $\Delta t = \frac{5}{252}$.

If the blue graph followed the red line, the result would mean that the model perfectly measured the true probability distribution. We can observe that the blue graph consistently cuts the red line slightly below the origin, which means that the mean is somewhat accurate for all the time-frames. The graph also looks linear for the first three plots, which means that the distribution is correct. The final two plots look a bit more distorted with some variation in the left end, but the common trend infers that the distribution was accurately captured. Lastly, the blue graph has less slope than the red, which means that the measured distribution has a larger variance than the true distribution. This result is not surprising, since the model is calibrated to market data, which is known to overestimate the volatility when compared to the realized, causing a negative volatility premium in the market.
6 Discussion

In this section we discuss mainly what the results and analysis lead to and what conclusions can be drawn. Furthermore we discuss some of the challenges we have faced as well as some ethical aspects.

6.1 The Local Volatility Surface

The main objective when constructing the local volatility surface was to obtain realistic results that were consistent with observable market data, to use as input to the analysis of the systematic changes. By analyzing the resulting local volatility we can confirm that the method we chose and implemented can achieve that objective to a certain extent. The local volatility surface has a realistic appearance in terms of smoothness but have an offset from at-the-money strike level in the skew structure. It is consistent with observable market data to the extent where it is optimized to be more consistent with option prices for options that are liquid and thereby contain more information. The realistic properties don’t hold for all dates and there is a varying irregular pattern in pricing error depending on date and what option we choose to price. The inconsistency can partly be explained by the choice of method.

The finite difference method approach for pricing options do have its limitations. When discretizing the spectrum of possible future values of the underlying asset and future time points into a grid there is an error that cannot be avoided. When pricing the options, strike prices and times to maturity will be mismatched with the grid points. We did match the options’ time to maturity but not the strike level, which is an obvious source of pricing error.

Another choice of method that introduces both pricing errors and impairment of the local volatility surface’s appearance is the parameterization approach. By using a pre-decided parameterization, in our case a bicubic spline, we decide beforehand the shape of the local volatility surface. A non-parametric approach is generally preferred. The bicubic spline also has a somewhat bad feature that if one knot value changes, the entire surface will change. This should be a problem for the optimization solver and make it harder to converge to a solution with precise repricing of options, or at least require more time and iterations. A third problem with the problem formulation of parameterizing the local volatility surface and minimizing the pricing error is that there do exist local optimas. We only investigated this for the first date of the calibration period and made sure that we found a global optima, but there is a chance that local optimas are found for other dates. That could cause the local volatility surface to look strange and unrealistic, which would contribute to noisier input to the Principal Component Analysis.

But the varying quality of the local volatility surface over different dates can not only be explained by choices of method. It can also be explained by some practical issues such as choice of parameters. For example the discretization distance in the finite difference grid, the number of spline knots, placement of the knots and choice of weights for the optimization problem.

When discretizing the finite difference grid, the choice of distance in both dimensions is done by weighing pricing accuracy versus computational time. We chose $N = 299$, i.e. 301 strike levels in the grid. With more computational power we could choose a finer discretization of the grid and it would most certainly result in more accurate pricing of options over the whole strike-maturity spectrum.

When choosing the number of spline knots, computational time is also the limiting factor. We are certain that the pricing error would decrease with more spline knots, since it would give the optimization solver more points on the surface to change to fit it to the market data. However, it is unclear how it would affect the appearance of the surface. Intuitively it could become more oscillating or “bumpy”, given the nature of the bicubic spline. When studying the pricing error we could not see a clear pattern from day to day, but we could see some patterns for each specific date. There seem to appear certain areas on the strike-maturity grid where almost all options are overpriced as opposed to the majority of the others being underpriced. It is a reasonable result since there are so few spline knots. If one knot is placed so that overpricing is done it will affect a large area of the grid in the same way resulting in the pattern we observe. Unfortunately we are not able to experiment with the number of knots used due to computational time. Regarding the placement of the spline knots, it is done arbitrarily with inspiration from the literature. We realize we should try to change the knot placement in mainly two ways. Firstly, place them symmetrically around at-the-money strike to possibly achieve a more realistic skew structure that isn’t as offset from the at-the-money strike level. Secondly, the time to maturity knot-levels should be placed so that they are only covering the area of the grid where there is available information, i.e.
observable option prices. The longest maturity for the spline knots is 1.5 years which is not that suitable for the time period used for calibration and evaluation, since the longest maturities in the market during that time period is 1 year. However, if the model were to be used for arbitrary data, for example from 2019, the maximum time to maturity is up to 2 years. Choosing the knot placement and other design choices based on what data is available isn’t really a tenable strategy since it makes the model overfitted and forces results to be good. In our case it doesn’t matter since the knots have to be placed at the same level of moneyness and time to maturity to make the Principal Component Analysis possible over a sufficient time period.

