Linköping University
Institute of Technology
Department of Management and Engineering
Division of Production Economics

Thesis for a M. Sc. in Applied Financial Mathematics
(prog. Applied Physics and Electrical Engineering)

Optimizing Smooth Local Volatility Surfaces with Power Utility Functions

ISRN LIU-IEI-TEK-A–15/02298–SE

Authors:
Gustav Sällberg
Pontus Söderbäck

External Supervisor:
Szabolcs Gaal
Internal Supervisor:
Jörgen Blomvall
Examiner:
Ou Tang

July 1, 2015

Linköping University
INSTITUTE OF TECHNOLOGY

Abstract

The master thesis is focused on how a local volatility surfaces can be extracted by optimization with respect to smoothness and price error. The pricing is based on utility based pricing, and developed to be set in a risk neutral pricing setting. The pricing is done in a discrete multinomial recombining tree, where the time and price increments optionally can be equidistant. An interpolation algorithm is used if the option that shall be priced is not matched in the tree discretization. Power utility functions are utilized, where the log-utility preference is especially studied, which coincides with the (Kelly) portfolio that systematically outperforms any other portfolio. A fine resolution of the discretization is generally a property that is sought after, thus a series of derivations for the implementation are done to restrict the computational encumbrance and thus allow finer discretization.

The thesis is mainly focused on the derivation of the method rather than finding optimal parameters that generate the local volatility surfaces. The method has shown that smooth surfaces can be extracted, which consider market prices. However, due to lacking available interest and dividend data, the pricing error increases symmetrically for longer option maturities. However, the method shows exponential convergence and robustness to different initial (flat) volatilities for the optimization initiation.

Given an optimal smooth local volatility surface, an arbitrary payoff function can then be used to price the corresponding option, which could be path-dependent, such as barrier options. However, only vanilla options will be considered in this thesis. Finally, we find that the developed method is valid for extracting local volatility surfaces, given adequate data access and some refinements.

Keywords: local volatility surface, LVS, optimization, roughness, smooth, risk neutral pricing, optimal growth, pricing error, automatic differentiation, algorithmic differentiation

Acknowledgements

We would like to thank both our internal supervisor Jörgen Blomvall and external supervisor Szabolcs Gaal at Nordea, for many insights and numerous suggestions. We would also like to thank our examiner Ou Tang. Finally, we would like to thank our fellow students in Börssalen for your humor and companionship.

Nomenclature

Conventions

a	Scalar, lowercase not bold
$\mathbf{a}$	Column vector, lowercase bold.
A	Matrix, uppercase not bold
$\mathbb{R}$	Real line
$\mathbb{R}^n$	Euclidean space in n dimensions
$\mathbb{N}$	Natural number, include the zero
$\mathbb{N}^+$	Natural number, not include the zero

Symbols

σ_{imp}	Implied volatility
σ	Local volatility
σ^2	Local variance
μ	Expected return
S	Stock price
K	Strike price
T	Time
δ	Continuous dividend yield
r_f	Continuous risk-free rate
r	Growth rate
ψ	Price of derivative
ξ	Payoff function for vanilla options
G	Grid
$\tilde{G}$	Extended grid
Π	Local volatility surface
$\boldsymbol{\pi}$	Vectorized local volatility surface
C	Covariance matrix
$\mathcal{I}_G$	Set of integers, representing the strike levels in grid G
$\mathcal{J}_G$	Set of integers, representing the discrete time in grid G
$\mathcal{I}^{i,t}$	Set of integers corresponding to the outgoing branches of node (S_i, T_t)
m_c	The c-th central moment
n_c	Number of central moments matched
Ω	Sample space
p	Physical probabilities
q	Risk-neutral probabilities

Functions

$\phi(\cdot)$	Gaussian probability density function
$\Phi(\cdot)$	Gaussian cumulative distribution function
$\Phi^{-1}(\cdot)$	Gaussian inverse cumulative distribution function
!!	Double factorial
$\mathrm{E}[\,\cdot\,]$	Expected value of $\cdot$
$\mathrm{Var}[\,\cdot\,]$	Variance of $\cdot$
$A \subset B$	A is a subset (or possibly equal) to B
$\circ$	Hadamard product
$vec(\cdot)$	Vectorization of a matrix
$\lvert \cdot \rvert$	Cardinality of a matrix $\cdot$
$\lvert \cdot \rvert$	Absolute value of the scalar $\cdot$
$\Vert \cdot \Vert \equiv \Vert \cdot \Vert_2$	Euclidean norm
A^T	Transpose of the matrix A
$\times$	Cartesian product
$\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}$	Floor

Contents

List of Figures

List of Tables

1 Introduction

The turbulent financial markets require increasingly accurate and consistent risk measures. Thus, more sophisticated models are required to accurately model the financial markets, especially during economical downturns. For some time, a way to measure an instruments risk is by implied volatility, but it is as most measures flawed. The implied volatility is in some sense an average of the expected volatility (Derman et al., 1995) of the instrument until expiration and does thus not properly reflect the time component in the volatility. However, *local volatility* can be compared with forward rate contributions to the spot rate and should thus contain more local information. Unfortunately, realistic local volatility surfaces are not a trivial matter to extract.

General approaches to extract local volatility surfaces can be based on finite difference methods, which mostly yield a quite rough surface. Additionally, adapting these methods to cope with exotic options is often quite tricky. Another quite common approach is to use a given implied volatility surface and then model the local volatility surface with a parametric model, e.g. SABR, (Derman et al., 1995). Furthermore, by going through previous work and literature, it is found uncommon to use optimization to extract local volatility surfaces, with respect to smoothness.

Given a local volatility surface it is possible to price many types of option payoffs and particularly path dependent options. To our knowledge there are not any published multinomial utility based pricing models and furthermore we have found it uncommon to use optimization to extract the surface.

1.1 Purpose

The purpose is to develop and evaluate a utility based pricing methodology of extracting a local volatility surface, which minimizes a combination of roughness and pricing error.

1.1.1 Problem Formulation

Determine a smooth and market consistent local volatility surface of an underlying asset.

1.2 Delimitations

The methodology is intended to be quite general and should thus in the future be able to be generalized to cope with most asset classes. However, since the focus is to investigate the methodology and not the use of many types of derivatives, vanilla options will be used. Furthermore, preferably, an existing model with corresponding data series and results will be used to compare our results, since time will most likely be scarce. Additionally, the model will assume that the markets are efficient and that there is generally no arbitrage.

1.3 Document Structure

The mathematical scope will be defined in the local volatility environment. In this local volatility setting, the master optimization problem will be defined followed by the description of the problems subordinate problems and its derivations. These consist of deriving an optimal utility preference of optimal wealth growth, in the discretized multinomial recombining tree representation. This optimization problem is quite computationally strenuous and thus automatic differentiation is implemented when the function and its first order derivative simultaneously are required. When the smooth local volatility surface is extracted it will be evaluated and the results will be documented. Lastly, our findings will be discussed and summarized.

2 Volatility

In financial markets it is essential to measure and manage risk. Variance is an intuitive way of measuring risk, which is directly related to the volatility or in other words standard deviation of for instance a stock return. However, there are several other less trivial variations of volatility, which have pleasant properties and applications. The most well known and applied versions of these volatilities is the *implied volatility*, which is derived as the volatility that makes the theoretical Black-Scholes-Merton (BSM) price conformed to the quoted market price. Furthermore, the *local volatility* is essential and describes the instantaneous risk.

2.1 Implied Volatility

The Black-Scholes-Merton, BSM, formula was published in 1973 and was quickly utilized. The formula was unmodified until the 1987 crash, where the normal distribution assumptions inadequately considered black swan events or in other words big economical downturns. Thus, the distribution of probabilities needed to be adjusted to cope with the long tail risk associated with the market movements. The formula's use was refined, (Latané and Rendleman, 1976, pp. 369-371), to extract the implied standard deviation from the BSM formula, which later was relabeled as the *implied volatility*. This implied volatility is determined as the solution where the BSM-formula coincides with the market quoted price.

2.1.1 Extraction of Implied Volatility

The implied volatility must be extracted numerically since there does not exist an analytical inverse. There are several ways of estimating the implied volatility, σ_{imp} at time t. The implied volatility for a call option with the Black-Scholes price, C, can be found by the Newton-Rahpson method,

$$f(\sigma_{imp}) = Se^{-\delta^{t,T}(T-t)}\Phi_{0,1}(d_1) - Ke^{-r_f^{t,T}(T-t)}\Phi_{0,1}(d_2) - C, \tag{1}$$

with

$$\begin{aligned} d_1 &= \frac{\ln(\frac{S}{K}) + (r_f^{t,T} - \delta^{t,T} + \frac{1}{2}\sigma_{imp}^2)(T-t)}{\sigma_{imp}\sqrt{T-t}}, \\ d_2 &= d_1 - \sigma_{imp}\sqrt{T-t}, \end{aligned} \tag{2}$$

where the option has the strike price K and maturity at time, T. The underlying price is S and the continuous dividend yield $\delta^{t,T}$ and the continuous risk free rate $r_f^{t,T}$. The standard normal cumulative distribution function is represented by $\Phi_{0,1}$.