The choice of weights in the optimization problem should also have significant impact on the resulting local volatility surface. We see from the resulting pricing error that the weights serve their purpose, giving more weight to more liquid options, i.e. options close to at-the-money and with short time to maturity. However, there might be other weighting strategies that minimize the overall pricing error. Once again there is no time to experiment due to the computational cost of constructing the local volatility surface. We do check for a couple of dates to see if the weights fulfill their purpose, but a more rigorous evaluation should be done. An alternative approach would be to fully exclude market data fulfilling some criteria of uncertainty. For example if the moneyness exceeds or undercut certain levels.

There is also the aspect of varying properties of the market data, which should pose an impact to the measured local volatility surface. Pricing options with short time to maturity requires a steep skew structure as opposed to pricing options with longer time to maturity. The shortest time to maturity in the used data set varies from 1 day to 1 month, which should cause the skew structure to vary heavily from day to day as a batch of options mature. If we use market data for weekly options as well we could obtain a more homogeneous data set which could result in a more stable input of local volatility surface variation to the Principal Component Analysis.

The last issue with the constructed local volatility surface is the possibility of negative local volatility for short time to maturity. In Figure 8 we see that the lowest level of local volatility is close to 0. For some dates, negative local volatility appear in the solution from the optimization. Even if negative local volatility is unrealistic it doesn’t cause any problems in the option pricing method. Study the process for the underlying asset,

\[
\frac{dS_t}{S_t} = (r - q)dt + \sigma(S_t, t)dZ_t^Q,
\]

in the market model for local volatility. Since the Weiner process, \(Z_t^Q\), varies independently of sign, the sign of the local volatility, \(\sigma(S_t, t)\) is irrelevant. Additionally, when using the finite difference approach to price options, the local volatility is converted to local variance \(\sigma(S_t, t)^2\) which also makes the sign irrelevant. Thus it makes it plausible that the optimization solver finds a solution with negative local volatility that still reprices options accurately. Imposing additional constraints in the optimization problem is one alternative fix to the problem. However, the problem is already heavily constrained not only by the formulated constraints but also by the spline parameterization. Thus, adding additional constraints might cause more harm than good. It could also only have no effect or negative effect on minimizing the pricing error as it would be a contraction of the space of possible solutions. Another possible fix to this issue would be to measure the local variance directly instead.

### 6.2 Systematic Changes of the Local Volatility Surface

The aim of performing Principal Component Analysis is to find a reduced set of independent factors driving the systematic changes of the local volatility surface. The S&P 500 index has a lot of history and ideally you would want to input the surface from all previous dates in order to capture its long term behaviour and sensitivities to different kinds of systematic changes. In this thesis, we only managed to analyze the local volatility surface from 20 dates due to lack of computational power. The results though showed that it was possible to explain over 99% of the day to day variation with only 7 of the 16 principal components. In terms of simulating the surface for future dates, this result meant that the stochasticity could be explained with these 7 principal components.

The only interpretation regarding systematic changes of the local volatility surface that can be made by analyzing the resulting eigenvectors is that points on the surface with longer time to maturity seem to have a higher sensitivity towards the first three principal components. The first three principal components explain over 86% of the variance. This implies that the points with longer time to maturity are most prone to variation from day
to day. However, it is hard to validate if this observation is correct since the Principal Component Analysis is only done over a period of 20 dates. It might also be the result of lack of information in the market data for options with longer time to maturity than 1 year for that specific time period.

If time and the speed of our computers allowed, the systematic changes would have been studied for a much longer time period, which could capture more of the systematic changes and allow for more accurate simulations. However, since the principal components show auto correlation, we cannot guarantee that the simulations would become more accurate. We discuss the auto correlation in the principal components more in Section 6.3.

6.3 Accuracy of the Measured Probability Distribution

The result from the Q-Q plots is that the shape of the measured probability distribution is correct, but the standard deviation, or volatility, is overestimated. There are numerous possible reasons for this, but there are two main reasons we find most likely. Firstly, the principal components show significant auto correlation. Secondly, there is an expected difference between market consistent volatility and realized volatility of the underlying asset.

6.3.1 Auto Correlation in Principal Components

As was shown in Figure 18, the optimized local volatility surface tends to vary heavily from day to day. This means that the model has trouble finding the "correct" surface, which partly is due to noise in the market data, but mostly because the model for measuring the local volatility is not good enough, i.e. measurement noise. The noise in the measurement is present every day and the result is that the surface varies around the "optimal" solution that we try to measure. The phenomenon is visually explained in a simple two-dimensional case in Figure 22. Imagine the straight line is the true unobservable but measurable local volatility surface that we try to measure. The blue points and dotted line represents our measurements. Since the model is unable to find the line with certainty, the measurements are placed around the line with varying error size.

![Figure 22: Example of error in measurement.](image)

The result from a varying error in measurement leads to the presented "jumping" principal components, i.e. the same pattern as in Figure 22, which can be seen in Figures 16 and 17. This results in significant auto correlation in the principal components, implying that they are not independent of their previous states. Remember the process from (4.76),

\[
v_{t+\Delta t} = v_t + a_t + \sum_{i=1}^{k_{PCA}} \sqrt{\lambda_i} q_{t,i} \xi_{t,i}.
\]

The reliability of the simulation is reduced since it is based on the assumption that no auto correlation exist and each simulation from \( t \) to \( t + \Delta t \) can be done independently. This is clearly not the case and some bias in the simulation can be expected, reducing the explanatory ability of the PCA. This is an obvious flaw with the model and the measured probability distribution is directly affected by it.

6.3.2 Market Consistent Volatility versus Realized Volatility

We use market consistent volatility, i.e. the constructed local volatility surface, to simulate the underlying asset when simulating and measuring the probability distribution for the portfolio. The volatility of the underlying
asset will have a direct impact on the volatility of the portfolio value, since an option’s value is a function of time and the value of the underlying asset. We simulate the underlying asset from time $t$ to $t + \Delta t$ with a volatility that is market consistent at $t$. The resulting simulated value is input to the valuation of the portfolio that in turn is used as one of the simulated values for the kernel density estimation. It is well known that the market has a negative volatility risk premium meaning that the realized volatility, i.e. the volatility that actually occurred from time $t$ to $t + \Delta t$, will often be lower than the market consistent volatility at time $t$. We believe this phenomenon is a significant explaining factor to the resulting measurement of the probability distribution resulting in an overestimated volatility of the portfolio, which is seen in the Q-Q plots.

### 6.4 Implementation and Practical Issues

There are some practical issues that prohibit us from conducting a more thorough study, mostly related to implementation and computational cost. We choose to implement the model in Python since it is an easy and straightforward language. However, given the computational cost of some routines in our implementation, a more suitable language might be some compiled language such as C++ or C#. Another drawback with choosing Python is that all available packages for automatic differentiation are implemented using Operator Overloading, whereas there are no packages that use Source Code Transformation. This is a problem since Operator Overloading is much slower than Source Code Transformation. Implementing Operator Overloading on routines that are already computationally costly and with many variables introduces a significant overhead that results in long run times, despite numerous attempts to refactor the code to be faster. If implementation of at least some routines is done in for example C++ there would be packages such as Tapenade och ADIC, which both use Source Code Transformation. Faster run time and faster automatic differentiation would make it possible to run the model for more data and also give the opportunity for sensitivity analysis for choices of parameters. It would help calibrate the model to perform better. Another practical issue is handling market data for options, futures, and the underlying index price. Apart from missing data points, there are data points that are faulty for unknown reason. That can be for example a bid price being significantly larger or smaller than the ask price for the same option or the fact that the option, futures and stock market doesn’t close at the same time of day so that the data is always a little out of sync. The latter sometimes lead to strange values for e.g. dividend yield which makes it hard to evaluate the plausibility.