The procedure is then to determine a starting solution, σ_{imp}^0 and then the function

$$\sigma_{imp}^{i+1} = \sigma_{imp}^{i} - \frac{f(\sigma_{imp}^i)}{\frac{\partial f(\sigma_{imp}^i)}{\partial \sigma}} \tag{3}$$

iterates until the stop criterion

$$|\sigma_{imp}^{i+1} - \sigma_{imp}^{i}| < \epsilon, \tag{4}$$

is satisfied, where ϵ is a small number. The derivative, also known as *vega*, is given by

$$\frac{\partial f}{\partial \sigma_{imp}} = Se^{-\delta^{t,T}(T-t)}\Phi(d_1)\sqrt{T-t}. \tag{5}$$

2.1.2 Smile and Skew

Several problems remain in the assumptions of the BSM framework. The model assumes a constant volatility, (Black and Scholes, 1973), which was pointed out as unrealistic (Latané and Rendleman, 1976, pp. 370-371) since different implied volatilities need to be assigned depending on the options strike and maturity. This concept is still essential in the present market. Furthermore, the *volatility skew* and *volatility smile*, could not be observed in the market prior to 1987, (Hull, 2011, pp.415-416), and thus the implied volatility was in general constant for all strikes.

2.2 Local Volatility

The local volatility is not derived from the BSM formula and does thus not share the same draw downs. Furthermore, the local volatility is instantaneous and is therefore only valid in a infinitesimally local area in contrast to the implied volatility, which is determined as an average throughout the time to maturity (Derman et al., 1995). There are parallels that can be drawn to interest rate curves, where the spot rate curve can be interpreted as an average of an corresponding forward rate curve.

The local volatility is a *deterministic function*, $\sigma(K,t)$, dependent on a strike price and a reference time, usually $t = t_0$. Derman and Kani (1997, p. 10) call this an *effective theory*, where the assets process can be written as

$$\frac{dS}{S} = \mu(t)dt + \sigma(S,t)dZ, \tag{6}$$

where S is the asset price, μ is the drift of the process and Z is the Wiener process.

Derman and Kani were among the first to work with local volatility. They used a time discrete binomial tree approach to derive local volatilities that matched the volatility smile, (Derman and Kani, 1994b), and later trinomial trees were used, (Derman and Kani, 1997).

Dupire (1994) was also among the pioneers, but with a continuous time approach. From Dupire's method the expression

$$\begin{aligned}\frac{\partial C(K,T)}{\partial T} &= \frac{1}{2}\sigma(K,T)^2 K^2 \frac{\partial^2 C(K,T)}{\partial K^2} - \delta^{t,T} C(K,T) - (r_f^{t,T} - \delta^{t,T}) K \frac{\partial C(T,K)}{\partial K} \Leftrightarrow \\ \sigma(K,T)^2 &= 2\frac{\frac{\partial C(K,T)}{\partial T} + (r_f^{t,T} - \delta^{t,T}) K \frac{\partial C(K,T)}{\partial K} + \delta^{t,T} C(K,T)}{K^2 \frac{\partial^2 C(K,T)}{\partial K^2}},\end{aligned} \tag{7}$$

was derived by Derman and Kani (1994a). This formula is quite often central in other methods for extracting local volatilities from implied volatility or price surfaces.

2.2.1 Change of Variable

The deterministic local volatility function, $\sigma(K,t)$, was presented as a function of strike price and time. It can be shown, (Derman and Kani, 1997), that it is possible to derive a deterministic volatility function as a function of the underlying price, S and time t.

$$\sigma(S,t) = \sigma(K,t)|_{S=K} \tag{8}$$

For a given current time, t_0 the deterministic local volatility expression remains fixed for times $t > t_0$ for future time and asset prices. As the price of an underlying is unique at any point in time the *instantaneous volatility*,

$$\sigma(t) = \sigma(S_t, t), \tag{9}$$

can be defined with the corresponding price process

$$\frac{dS_t}{S_t} = \mu_t dt + \sigma(t) dZ_t, \tag{10}$$

where dZ_t is the standard Wiener process, (Derman and Kani, 1997).

2.2.2 Average of Stochastic Volatility

The procedure above can be seen as an average of the sources of stochastic volatility, where all contributions are offset, except the index price. Thus, S_t, is the only remaining contribution at the deterministic time t, (Derman and Kani, 1997). Therefore, the price from stochastic and local volatility models should coincide.

2.2.3 Local Volatility as a Surface

In theory, a function $\sigma(t,K)$ is impractical to utilize. Therefore we choose to define a *local volatility surface.*

Definition 1. *A **deterministic local volatility function** in its domain,*

$$(S,T) \in G = \{(k,t)|k = K_1, \ldots, K_{n_K}, \, T = T_0, \ldots, T_{n_T}, \} \tag{11}$$

can be written as a local volatility surface,

$$U_{[0,n_K],[0,n_T]} = \begin{bmatrix} \sigma(S_0, T_0) & \dots & \sigma(S_0, T_{n_T}) \\ \vdots & \ddots & \vdots \\ \sigma(S_{n_K}, T_0) & \dots & \sigma(S_{n_K}, T_{n_T}) \end{bmatrix} = \begin{bmatrix} \sigma_{0,0} & \dots & \sigma_{0,n_T} \\ \vdots & \ddots & \vdots \\ \sigma_{n_K,0} & \dots & \sigma_{n_K,n_T} \end{bmatrix}. \tag{12}$$

For a specific point and thus evaluation of the deterministic volatility function can be denoted as

$$U_{k,t} \equiv \sigma(K_k, T_t). \tag{13}$$

3 Optimization Problem

The master optimization problem, for the local volatility surface g, needs individual subordinate parts such as roughness and pricing error, which are derived in this chapter. The surface roughness is measured by the function h and the pricing will be computed by $\boldsymbol{g}$, while $\boldsymbol{b}_e$ contains the market quoted prices. The optimization problem can be formulated as

$$\begin{aligned} \min_{U,\mathbf{z}} \quad & L(U) = h(U) + \frac{1}{2}\mathbf{z}^{\mathbf{T}} E\mathbf{z} \\ \text{s.t.} \quad & \boldsymbol{g}(U) + F\mathbf{z} = \mathbf{b}_e \\ & U \geq U_l, \end{aligned} \tag{14}$$

where E and F are a diagonal matrices. The purpose of E and F is to relate the penalties for the pricing error and smoothness. The second constraint sets a lower limit, U_l, for the volatilities, which should for instance be at least greater or equal to zero. The prices of the, n individual options are determined by the function $\boldsymbol{g} : \mathbb{R}^{n_t \times n_K} \to \mathbb{R}^n$, where n_t and n_K is the number of points in time and strike prices respectively. The vector $\mathbf{b}_e \in \mathbb{R}^n$ is the market quoted prices and furthermore the pricing error is

$$\mathbf{z} = F^{-1}(\mathbf{b_e} - \boldsymbol{g}(U)), \tag{15}$$

which inserted in (14) gives

$$\begin{aligned} \min_{U} \quad & L(U) = h(U) + \frac{1}{2}(\mathbf{F^{-1}}(\mathbf{b_e} - \boldsymbol{g}(\mathbf{U})))^{\mathbf{T}} E(\mathbf{F^{-1}}(\mathbf{b_e} - \boldsymbol{g}(\mathbf{U}))) \\ \text{s.t.} \quad & U \geq U_l. \end{aligned} \tag{16}$$

3.1 Roughness Measure

The function h measures the roughness of the surface with respect to the first and second order of derivative, where higher roughness corresponds to a greater value. The complete expression of the roughness measure, as a function of $\boldsymbol{u} \equiv vec(U)$[1], can be written as

$$h(U) = \frac{1}{2}\boldsymbol{u}^T H_h \boldsymbol{u}, \tag{17}$$

where H_h contains the penalties for the roughness. This form is compact but it is hard to interpret how different penalties are represented. This chapter and its sections will be dedicated to derive this form.

Firstly, the first order derivative can be approximation as

$$\frac{\partial f}{\partial x}(a,b) = \lim_{\Delta_x \to 0} \frac{f(a+\Delta_x, b) - f(a,b)}{\Delta_x} \approx \frac{f(a+\Delta_x, b) - f(a,b)}{\Delta_x}. \tag{18}$$

The second order has both ordinary and mixed partial derivatives.

$$\frac{\partial^2 f}{\partial x}(a,b) = \lim_{\Delta_x \to 0} \frac{f(a+\Delta_x, b) - 2f(a,b) + f(a-\Delta_x, b)}{\Delta_x^2} \approx \frac{f(a+\Delta_x, b) - 2f(a,b) + f(a-\Delta_x, b)}{\Delta_x^2}, \tag{19a}$$

$$\frac{\partial^2 f}{\partial x \partial y}(a,b) \approx \frac{f\left(a+\Delta_x, b+\Delta_y\right) - f(a+\Delta_x, b-\Delta_y) - f\left(a-\Delta_x, b+\Delta_y\right) + f\left(a-\Delta_x, b-\Delta_y\right)}{4\Delta_x \Delta_y}. \tag{19b}$$

The surface U is discretized in order to use the derivative expression. The domain of the surface consists of $(n_T + 1)$ points in time and $(n_K + 1)$ points in strike level, and will referred to as *grid*. The difference between the points T_t and T_{t+1} in time is denoted ΔT_t and analogously ΔK_k is the difference between K_k and K_{k+1}.

[1] The matrix U is transformed to a vector by columnwise vectorization, see appendix B.

From these general expressions it is possible to formulate the function h as the combinations of the two first orders of derivatives with respect to the time and strike spectrum,

$$\begin{aligned}
h(U) = \frac{1}{2}\sum_{t=0}^{n_T-1}\sum_{k=0}^{n_K} a^T_{k,t}\left(\frac{U_{k,t+1}-U_{k,t}}{\Delta_{T_t}}\right)^2 \Delta_{T_t} &+ \frac{1}{2}\sum_{t=0}^{n_T}\sum_{k=0}^{n_K-1} a^K_{k,t}\left(\frac{U_{k+1,t}-U_{k,t}}{\Delta_{K_k}}\right)^2 \Delta_{K_k} \\
+\frac{1}{2}\sum_{t=1}^{n_T-1}\sum_{k=0}^{n_K} a^{TT}_{k,t}&\left(\frac{\frac{U_{k,t+1}-U_{k,t}}{\Delta_{T_t}}-\frac{U_{k,t}-U_{k,t-1}}{\Delta_{T_{t-1}}}}{\frac{\Delta_{T_t}+\Delta_{T_{t-1}}}{2}}\right)^2 \frac{\Delta_{T_t}+\Delta_{T_{t-1}}}{2} \\
+\frac{1}{2}\sum_{t=0}^{n_T}\sum_{k=1}^{n_K-1} a^{KK}_{k,t}&\left(\frac{\frac{U_{k+1,t}-U_{k,t}}{\Delta_{K_k}}-\frac{U_{k,t}-U_{k-1,t}}{\Delta_{K_{k-1}}}}{\frac{\Delta_{K_k}+\Delta_{K_{k-1}}}{2}}\right)^2 \frac{\Delta_{K_k}+\Delta_{K_{k-1}}}{2} \\
+\frac{1}{2}\sum_{t=1}^{n_T-1}\sum_{k=1}^{n_K-1} a^{KT}_{k,t}\left(\frac{U_{k+1,t+1}-U_{k-1,t+1}-U_{k+1,t-1}+U_{k-1,t-1}}{(\Delta_{T_t}+\Delta_{T_{t-1}})(\Delta_{K_k}+\Delta_{K_{k-1}})}\right)^2 &\frac{(\Delta_{T_t}+\Delta_{T_{t-1}})}{2}\cdot\frac{(\Delta_{K_k}+\Delta_{K_{k-1}})}{2}.
\end{aligned} \tag{20}$$

Note that an integration factor is introduced in (20) to cope with non-equidistant discretization and where a^T, a^K, a^{TT}, a^{KK} and a^{KT} are the penalties for the different derivatives. The first two double summation is the first derivative with respect to time and strike level respectively. The third and fourth is the "simple" second derivative with respect to time and strike level respectively and the fifth double summation is the mixed second derivative. A more compact notation is,

$$\begin{aligned}
h(U) = \frac{1}{2}\sum_{t=0}^{n_T-1}\sum_{k=0}^{n_K} \hat{a}^T_{k,t}\left(U_{k,t+1}-U_{k,t}\right)^2 &+ \frac{1}{2}\sum_{t=0}^{n_T}\sum_{k=0}^{n_K-1} \hat{a}^K_{k,t}\left(U_{k+1,t}-U_{k,t}\right)^2 \\
&+\frac{1}{2}\sum_{t=1}^{n_T-1}\sum_{k=0}^{n_K} \hat{a}^{TT}_{k,t}\Big(\Delta_{t-1}(U_{k,t+1}-U_{k,t})-\Delta_t(U_{k,t}-U_{k,t-1})\Big)^2 \\
&+\frac{1}{2}\sum_{t=0}^{n_T}\sum_{k=1}^{n_K-1} \hat{a}^{KK}_{k,t}\Big(\Delta_{k-1}(U_{k+1,t}-U_{k,t})-\Delta_k(U_{k,t}-U_{k-1,t})\Big)^2 \\
&+\frac{1}{2}\sum_{t=1}^{n_T-1}\sum_{k=1}^{n_K-1} \hat{a}^{KT}_{k,t}\Big(U_{k+1,t+1}-U_{k-1,t+1}-U_{k+1,t-1}+U_{k-1,t-1}\Big)^2,
\end{aligned} \tag{21}$$

where

$$\begin{aligned}
\hat{a}^T_{k,t} &= \frac{a^T_{k,t}}{\Delta_{T_t}},\ \forall k=0,\dots,n_K, \forall t=0,\dots,n_T-1 \\
\hat{a}^K_{k,t} &= \frac{a^K_{k,t}}{\Delta_{K_k}}\ \forall k=0,\dots,n_K-1,\ \forall t=0,\dots,n_T \\
\hat{a}^{TT}_{k,t} &= \frac{2a^{TT}_{k,t}}{\Delta^2_{T_t}\Delta^2_{T_{t-1}}(\Delta_{T_t}+\Delta_{T_{t-1}})}\ \forall k=0,\dots,n_K,\ \forall t=1,\dots,n_T-1 \\
\hat{a}^{KK}_{k,t} &= \frac{2a^{KK}_{k,t}}{\Delta^2_{K_k}\Delta^2_{K_{k-1}}(\Delta_{K_k}+\Delta_{K_{k-1}})}\ \forall k=1,\dots,n_K-1,\ \forall t=0,\dots,n_T \\
\hat{a}^{KT}_{k,t} &= \frac{a^{KT}_{k,t}}{4(\Delta_{T_t}+\Delta_{T_{t-1}})(\Delta_{K_k}+\Delta_{K_{k-1}})}\forall k=1,\dots,n_K-1,\ \forall t=1,\dots,n_T-1.
\end{aligned} \tag{22}$$

The derivation of the first term will be presented, while the other four are derived in appendix A. For convenience, the matrix U is vectorized, $vec(U)=\boldsymbol{u}$ and analogously its transpose, $vec(U^T)=\boldsymbol{u}_T$. Furthermore, there exists a linear transformation P such that $\boldsymbol{u}_T=P\boldsymbol{u}$. The elements in this linear mapping can be determined as

$$P_{i,j} = \begin{cases} 1 & \text{if } i,j = 1 + \left\lfloor \frac{i-1}{n_T+1} \right\rfloor + (i-1)(n_K+1) - (n_K+1)(n_T+1) \left\lfloor \frac{i-1}{n_T+1} \right\rfloor \\ 0 & \text{otherwise} \end{cases}, \ \forall i = 1, \dots, (n_K+1) \cdot (n_T+1). \tag{23}$$