### 6.5 Time Limitation

There are improvements to this thesis that we would do if there was more time. One very central element of this thesis that we left out in the method is the no-arbitrage conditions for the drift terms in the dynamics of the local volatility surface. The reason is twofold.

Firstly, the condition we obtained from Derman and Kani (1997) that we also derive in Appendix A is expressed in the variables strike price $K$ and time of maturity $T$, which is problematic. Due to the expected property of the local volatility surface that the skew structure should be flatter (steeper) for longer (shorter) time to maturity, a more proper variable would be time to maturity $\tau = T - t$, where $t$ is the current date. To explain the difficulty, consider the current date $t_0$ and a point on the local volatility surface representing a time of maturity $T$ and time to maturity $\tau_0 = T - t_0$. Now, at $t_0$ the point of $T$ and $\tau_0$ coincide. But, if we are to simulate the local volatility surface $\Delta t$ into the future to time $t_1 = t_0 + \Delta t$ we still have the same time of maturity $T$ but that point will have a corresponding time to maturity $\tau_1 = T - t_1 = T - t_0 - \Delta t = \tau_0 - \Delta t$. So the point on the local volatility surface will have a $\Delta t$ shorter time to maturity and is no longer comparable to the point prior to the simulation. This is the reason we change the variable $T$ to $\tau$ for the Principal Component Analysis and also the reason why a change of variable is needed in the no arbitrage conditions. If the change of variable is not made, the future, i.e. simulated, volatility skew will be underestimated, i.e. too flat. We do not make the change of variable in the no-arbitrage condition since it needs a rigorous proof that we can not find any guidance for in the literature, which in turn makes it hard to finish within our time frame. Secondly, the limited time frame for this thesis lead us to realize that it would not only be hard to find time to derive the change of variable in theory, but also to implement it. Change of variable in- and implementation of the no-arbitrage drift conditions will probably add another two weeks to the project that is already running out of time.

The choice to simplify and use a zero drift of course has its drawbacks. The drift condition is the part of the local volatility dynamics that assure an arbitrage free model. Without it we simply find sensitivities to the
systematic stochastic changes of the local volatility but ignore the deterministic change that is needed for the underlying asset to be a martingale under the equivalent martingale measure. If the underlying asset is not a martingale then the equivalent measure is not a martingale measure and we can not guarantee that the market model is arbitrage free. It would be interesting to see the result of a similar model to ours where the no-arbitrage conditions are implemented. However, the drift resulting from the no-arbitrage conditions is probably relatively close to 0. The drift coefficients are for local variance which typically is not very large, since it is the squared local volatility. Therefore, we are certain that the previously mentioned issues of auto correlation in the principal components and the negative volatility risk premium will have a significantly larger impact on the accuracy of the measured probability distribution.

6.6 Inadequate Evaluation
The model is tested and evaluated for a very limited set of simulation horizons (1, 2, 3, 4 and 5 days) and for a very simple portfolio setup of one equity index option with time to maturity of 0.5 years. Since the pricing is more accurate for at-the-money options the model would probably perform worse if options further from at-the-money are included in the portfolio. Intuitively it should perform better for a portfolio of options with shorter time to maturity and vice versa. Although, the purpose of this model is depleted if it is used for very liquid options that are probably priced correctly by the market.

Another aspect that makes the evaluation inadequate is the amount of data used. Initially we wanted to calibrate the model, i.e. perform the PCA, over a two-year period and evaluate it over a seven-year period. This was reduced to one month and five months respectively due to practical issues previously mentioned.

6.7 Ethical Aspects
All of society is affected by the financial markets. One of many examples is the financial crisis that started in 2007 when blunt risk management and sloppy governance of securities’ fair value caused a collapse that made millions loose their jobs and businesses to go bankrupt. Thus, from an ethical aspect it is clearly important with sophisticated models for risk management, pricing, and governance. The model we have developed is sophisticated in theory, but as we have repeatedly stated in this thesis there are practical issues. Computational cost makes the model unstable since we cannot choose parameters freely. Noisy market data causes the output to be untrustworthy in some cases. There are also some method-related issues as well that causes irregular pricing errors. Thereby we would not recommend anyone to use this model for managing the risk or pricing derivatives in their portfolio without properly calibrating it. Sensitivity analysis for all choices of parameters and an adequate (longer) calibration period would perhaps make the model more reliable and ethically acceptable to use.

Additionally, we have only tested the model for a very simple portfolio setup consisting of only one equity index option. Even if the resulting Q-Q plots would have led us to a conclusion that we may have measured the probability distribution correctly or incorrectly, the result may vary to better or worse for a different or more complicated setup. Therefore, we cannot from an ethical standpoint claim that our model works well enough to be used in practice, where its performance could have such a large impact on several parts of society.

6.8 Suggestions for Future Studies
The purpose of this thesis is to measure the probability distribution for the future value of a portfolio consisting of equity index options. A sophisticated model for simulating the local volatility surface representing the distribution is theoretically laid out. This include deriving conditions making the simulation arbitrage free to mimic the real world. This part of the thesis was not successfully implemented in time which should impact the resulting distribution. It would be interesting to see how the model performs with the no-arbitrage conditions.

Our method to measure the probability distribution is not dependent on how the local volatility surface is constructed. Other methods to construct the surface can be chosen, as long as they are realistic and accurately prices options on the underlying index, and the result can work as input to the Principal Component Analysis. This sets up for future studies of comparing different methods. It would be interesting to see how the model performed if the local volatility surface were constructed using a sophisticated non-parametric model that is stable over time, see for example Barkhagen and Blomvall (2015).
To measure the probability distribution, we use a Kernel Density Estimator with a Gaussian kernel. One important parameter to take into account that affects the result is the bandwidth of the kernel. The choice we made was to use a bandwidth proven to be optimal for Gaussian distributed data. This is of course a flaw in our model since the financial market has shown to realize extreme events much more often than the normal distribution implies. We do not make any effort to try different bandwidths in the work of this thesis, but it would be interesting to see if it could improve the results.

To retrieve a result, we choose an evaluation period of five months which initially was supposed to be seven years. To see significant results and draw accurate conclusions of the model’s performance, a much longer period would need to be studied. This delimitation is mainly due to lack of computational power and time-frame of the thesis. Intuitively, the model should perform better with more calibration data than one month, but it ought to be tested.

6.9 Conclusions
To fulfill the purpose of the thesis, four main problems are solved. Firstly, we construct the local volatility for a set of historical dates. Secondly, we analyze the systematic changes of said local volatility surface to translate it to a stochastic process that describes how it changes over time. Thirdly, we use the resulting stochastic process to measure the probability distribution for a portfolio of equity index options. Lastly, we evaluate the model’s performance to determine how accurate the probability distribution is measured. Next we present the main conclusions that can be drawn and how they answer the purpose of the thesis.

The local volatility surface is constructed and consistent with market prices to an extent where it is more accurate for more liquid options. It is in most cases realistic with respect to smoothness, but have an unexpectedly large offset from the at-the-money strike level in the skew structure. It is unstable from date to date and also significantly dependent on choice of parameters, limited by computational power, and input data.

The systematic changes are analyzed through Principal Component Analysis of logarithmic local variance and translated into a stochastic process that describes the dynamics of the local volatility surface. However, the explanatory ability is impaired by auto correlation in the principal components.

The shape of the measured probability distribution is accurately measured under the simple evaluation conditions, i.e. simple portfolio, simplifications, and short evaluation period, but the volatility is overestimated. The evaluation is inadequate so it is not possible to draw any general conclusions regarding the performance of the model for a more general setting and without simplifications.

With these conclusions we deem the purpose of this thesis fulfilled. We have provided a theoretical model for measuring the risk-neutral probability distribution for a portfolio of equity index options using Derman and Kani’s (1997) local volatility model. We have implemented the model to test it under simplified conditions and evaluated its performance.
References


Derman, Emanuel et al. (1996b). The Local Volatility Surface: Unlocking the Information in Index Option Prices.