These five double summations can be expressed on matrix form,

$$h_1 = \frac{1}{2} \boldsymbol{u}^T P^T A_T^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{T}}) A_T P \boldsymbol{u} \qquad = \frac{1}{2} \sum_{t=0}^{n_T-1} \sum_{k=0}^{n_K} \hat{a}_{k,t}^T \left(U_{k,t+1} - U_{k,t}\right)^2 \tag{24}$$

$$h_2 = \frac{1}{2} \boldsymbol{u}^T A_K^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{K}}) A_K \boldsymbol{u} \qquad = \frac{1}{2} \sum_{t=0}^{n_T} \sum_{k=0}^{n_K-1} \hat{a}_{k,t}^K \left(U_{k+1,t} - U_{k,t}\right)^2 \tag{25}$$

$$h_3 = \frac{1}{2} \boldsymbol{u}^T P^T A_{TT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{TT}}) A_{TT} P \boldsymbol{u} \qquad = \frac{1}{2} \sum_{t=1}^{n_T-1} \sum_{k=0}^{n_K} \hat{a}_{k,t}^{TT} \Big(\Delta_{t-1}(U_{k,t+1} - U_{k,t}) - \Delta_t (U_{k,t} - U_{k,t-1}) \Big)^2 \tag{26}$$

$$h_4 = \frac{1}{2} \boldsymbol{u}^T A_{KK}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KK}}) A_{KK} \boldsymbol{u} \qquad = \frac{1}{2} \sum_{t=0}^{n_T} \sum_{k=1}^{n_K-1} \hat{a}_{k,t}^{KK} \Big(\Delta_{k-1}(U_{k+1,t} - U_{k,t}) - \Delta_k (U_{k,t} - U_{k-1,t}) \Big)^2 \tag{27}$$

$$h_5 = \frac{1}{2} \boldsymbol{u}^T A_{KT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KT}}) A_{KT} \boldsymbol{u}. \qquad = \frac{1}{2} \sum_{t=1}^{n_T-1} \sum_{k=1}^{n_K-1} \hat{a}_{k,t}^{KT} \Big(U_{k+1,t+1} - U_{k-1,t+1} - U_{k+1,t-1} + U_{k-1,t-1} \Big)^2, \tag{28}$$

where h_1 is derived below and the other four are derived in appendix A. First the block,

$$B_T = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}, \tag{29}$$

where the dimension of the matrix is $(n_T) \times (n_T + 1)$ is constructed. A block corresponding to all $k = 0, \dots, n_K$ are then used to build the matrix,

$$A_T = \begin{bmatrix} B_T & & \\ & \ddots & \\ & & B_T \end{bmatrix}, \tag{30}$$

which has the dimension, $[(n_K + 1)n_T \times (n_K + 1)(n_T + 1)]$. The weights are linearized to the vector

$$\hat{\boldsymbol{a}}^{\boldsymbol{T}} = \left(\hat{a}_{0,0}^T, \quad \dots \quad \hat{a}_{0,n_T-1}^T, \quad \hat{a}_{1,0}^T, \quad \dots \quad \hat{a}_{n_K,n_T-1}^T \right)^T. \tag{31}$$

3.1.1 Complete Expression

The complete expression for h can now be written as

$$\begin{aligned} h &= h_1 + h_2 + h_3 + h_4 + h_5 \\ &= \frac{1}{2} \boldsymbol{u}^T \Big(P^T A_T^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{T}}) A_T P + A_K^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{K}}) A_K + P^T A_{TT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{TT}}) A_{TT} P \\ &\quad + A_{KK}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KK}}) A_{KK} + A_{KT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KT}}) A_{KT} \Big) \boldsymbol{u} \end{aligned} \tag{32}$$

with

$$\begin{aligned} H_h \equiv\, & P^T A_T^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{T}}) A_T P + A_K^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{K}}) A_K + P^T A_{TT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{TT}}) A_{TT} P \\ & + A_{KK}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KK}}) A_{KK} + A_{KT}^T \operatorname{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KT}}) A_{KT}. \end{aligned} \tag{33}$$

The complete expression can be written as

$$h(u) = \frac{1}{2} \boldsymbol{u}^T H_h \boldsymbol{u},$$

which was stated in equation (17).

3.2 Optimization Preparation

All surfaces can be expressed as the current surface, $\bar{\boldsymbol{u}}$ and a surface change, $\Delta\boldsymbol{u}$, as

$$\boldsymbol{u} = \bar{\boldsymbol{u}} + \Delta\boldsymbol{u}. \tag{34}$$

With this notation, equation (17) can be rewritten to,

$$\frac{1}{2}(\bar{\boldsymbol{u}} + \Delta\boldsymbol{u})^T H_h(\bar{\boldsymbol{u}} + \Delta\boldsymbol{u}) = \frac{1}{2}\left(\bar{\boldsymbol{u}}^T H_h\bar{\boldsymbol{u}} + \bar{\boldsymbol{u}}^T H_h\Delta\boldsymbol{u} + \Delta\boldsymbol{u}^T H_h\bar{\boldsymbol{u}} + \Delta\boldsymbol{u}^T H_h\Delta\boldsymbol{u}\right), \tag{35}$$

where we can observe that $\Delta\boldsymbol{u}^T\Gamma\bar{\boldsymbol{u}}$ is a scalar and therefore

$$\Delta\boldsymbol{u}^T H_h\bar{\boldsymbol{u}} = \left(\Delta\boldsymbol{u}^T H_h\bar{\boldsymbol{u}}\right)^T = \bar{\boldsymbol{u}}^T H_h^T\Delta\boldsymbol{u} = \bar{\boldsymbol{u}}^T H_h\Delta\bar{\boldsymbol{u}}, \tag{36}$$

where the third equality holds because H_h is symmetric. Hence, it is possible to write equation (17) as

$$h(u) = a_h + \mathbf{b}_h^T\Delta\boldsymbol{u} + \frac{1}{2}\Delta\boldsymbol{u}^T H_h\Delta\boldsymbol{u}, \tag{37}$$

where

$$\begin{aligned} a_h &\equiv \frac{1}{2}\bar{\boldsymbol{u}}^T H_h\bar{\boldsymbol{u}}, \\ \mathbf{b}_h^T &\equiv \bar{\boldsymbol{u}}^T H_h, \\ H_h &\equiv H_h. \end{aligned}$$

3.3 Pricing Error Measure Linearization

The function $\boldsymbol{g}$ is generally a non-linear function that we address by linearization. The Hessian of the linear approximation of the objective function is much easier (possible) to extract. Linearization of $\boldsymbol{g}$ from the vectorized surface $\boldsymbol{u}$ is given by

$$\boldsymbol{g}(u) \approx \bar{\boldsymbol{g}}(\boldsymbol{u}) \equiv \boldsymbol{g}(\bar{\boldsymbol{u}}) + \nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}})(\boldsymbol{u} - \bar{\boldsymbol{u}}) \overset{(34)}{=} \boldsymbol{g}(\bar{\boldsymbol{u}}) + \nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}})\Delta\boldsymbol{u}, \tag{38}$$

which makes it possible to rewrite (15) to

$$\mathbf{z} = F^{-1}\Big(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}}) - \nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}})\Delta\boldsymbol{u}\Big). \tag{39}$$

We assume that the expression for $\boldsymbol{g}$ and $\nabla\boldsymbol{g}$ are known, which are described in chapter 7 and 8 respectively. The second term in the objective function in (14) can be rewritten as

$$\frac{1}{2}\mathbf{z}^T E\mathbf{z} = \frac{1}{2}\left(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}}) - \nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}})\Delta\boldsymbol{u}\right)^T F^{-T}EF^{-1}\left(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}}) - \nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}})\Delta\boldsymbol{u}\right). \tag{40}$$

Since F is diagonal, $F^{-T} = F^{-1}$, and we can define

$$\begin{aligned} a_g &\equiv \frac{1}{2}\left(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}})\right)^T F^{-1}EF^{-1}\left(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}})\right), \\ \mathbf{b}_g^T &\equiv -\left(\mathbf{b}_e - \boldsymbol{g}(\bar{\boldsymbol{u}})\right)^T F^{-1}EF^{-1}\nabla_u\boldsymbol{g}(\bar{\boldsymbol{u}}), \\ H_g &\equiv \nabla_u(\boldsymbol{g}(\bar{\boldsymbol{u}}))^T F^{-1}EF^{-1}\nabla_u(\boldsymbol{g}(\bar{\boldsymbol{u}})), \end{aligned}$$

and rewrite equation (40) to

$$\frac{1}{2}\mathbf{z}^T E\mathbf{z} = a_g + \mathbf{b}_g^T\Delta\boldsymbol{u} + \frac{1}{2}\Delta\boldsymbol{u}^T H_g\Delta\boldsymbol{u}. \tag{41}$$

3.4 Complete Objective Function

It is possible to rewrite the objective function and the first constraint in (14), with equation (37) and (41), to

$$\min_{\Delta \boldsymbol{u}} \tilde{L}(\Delta \boldsymbol{u}) = a + \mathbf{b}^T \Delta \boldsymbol{u} + \frac{1}{2} \Delta \boldsymbol{u}^T H \Delta \boldsymbol{u}, \tag{42}$$

where

$$a \equiv a_h + a_g, \tag{43}$$
$$\mathbf{b} \equiv \mathbf{b}_h + \mathbf{b}_g, \tag{44}$$
$$H \equiv H_h + H_g. \tag{45}$$

The gradient and Hessian of L,

$$\begin{aligned} \nabla_u \tilde{L} &= \mathbf{b} + H \Delta \boldsymbol{u}, \\ \nabla_u^2 \tilde{L} &= H, \end{aligned} \tag{46}$$

have important roles for how the solution of the problem is constructed, see section 3.6.