Appendix A  Proof of the No-arbitrage Drift Conditions

\[
\frac{d\sigma_{K,T}^2(t,S)}{\sigma_{K,T}^2(t,S)} = \alpha_{K,T}(t,S)dt + \sum_{i=0}^n \theta_i^{K,T}(t,S)dW^i_t, \tag{A.1}
\]

where \(W^i_t\) are independent Wiener processes. We study transition probabilities. \(P_{K,T}(t,S) = p(t,S,T,K)\) denotes the probability that the underlying asset will reach level \(K\) at time \(T\) given that it has value \(S\) at \(t\).

Some properties are

\[
\begin{align*}
\frac{\delta P_{K,T}(t,S)}{\delta \sigma_{K',T'}(t,S)} &= \frac{1}{2} p(t,S,T',K') K'^2 \frac{\partial^2}{\partial K'^2} p(T',K',T,K), \\
\frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} &= \frac{1}{4} p(t,S,T',K') K'^2 \frac{\partial^2}{\partial K'^2} p(T',K',T'',K'') K''^2 \frac{\partial^2}{\partial K''^2} p(T'',K'',T,K), \\
\frac{\partial P_{K,T}(t,S)}{\partial t} + (r-q)dP_{K,T}(t,S) + \frac{1}{2} \sigma_{K,T}(t,S) \frac{\partial^2 P_{K,T}(t,S)}{\partial S^2} &= 0, 
\end{align*}
\]

\(t \leq T' \leq T'' \leq T\) (Derman and Kani 1997). The dynamic is

\[
dP_{K,T}(t,S) = \left( \frac{\partial P_{K,T}(t,S)}{\partial t} + \mu_S \frac{\partial P_{K,T}(t,S)}{\partial S} + \frac{1}{2} \sigma_{K,T}(t,S) S^2 \frac{\partial^2 P_{K,T}(t,S)}{\partial S^2} \right) dt + \sigma_{K,T}(t,S) \frac{\partial P_{K,T}(t,S)}{\partial S} dW^0_t + \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S)} d\sigma_{K',T'}^2(t,S) dK' dT' + \frac{1}{2} \int_t^T \int_0^T \int_0^{\infty} \frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} d\sigma_{K',T'}^2(t,S) d\sigma_{K'',T''}(t,S) dK' dK'' dT' dT'' \tag{A.5}
\]

The integrand in the last term of (A.5) is symmetric with respect to \(T'\) and \(T''\). We use that and later the fact that we can integrate the order of integration to rewrite it as

\[
\begin{align*}
\frac{1}{2} \int_t^T \int_0^T \int_0^{\infty} \frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} d\sigma_{K',T'}^2(t,S) d\sigma_{K'',T''}(t,S) dK' dK'' dT' dT'' \\
= \frac{1}{2} \int_t^T \int_0^T g(T',T'')dT' dT'' = \int_0^T \int_0^T g(T',T'')dT' dT'' = \frac{1}{2} \cdot 2 \int_t^T \int_t^T g(T',T'')dT' dT'' \\
= \int_t^T \int_0^T \int_0^{\infty} \frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} d\sigma_{K',T'}^2(t,S) d\sigma_{K'',T''}(t,S) dK' dK'' dT' dT' \\
= \int_0^{\infty} \int_0^T \int_t^T \frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} d\sigma_{K',T'}^2(t,S) d\sigma_{K'',T''}(t,S) dT' dT' dK' dK'' \tag{A.6}
\end{align*}
\]

(A.1), (A.4) and (A.6) in (A.5) gives

\[
dP_{K,T}(t,S) = \sigma_{K,T}(t,S) \frac{\partial P_{K,T}(t,S)}{\partial S} \left( dW^0_t + \frac{\mu(t) - r + q}{\sigma_{K,T}(t,S)} dt \right) + \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S)} \sigma_{K',T'}^2(t,S) dt + \sum_{i=0}^n \theta_i^{K',T'}(t,S) dW^i_t \right) dK' dT' \\
+ \int_0^{\infty} \int_0^T \int_t^T \frac{\delta^2 P_{K,T}(t,S)}{\delta \sigma_{K',T'}^2(t,S) \delta \sigma_{K'',T''}(t,S)} \sigma_{K',T'}^2(t,S) dt + \sum_{i=0}^n \theta_i^{K',T'}(t,S) dW^i_t \right) dT'' dT' dK' dK'' \tag{A.7}
\]
It can be shown that