3.5 Unique Optimum - Positive Definite Hessian

If H is positive definite then the function is convex, thus the optimization problem is much easier to solve. For a general diagonal matrix, M, whose elements are greater or equal to zero, it holds that

$$\mathbf{v}^T M \mathbf{v} = \sum_{i=1}^{n} m_{i,i} v_i^2 \geq 0, \tag{47}$$

First, let $H = H_g + H_h$ and hence $\mathbf{v}^T H \mathbf{v} = \mathbf{v}^T H_g \mathbf{v} + \mathbf{v}^T H_h \mathbf{v}$. The first term, H_g, can be written as

$$\mathbf{v}^T H_g \mathbf{v} = \mathbf{v}^T \left(\nabla_u (\boldsymbol{g}(\bar{\boldsymbol{u}})^T F^{-T} E F^{-1} \nabla_u (\boldsymbol{g}(\bar{\boldsymbol{u}})) \right) \mathbf{v} = \mathbf{x}^T E \mathbf{x} \geq \{(47)\} \geq 0, \tag{48}$$

where $\mathbf{x} \equiv F^{-1} \nabla_u (\boldsymbol{g}(\bar{\boldsymbol{u}}))\mathbf{v}$. Since the last inequality holds for all $\mathbf{x} \in \mathbb{R}^n$ and since $F^{-1} \nabla_u (\boldsymbol{g}(\bar{\boldsymbol{u}}))\mathbf{v} \subset \mathbb{R}^n$, the property follows. A similar methodology is used for all the terms in the second term H_h. In practice, the inequality is always strict and thus H is *positive definite*. To clarify this results means that the linearization, in a specific point, of the objective function is convex. This do not imply anything about the convexity of the unlinearized objective function. This means that in every specific point there exist a unique solution, but not that the problem in itself has a unique solution.

3.6 Optimization Solver

We will now turn our attention back to equation (14) where we concluded that the objective function and first constraint could be written as (42). Furthermore, if the last constraint in (14) is relaxed the optimization problem is reduced to

$$\min_{\Delta \boldsymbol{u}} \tilde{L}(\Delta \boldsymbol{u}) = \min_{\Delta \boldsymbol{u}} a + \mathbf{b}^T \Delta \boldsymbol{u} + \frac{1}{2} \Delta \boldsymbol{u}^T H \Delta \boldsymbol{u}, \tag{49}$$

which is an unconstrained optimization problem. The justification for the relaxation is that the constraint does not impact the optimal solution in practice. The most reasonable choice for the lower limit is U_l is that all elements should be greater than zero, which is far from feasible solutions and can therefore be relaxed. The optimization problem has good properties both with respect to the gradient and Hessian, which can be analytically derived, which was done in section 3.5. An algorithm for solving this problem can be constructed with a Newton method, a short description of this follows in the next section.

3.6.1 Solution Algorithm

An overview of the solving methodology is described is listed below.

0. Find a start solution, U_0 and set the counter k to zero.
1. Calculate $\mathbf{b}^T$ and H.
2. Check the stop criterion, $||\nabla_u L|| < \epsilon \Leftrightarrow ||\mathbf{b}|| < \epsilon$ and stop if true
3. Update solution
   (a) Solve the equation system, $\mathbf{b} + \nabla_u^2 L\Delta\boldsymbol{u} = \mathbf{0} \Leftrightarrow \mathbf{b} + H\Delta\boldsymbol{u} = \mathbf{0}$.
   (b) Determine the step length, s_k.
   (c) Take a step, $\boldsymbol{u}_{k+1} = \boldsymbol{u}_k + s_k\Delta\boldsymbol{u}$.
4. Update the counter, k=k+1 and go to step 1.

3.6.2 Practicalities and Starting Solution

In the derivation above, the lower limit U_l is assumed to be unbreached and is thus unused. Furthermore, the pricing function $\boldsymbol{g}$ is also approximated by linearization. These approximations might impact the functionality of the method, and will in that case be addressed after the error is detected in the implementation.

The starting solution, U_0 can be derived with several approaches. One simple approach is to set the surface to a constant value, since the optimization procedure should be able to handle any starting solution. Furthermore, the linearized objective function has a quadratic form and therefore the optimal step is always one. Since the linearization, equation (38), is an approximation the optimization procedure needs to be iterative. Furthermore, this step might be to long and the relaxed constraint, might by breached, which can be addressed with a shorter step length.

4 Utility Based Pricing

In 1944 von Neumann Morgenstern presented the expected utility maximization of consumption in the work Theory of Games and Economic Behavior, (von Neumann and Morgenstern, 2007). In essence, an investor chooses an optimal investment allocation with respect to its marginal utility of deferred consumption. Thus, the allocation is such that the marginal utility loss of consumption today is equal to marginal utility gain of accumulating wealth. In this framework, it is possible to derive a methodology to determine the price from the expected utility, (Cochrane, 2001). Thus, in this framework it is possible to determine the maximum expected utility of wealth.

4.1 Choice of Utility Function

The utility based pricing framework is consistent for an arbitrary choice of utility function. However, an inadequate selection will yield an incorrect price. Thus, depending on the investors preferences an appropriate utility function should be used. Due to pleasant properties, many applications use the power utility function,

$$\begin{cases} U(W_t) = \frac{1}{\gamma} W_t^{\gamma}, & \gamma \neq 0 \\ U(W_t) = \ln(W_t) & \gamma = 0, \end{cases} \tag{50}$$

where γ is the investors risk aversion. The case when $\gamma = 0$ the utility function is called *log-utility function.*

4.2 Optimal Growth

Kelly (1956), was first to present that the optimal long run portfolio is achieved by maximizing the log-utility function. This portfolio is sometimes known as the *Kelly portfolio* and it was later shown by Platen (2009) that this portfolio is identical to the *numeraire portfolio.* The numeraire portfolio, which is a benchmarked approach, was discovered by Long (1990). Furthermore, Platen showed that this portfolio cannot be systematically outperformed by any other portfolio and that it is *myopic.* Thus, the optimal investment is determined iteratively for every subsequent time increment in contrast to extracting the whole time span directly. The log-utility function also mentioned in (Luenberger, 1998, p. 425) as the utility function that maximizes growth, given a long investment horizon. Thus, an investor with the optimal policy with respect to the logarithmic utility function will be wealthier than other investors as time goes to infinity.

4.3 Pricing Formula and Relative Risk Aversion

An asset pricing formula can be derived from the first order conditions, for a given utility and optimal portfolio allocation of the expected wealth. The expression is given by

$$g_t = \frac{\mathrm{E}_t[U'(W_T) g_T]}{\mathrm{E}_t[U'(W_T)] e^{r_f^{t,T}(T-t)}}, \tag{51}$$

with the stochastic wealth, W_t, marginal utility $U'(\cdot)$, payout p_T at maturity T and the continuously compounded risk free spot interest rate $r_f^{t,T}$. The choice of the utility function is arbitrary, but the method is not *preference-free* and will therefore naturally affect risk aversion and thus influence the price. The log utility function is preferable since it outperforms all other portfolios with a long time horizon. Furthermore, it belongs to the *power-utility class*, which all have a constant relative risk aversion,

$$RRA(W) = -W \frac{U''(W)}{U'(W)} = W \frac{\gamma W^{-\gamma-1}}{W^{-\gamma}} = \gamma \frac{W^{\gamma+1}}{W^{\gamma+1}} = \gamma, \tag{52}$$

and is generally reasonable for modeling rational investors behavior.

4.3.1 Pricing Formula Derivation

In order to properly describe the pricing theory a definition of a optimal policy will be presented.

Definition 2. *Define an investment strategy for a portfolio of risky assets and the risk free asset for the time period $[0, T]$ such that the random wealth at time T is given by the stochastic variable W_T. This investment strategy is defined as an optimal policy with respect to the utility function $U(\cdot)$ if it maximizes the expected future utility, $E[U(W_T)]$.* □

With this definition in place is it possible to provide the theorem that is the foundation for the utility based pricing methodology. The theorem and proof following was presented to us by Blomvall (2007).

Theorem 1. *Let W_T be the random wealth at time T given an optimal policy for the time period $[0, T]$ with respect to the utility function $U(\cdot)$. Let $r_f^{t,T}$ be the continuous compounded return for an risk-free investment at time $t \in [0, T]$ that matures at time T. Let g_T be the random future price of an asset, then the price at which an investor will neither buy nor sell the asset at time $t \in [0, T]$ is given by*

$$g_t = \frac{\mathrm{E}_t[U'(W_T)g_T]}{\mathrm{E}_t[U'(W_T)]e^{r_f^{t,T}(T-t)}} \tag{53}$$

Proof. *With an initial wealth W_t at time $t \in [0, T]$ then an investor can invest in the optimal policy, the asset and the risk-free asset. Given a portfolio allocation of α, β and γ the future portfolio value is given by*

$$V_T = W_t \left(\alpha \frac{W_T}{W_t} + \beta \frac{g_T}{g_t} + \gamma R \right) \tag{54}$$

where g_t is the asset's price at time t, and R is the risk-free growth $e^{r_f^{t,T}(T-t)}$ and $\alpha + \beta + \gamma = 1$. The rational investor would maximize the expected future utility which can be formulated as

$$\max_{\alpha,\beta,\gamma} \mathrm{E}_t[U(V_T)] = \max_{\alpha,\beta} \mathrm{E}_t \left[U \left(W_t \left(\alpha \left(\frac{W_T}{W_t} - R \right) + \beta \left(\frac{g_T}{g_t} - R \right) + R \right) \right) \right] = \max_{\alpha,\beta} f \tag{55}$$

The optimality conditions are

$$\frac{\partial f}{\partial \alpha} = \mathrm{E}_t \left[U' \left(W_t \left(\alpha \left(\frac{W_T}{W_t} - R \right) + \beta \left(\frac{g_T}{g_t} - R \right) + R \right) \right) W_t \left(\frac{W_T}{W_t} - R \right) \right] = 0 \tag{56}$$

$$\frac{\partial f}{\partial \beta} = \mathrm{E}_t \left[U' \left(W_t \left(\alpha \left(\frac{W_T}{W_t} - R \right) + \beta \left(\frac{g_T}{g_t} - R \right) + R \right) \right) W_t (\frac{g_T}{g_t} - R) \right] = 0. \tag{57}$$

Observe that $\beta = 0$ as long as the asset is correctly priced. Put this into (56) which results in

$$\left. \frac{\partial f}{\partial \alpha} \right|_{\beta=0} = \mathrm{E}_t \left[U' \left(W_t \left(\alpha \left(\frac{W_T}{W_t} - R \right) + R \right) \right) W_t \left(\frac{W_T}{W_t} - R \right) \right] = 0 \tag{58}$$

this is the optimality condition for

$$\max_{\alpha+\gamma=1} \mathrm{E}_t \left[U \left(W_t \left(\alpha \frac{W_T}{W_t} + \gamma R \right) \right) \right]. \tag{59}$$

Since W_T is an optimal policy with respect to $U(\cdot)$ we have that $\alpha = 1$. Finally with $\alpha = 1$ together with $\beta = 0$ put into (57)

$$\begin{aligned} \left. \frac{\partial f}{\partial \beta} \right|_{\alpha=1,\beta=0} &= \mathrm{E}_t \left[U' \left(W_t \frac{W_T}{W_t} \right) W_t \left(\frac{g_T}{g_t} - R \right) \right] = 0 \Leftrightarrow \\ g_t = \frac{\mathrm{E}_t[U'(W_T)g_T]}{\mathrm{E}_t[U'(W_T)]R} &= \frac{\mathrm{E}_t[U'(W_T)g_T]}{\mathrm{E}_t[U'(W_T)]e^{r_f^{t,T}(T-t)}}, \end{aligned} \tag{60}$$

which finishes the proof. □

5 Multinomial Tree

The implementation of the utility based pricing will be done in a discretization of continuous time. Given this approximation, a *grid* needs to be formed with proper characteristics. The multinomial tree will be constructed to align with the grid. In order to derive the characteristics some definitions are required.

5.1 Grid Definition

In section 2.2.3 a deterministic local volatility surface was defined. The domain of the surface will be referred to as a *grid*.

Definition 3. *The* ***strike spectrum*** *of the grid G is the set of possible strikes, $\mathcal{K}_G = \{K_0, \ldots, K_{n_k}\}$, where $0 \leq K_0 < K_1 < \ldots < K_{n_k}$. The* ***strike increments*** *are defined as*

$$\Delta_{K_i} = K_{i+1} - K_i > 0, \ \forall i = 0, 1, \ldots, n_k - 1. \tag{61}$$

Note that it is generally practical that this spectrum is constructed to align with the options' strikes, whilst having adequate resolution.

Definition 4. *The* ***time spectrum*** *of the grid G is the set of possible times, $\mathcal{T}_G = \{T_0, \ldots, T_{n_t}\}$, where $T_{n_T} > T_{n_T-1} > \ldots > T_0 \geq 0$, where T_0 is known as the reference time. The* ***time increments*** *of continuous points in time are defined as*

$$\Delta_{T_i} = T_{i+1} - T_i > 0, \ \forall i = 0, 1, \ldots, n_t - 1. \tag{62}$$

Definition 5. *The* ***grid****, G, is a (discrete) two dimensional space that contains the two-dimensional spectrum of time and strike combinations*

$$G_{\mathcal{K}_G, \mathcal{T}_G} = \{\mathbf{g} = (K, T) : K \in \mathcal{K}_G, \ T \in \mathcal{T}_G\}, \tag{63}$$

where the points $\mathbf{g}$ *are called* ***nodes****, figure 1 presents a schematic picture of a grid.*

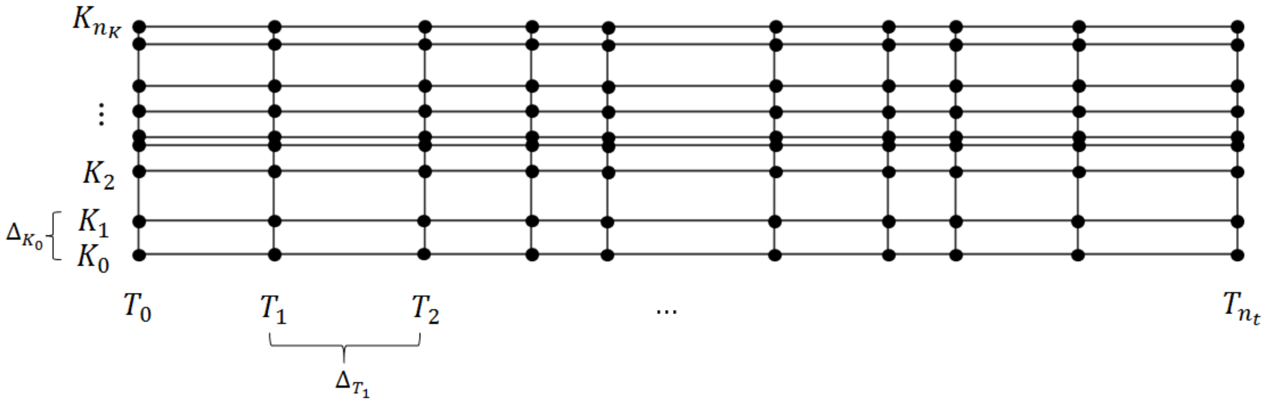


Figure 1: A schematic representation of a non uniform *discrete* grid, where the nodes are represented as black dots.

Definition 6. *An* ***equidistant grid*** *that satisfies the properties,* ***equidistant with respect to strike price****,*

$$\Delta_{K_i} = \Delta_{K_j}, \forall i = 0, \ldots, n_k - 1, \ \forall j = 0, \ldots, n_{k-1}. \tag{64}$$

and ***equidistant with respect to time****,*

$$\Delta_{T_i} = \Delta_{T_j}, \forall i = 0, \ldots, n_T - 1, \ \forall j = 0, \ldots, n_{T-1}. \tag{65}$$

5.2 From Grid to Tree

The nodes in the grid can be connected with *branches* of the corresponding tree. Furthermore, the *root node* is the node that corresponds to the price at the reference time. The *child nodes* are assumed to be linked with a branch to the subsequent time. Additionally, the branches are *cohesive*, when all intermediate strikes between K_{min} and K_{max} are accessed by branches from the given node.

5.2.1 Transition Probabilities

The branches can be assigned with a transition probability from a given node to the appropriate child nodes.
Definition 7. *The* ***transition probability****, $p_{i,t}^{j,\tau}$, links a given node (S_i, T_t) to the child node (S_j, T_τ). The branches must fulfill the conditions*

$$\begin{aligned} &0 \le p_{i,t}^{j,\tau} \le 1, \quad \forall i, j = 0, \dots, n_k,\ \forall t = 0, \dots, n_T - 1,\ \forall \tau = 1, \dots, n_T \\ &\sum_{j=0}^{n_K} p_{i,t}^{j,\tau} = 1, \quad \forall i = 0, \dots, n_k,\ \forall t = 0, \dots, n_T - 1,\ \forall \tau = 1, \dots, n_T. \end{aligned} \tag{66}$$

The notation is in most cases simplified to $p_{i,t}^{j} \equiv p_{i,t}^{j,t+1}$.

The tree will model how the underlying moves in time and therefore it is essential that the tree accurately models the distribution of the underlying, this is accomplished by determining adequate transition probabilities.