\[ (dt)^2 = 0, \quad (dtdW)^2 = 0, \quad (dW^i dW^j) = \begin{cases} 0, & i \neq j \quad \text{(independent)} \\ dt_i, & i = j \end{cases} \]

which gives

\[
dP_{K,T}(t, S) = \sigma_{K,T}(t, S) S \frac{\partial P_{K,T}(t, S)}{\partial S} \left( dw^0_t + \frac{\mu(t) - r + q}{\sigma_{K,T}(t, S)} dt \right) + \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t, S)}{\delta \sigma^2_{K,T}(t, S)} \sigma^2_{K,T}(t, S) \left( \alpha_{K,T}(t, S) dt + \sum_{i=0}^n \theta_{K',T}(t, S) dW^i \right) dK' dT' \\
+ \int_0^T \int_0^\infty \int_t^T \int_t^T \frac{\delta^2 P_{K,T}(t, S)}{\delta \sigma^2_{K',T}(t, S) \delta \sigma^2_{K'',T''}(t, S)} \sigma^2_{K',T}(t, S) \sigma^2_{K'',T''}(t, S) dK' dT' dK'' dT'' \\
= \sigma_{K,T}(t, S) S \frac{\partial P_{K,T}(t, S)}{\partial S} \left( dw^0_t + \frac{\mu(t) - r + q}{\sigma_{K,T}(t, S)} dt \right) + \sum_{i=0}^n \left( \int_t^T \int_0^\infty \int_t^T \frac{\delta P_{K,T}(t, S)}{\delta \sigma^2_{K,T}(t, S)} \sigma^2_{K,T}(t, S) \theta_{K',T}(t, S) dK' dT' \right) dW^i_t \\
+ \int_t^T \int_0^\infty \int_t^T \int_t^T \frac{\delta P_{K,T}(t, S)}{\delta \sigma^2_{K',T}(t, S)} \alpha_{K',T}(t, S) \sigma^2_{K',T}(t, S) dK' dT' dt \\
+ \int_0^\infty \int_t^\infty \int_t^T \int_t^T \frac{\delta^2 P_{K,T}(t, S)}{\delta \sigma^2_{K',T}(t, S) \delta \sigma^2_{K'',T''}(t, S)} \sigma^2_{K',T}(t, S) \sigma^2_{K'',T''}(t, S) \sum_{i=0}^n \theta_{K',T}(t, S) \theta_{K'',T''}(t, S) dT'' dT' dK' dK'' dt.
\]

Changing the order of integration again, the last term in (A.11) can be written, using (A.3) and that \( \frac{\delta^2 P_{K,T}}{\delta \sigma^2_{K',T} \delta \sigma^2_{K'',T''}} \) as

\[
\int_t^T \int_0^\infty \int_t^T \int_0^\infty \frac{1}{4} p(t, S, T'', K'') K''_{T''} \frac{\partial^2}{\partial K''_2} p(T'', K'', T', K') K'^2 \frac{\partial^2}{\partial K'^2} p(T', K', T, K) \\
\times \sigma^2_{K',T}(t, S) \sigma^2_{K'',T''}(t, S) \sum_{i=0}^n \theta_{K',T}(t, S) \theta_{K'',T''}(t, S) dK'' dT'' dK' dT' dt
\]

\[
= \frac{1}{2} \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t, S)}{\delta \sigma^2_{K',T}(t, S)} \sigma^2_{K',T}(t, S) \sum_{i=0}^n \theta_{K',T}(t, S) \frac{1}{p(t, S, T', K')} \times \\
\int_t^T \int_0^\infty \sigma^2_{K'',T''}(t, S) \theta_{K'',T''}(t, S) p(t, S, T'', K'') K''_{T''} \frac{\partial^2}{\partial K''^2} p(T'', K'', T', K') dK'' dT'' dK' dT' dt.
\]
Therefore the drift in (A.13) has to be zero for the market to be arbitrage free. We then see that the drift martingale for the market to be arbitrage free (Derman and Kani 1997). That is equivalent to the underlying

\[ \Pi^i \]

\[ (A.12) \text{ in } (A.11) \text{ gives} \]

\[ dP_{K,T}(t,S) = \sigma_{K,T}(t,S)S \frac{\partial P_{K,T}(t,S)}{\partial S} \left( dW_t^0 + \frac{\mu(t) - r + q_t}{\sigma_{K,T}(t,S)} dt \right) \]

\[ + \sum_{i=0}^n \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma^2_{K',T'}(t,S)} \sigma^2_{K',T'}(t,S) \theta^i_{K',T'}(t,S) dK'dT'dW_t^i \]

\[ + \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma^2_{K',T'}(t,S)} \sigma^2_{K',T'}(t,S) \left( \alpha_{K',T'}(t,S) + \frac{1}{2} \sum_{i=0}^n \theta^i_{K',T'}(t,S) \frac{1}{p(t,S,T',K')} \right) \times \int_t^T \int_0^\infty \sigma^2_{K',T'}(t,S) \theta^i_{K',T'}(t,S)p(t,S,T'',K'') dK''dT'' dK'dT' \right) dt. \]  

(A.13)

With an equivalent martingale measure,

\[ dW_t^0 = dW_t^0 + \frac{\mu(t) - r + q_t}{\sigma_{K,T}(t,S)} dt, \]  

(A.14)

\[ dW_t^i = dW_t^i + \Pi^i dt, \quad i = 1, \ldots, n, \]  

(A.15)

where \( \Pi^i \) is the market price of risk associated with the risk factor \( W_t^i \), the underlying asset has to be a martingale for the market to be arbitrage free (Derman and Kani 1997). That is equivalent to the underlying asset having a drift equal to zero, which is equivalent to the transition probabilities having a drift equal to zero. Therefore the drift in (A.13) has to be zero for the market to be arbitrage free. We then see that the drift parameters \( \alpha_{K,T}(t,S) \) in (A.1) must satisfy the no-arbitrage conditions

\[ \alpha_{K,T}(t,S) = - \sum_{i=0}^n \theta^i_{K,T}(t,S) \left( \frac{1}{2} \frac{1}{p(t,S,T,K)} \right) \]

\[ \times \int_t^T \int_0^\infty \sigma^2_{K',T'}(t,S) \theta^i_{K',T'}(t,S)p(t,S,T',K') dK''dT'' \right) dt. \]  

(A.16)

The resulting dynamic under the equivalent martingale measure is

\[ dP_{K,T}(t,S) = \sigma_{K,T}(t,S)S \frac{\partial P_{K,T}(t,S)}{\partial S} dW_t^0 + \sum_{i=0}^n \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma^2_{K',T'}(t,S)} \sigma^2_{K',T'}(t,S) \theta^i_{K',T'}(t,S) dK'dT'dW_t^i \]  

(A.17)

and the process does not imply arbitrage.

Note that if the dynamic in the original proof by Derman and Kani (ibid.),

\[ d\sigma_{K,T}^2(t,S) = \alpha_{K,T}(t,S)dt + \sum_{i=0}^n \theta^i_{K,T}(t,S)dW_t^i, \]  

(A.18)

is used, we get a similar result for the drift condition,

\[ \alpha_{K,T}(t,S) = - \sum_{i=0}^n \theta^i_{K,T}(t,S) \left( \frac{1}{2} \frac{1}{p(t,S,T,K)} \right) \]

\[ \times \int_t^T \int_0^\infty \theta^i_{K',T'}(t,S)p(t,S,T',K') dK''dT'' \right) dt. \]  

(A.19)

The dynamic under the EMM is

\[ dP_{K,T}(t,S) = \sigma_{K,T}(t,S)S \frac{\partial P_{K,T}(t,S)}{\partial S} dW_t^0 + \sum_{i=0}^n \int_t^T \int_0^\infty \frac{\delta P_{K,T}(t,S)}{\delta \sigma^2_{K',T'}(t,S)} \theta^i_{K',T'}(t,S) dK'dT'dW_t^i \]  

(A.20)