5.2.2 Logarithmic Changes

The logarithmic changes can be used to link the strikes of the underlying and their corresponding difference.
Definition 8. *The* ***logarithmic change***

$$r_{i,t}^{j,\tau} \equiv \ln\left(\frac{S_j}{S_i}\right) \Leftrightarrow S_i e^{r_{i,t}^{j,\tau}} = S_j,\ \forall i, j = 0, \dots, n_k,\ \forall t = 0, \dots, n_T - 1,\ \forall \tau = 1, \dots, n_T \tag{67}$$

is the return from node (S_i, t) to (S_j, τ). If the index τ is absent, the time increment is implied to be subsequent, $r_{i,t}^{j} \equiv r_{i,t}^{j,t+1}$.

5.3 Resolution

It is essential that the tree spans the central moments for the distribution. A practical, but not necessary, property is that the tree nodes match the strike levels of the options and the reference price of the underlying at the initial time. There is a trade off between speed and adding branches and hence accuracy. The transition probabilities assigned to branches should be meaningful and thus there is a limit where branches mainly add to computational complexity.

5.4 Bounded Tree

The grid can not be of infinite size and therefore the tree is also limited. This limitation can be addressed in several ways. A trivial approach is to limit the maximal growth to the border of the grid, where the probabilities are absorbed. However, this will induce difficulties while matching the proper distribution and it will be impossible to assign positive probabilities that fulfill the Kolmogorov axioms, see appendix B.4 or (Gut, 2009, p.4).

Another solution is to add a *moat*, above and below the initial grid, which have special properties, while the behavior in the initial grid is unchanged. In this moat the transition probabilities can be set such that the process remains constant at that price. With this approach it is also impossible to match a distribution, but the probabilities are not reflected back into the tree. The advantage of the second approach is that the probabilities remain in the moat and to not reenter the grid in a later time causing an inadequate distribution.

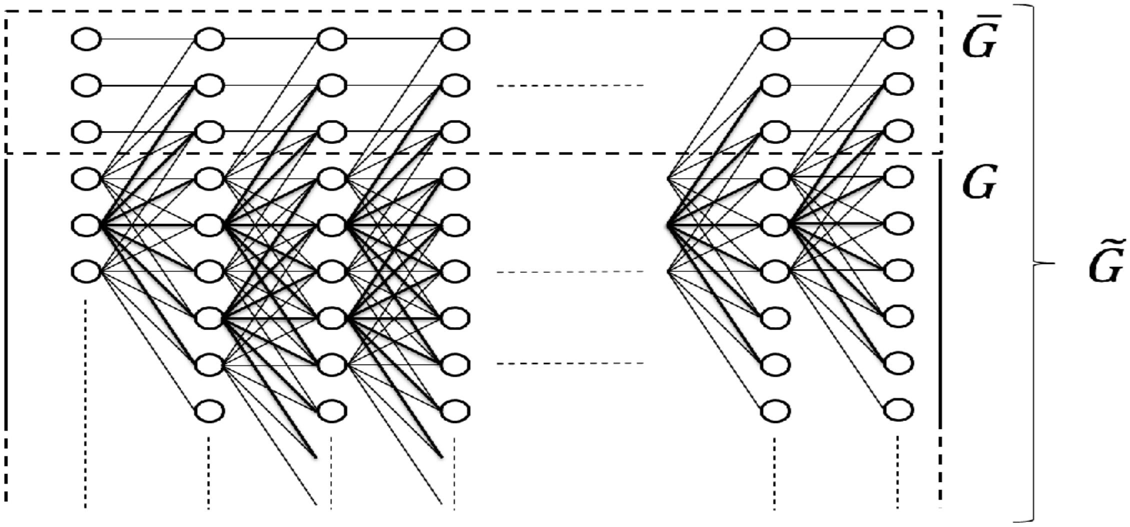


Figure 2: The upper part of a multinomial tree with seven branches and extended with a three strike wide moat

In order to apply the moat to the grid either an existing grid can be modified or two additional grids can be added to extend the original grid. The latter is found to be more convenient, since it can be added to an initial grid where there is no need for modification or redefinition.

Definition 9. *A grid $\bar{G}$ is a **strike extension** of the grid G if $\mathcal{T}_G = \mathcal{T}_{\bar{G}}$ and $\mathcal{K}_G \cap \mathcal{K}_{\bar{G}} = \emptyset$ (the grid G is mutual a strike extension of the grid $\bar{G}$).*

Figure 2 presents a schematic grid with an upper *grid extension* with respect to strike. It is essential that the probability to reach the moat is small to retain a proper distribution proxy. The moat will be presented mathematically in section 9.1.4. A grid G that is extended is denoted by $\tilde{G}$.

Remark 1. *Moat is a property of the grid's tree and a grid extension is just a part of the grid that may have the moat property. In practice, the parts of the tree that is in the extension are assigned with the moat property.*

6 Assign Probabilities

The characteristics of the probabilities in the mathematical framework are essential in the derivation of the optimization problem. Furthermore, an initial probability solution, $\bar{\mathbf{p}}$, will be adjusted as a minimal adjustment under required conditions. After the chapters derivation, the solution of the minimal adjustment ϵ is presented in a matrix form.

6.1 Minimal Adjustment

The probabilities defined in the previous chapter must be properly determined and carry properties such as an adequate distribution. A minimal adjustment is designed to ensure that these conditions are satisfied. Furthermore, the probabilities will be written on the form $\mathbf{p} = (\mathbf{1} + \boldsymbol{\epsilon})\bar{\mathbf{p}}$, where $\bar{\mathbf{p}}$ is an initial guess and $\boldsymbol{\epsilon}$ is an adjustment. The minimization problem is formed as

$$\begin{aligned} \min \quad & \frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} \\ \text{s.t.} \quad & X\boldsymbol{\epsilon} = \mathbf{y}. \end{aligned} \tag{68}$$

The problem (68) is analytically solvable, with a Lagrange multiplier, $\boldsymbol{\lambda}$,

$$L = \frac{1}{2}\boldsymbol{\epsilon}^T\boldsymbol{\epsilon} + \boldsymbol{\lambda}^T(\mathbf{y} - X\boldsymbol{\epsilon}). \tag{69}$$

The gradients are calculated and set equal to zero,

$$\nabla_{\epsilon} L = \boldsymbol{\epsilon} - X^T\boldsymbol{\lambda} = 0 \tag{70}$$

$$\nabla_{\lambda} L = \mathbf{y} - X\boldsymbol{\epsilon} = 0, \tag{71}$$

and from (70) we get

$$\boldsymbol{\epsilon} = X^T\boldsymbol{\lambda} \overset{(71)}{\Leftrightarrow} -XX^T\boldsymbol{\lambda} + \mathbf{y} = 0 \Leftrightarrow \boldsymbol{\lambda} = (XX^T)^{-1}\mathbf{y}, \tag{72}$$

which inserted in (70) gives

$$\boldsymbol{\epsilon} = X^T(XX^T)^{-1}\mathbf{y}. \tag{73}$$

Remark 2. *The matrix X is generally non-square and therefore the inverse of X does not exist. If X is a square matrix, where the inverse exists, it is possible to solve the problem directly from* (71)*, $\boldsymbol{\epsilon} = X^{-1}\mathbf{y}$.*

6.2 Expected Return with Respect to Volatility

The expression for the expected return will be derived for the power utility class since it is crucial for both the drift condition and the initial probability guess.

6.2.1 Optimal Portfolio Allocation

The equation (51) requires the *optimal policy*, which is optimally managed with respect to the given utility function. Bodie, Marcus, and Kane (2014, p. 294) state that given this policy all investors allocate all their investments in the market portfolio since the net lending must be zero. Hence, the choice of allocation in the market portfolio and the risk-free asset is already predetermined as a full market portfolio allocation in every investment period.

6.2.2 Expected Return

The second central moment for the normal distribution is given by the local volatility surface. However, the expected return must be derived in some other way. One approach is to formulate an optimization problem, which is similar to a *capital asset pricing model (CAPM)* derivation. The derivation adapts a maximized utility of a single investment period with respect to expected return and variance. In this setting the markets are frictionless efficient markets with full information, no tax, zero transaction costs and unlimited borrowing and lending. The rest of this section, 6.2.2, was explained by Blomvall (2007).

Let the initial wealth be W_0 allocated in the asset weights, including in the risk free asset, $\bar{\mathbf{w}}$ with the returns $\mathbf{r}$. The optimization problem can now be formed as

$$\begin{aligned} \max_{\bar{\mathbf{w}}} \quad & \mathrm{E}\left[U(W_0 \cdot \mathbf{r}^T\bar{w})\right] \approx \boldsymbol{\mu_r}^T\bar{\mathbf{w}} + \frac{\gamma - 1}{2}\bar{\mathbf{w}}^T\bar{C}\bar{\mathbf{w}} \\ \text{s.t.} \quad & \mathbf{1}^T\bar{\mathbf{w}} = 1, \end{aligned} \tag{74}$$

where the approximation is a Taylor expansion, assuming a power utility function, see (Merton and Samuelson, p. 81), and where $\boldsymbol{\mu_r} = \mathrm{E}[\mathbf{r}]$ and $\bar{C} = \mathrm{Var}[\mathbf{r}]$. We have also used the property that the power utility function has constant relative risk aversion independent of the size of wealth, see section 4.3.1. The risk-free assets have zero variance and also no covariance with any other asset. Furthermore, the weights $\bar{\mathbf{w}}$ can be divided into the risky asset weights $\mathbf{w}$ and risk free y, with the corresponding expected return $\boldsymbol{\mu}$ and r_f. The covariance of the risky assets is denoted by C and γ is the risk aversion parameter. From these we can formulate an optimization problem,

$$\begin{aligned} \min_{\mathbf{w}\in\mathbb{R}^n, y\in\mathbb{R}} \quad & L = \mathbf{w}^T\boldsymbol{\mu} + y \cdot r_f + \frac{1-\gamma}{2}\mathbf{w}^T C\mathbf{w} \\ \text{s.t.} \quad & \mathbf{1}^T\mathbf{w} + y = 1, \end{aligned} \tag{75}$$

where $\mathbf{w}$ is the weights in risky assets and y is the weight in risk-free assets. The weight y in the objective function can be replaced by the constraint and reformulated to a unconstrained optimization problem

$$\min_{\mathbf{w}\in\mathbb{R}^n} \quad L = \mathbf{w}^T\boldsymbol{\mu} + (1 - \mathbf{1}^T\mathbf{w}) \cdot r_f + \frac{1-\gamma}{2}\mathbf{w}^T C\mathbf{w}. \tag{76}$$

This unconstrained optimization can be solved by

$$\nabla_w L = 0 \Leftrightarrow \boldsymbol{\mu} - \mathbf{1} \cdot r_f - (1-\gamma)C\mathbf{w} = 0 \Leftrightarrow \mathbf{w} = \frac{1}{1-\gamma}C^{-1}\left(\boldsymbol{\mu} - \mathbf{1} \cdot r_f\right) \equiv \mathbf{w}^* \tag{77}$$

with the Hessian

$$H_w = \nabla_w^2 L = -(1-\gamma) \cdot C. \tag{78}$$

The covariance matrix C is positive definite and thus $\nabla_w^2 L$ is also positive definite for all $\gamma < 1$, which is the region of interest. Since the $\nabla_w^2 L$ is a positive definite we know that the extreme point $\mathbf{w}^*$ is a unique minimum. The point $\mathbf{w}^*$ can be rewritten, by multiplying with a scalar (a fraction of identical numerator and denominator of dimension 1),

$$\mathbf{w}^* = \frac{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)}{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)} \cdot \frac{1}{1-\gamma}C^{-1}\left(\boldsymbol{\mu} - \mathbf{1} \cdot r_f\right) = \frac{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)}{1-\gamma} \frac{C^{-1}\left(\boldsymbol{\mu} - \mathbf{1} \cdot r_f\right)}{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)} \tag{79}$$

where we can identify

$$\eta = \frac{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)}{1-\gamma} \tag{80}$$

and

$$w_M = \frac{C^{-1}\left(\boldsymbol{\mu} - \mathbf{1} \cdot r_f\right)}{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)}, \tag{81}$$

where w_M is the asset weights of the market portfolio and η is a scalar stating the leverage in the market portfolio. Furthermore, we have already stated in the previous section 6.2.1, that there is no investment in the risk free asset and thus $\eta = 1$, which inserted in (80) gives,

$$1 = \frac{\mathbf{1}^T C^{-1}(\boldsymbol{\mu} - \mathbf{1} \cdot r_f)}{1-\gamma}. \tag{82}$$

Furthermore, CAPM gives that $\mu = \mathbf{1} \cdot r_f + \frac{C\mathbf{w}_M}{\sigma_M^2}(\mu_M - r_f)$ and therefore (82) can be written as

$$\begin{aligned} 1 &= \frac{\mathbf{1}^T C^{-1}\left(\mathbf{1} \cdot r_f + \frac{C\mathbf{w}_M}{\sigma_M^2}(\mu_M - r_f) - \mathbf{1} \cdot r_f)\right)}{1-\gamma} \Leftrightarrow \\ 1-\gamma &= \frac{\mathbf{1}^T\mathbf{w}_M(\mu_M - r_f)}{\sigma_M^2} \Leftrightarrow \left\{\mathbf{1}^T\mathbf{w}_M = 1\right\} \Leftrightarrow \mu_M = r_f + (1-\gamma)\sigma_M^2, \end{aligned} \tag{83}$$

where the market return $\mu_M = \mu_g + \delta$, dividend δ and capital gain μ_g. Equation (83) can then be formulated as

$$\mu_g = r_f - \delta + (1-\gamma)\sigma_M^2. \tag{84}$$

The assets that will be studied are indices with no reinvestment, but where the underlying asset parts can have dividends. Thus, the underlying index indirectly has dividends, which will be approximated by a dividend yield reducing the drift component. Hence, in essence the expected value for the distribution is μ_g.

6.3 Initial Probability

The initial probabilities can be seen as an almost correct guess for the specific distribution. These probabilities are then adjusted by minimization of change. For convenience, some sets of index integers are defined. The first set $\mathcal{I}_G = \{0, 1, \dots, n_K\}$ contains all the index integers of the *strike spectrum* for the grid G, which can be extended. The set $\mathcal{I}_G^{i,t}$ contains the index of child nodes for the given node $(S_i, T_t), \forall t \in \mathcal{J}_G = \{0, \dots, n_T - 1\}, \forall i \in \mathcal{I}_G$. The notation G will be dropped, when unnecessary.

We here assume that it is the normal distribution that is matched. A way of approximating the normal distribution is to evaluate the probability density function and then normalize it with the sum of elements,

$$\begin{cases} \bar{p}_{i,t}^j = \frac{\phi_{\hat{\mu},\hat{\sigma}}(r_{i,t}^j)}{\sum_{k \in \mathcal{I}^{i,t}} \phi_{\hat{\mu},\hat{\sigma}}(r_{i,t}^j)} & \forall j \in \mathcal{I}^{i,t} \\ 0 & \forall j \notin \mathcal{I}^{i,t} \end{cases}, \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G, \tag{85}$$

where $\phi_{\hat{\mu},\hat{\sigma}}(\cdot)$ denotes the probability density function of a distribution with expected return $\hat{\mu}$ and standard deviation $\hat{\sigma}$ adjusted to the time increment. The normal probability density function can be expanded to

$$\begin{aligned} \phi_{\hat{\mu}_{i,t},\hat{\sigma}_{i,t}}(r_{i,t}^j) &= \frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} \exp\left(-\frac{(\hat{\nu}_{i,t} - r_{i,t}^j)^2}{2\hat{\sigma}_{i,t}^2} \right) \overset{\substack{\text{(84) and} \\ \text{(Ekblom, 2014)}}}{=} \left\{ \hat{\nu}_{i,t} = r_f^t - \delta^t + (1-\gamma)\hat{\sigma}_{i,t}^2 - \frac{1}{2}\hat{\sigma}_{i,t}^2 \right\} \\ &= \frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} \exp\left(-\frac{((\frac{1}{2}-\gamma)\hat{\sigma}_{i,t}^2 + r_f^t - \delta^t - r_{i,t}^j)^2}{2\hat{\sigma}_{i,t}^2} \right) = \left\{ a^t = r_f^t - \delta^t \right\} \\ &= \frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} \exp\left(-\frac{(\frac{1}{2}-\gamma)^2\hat{\sigma}_{i,t}^2}{2} - (\frac{1}{2}-\gamma)a^t - \frac{(a^t)^2}{2\hat{\sigma}_{i,t}^2} + \frac{ar_{i,t}^j}{\hat{\sigma}_{i,t}^2} + (\frac{1}{2}-\gamma)r_{i,t}^j - \frac{(r_{i,j}^t)^2}{2\hat{\sigma}_{i,t}^2} \right) \\ &= \left\{ b_{i,t}(\hat{\sigma}_{i,t}) = -\frac{(\frac{1}{2}-\gamma)^2\hat{\sigma}_{i,t}^2}{2} - (\frac{1}{2}-\gamma)a^t - \frac{(a^t)^2}{2\hat{\sigma}_{i,t}^2} \quad , \quad c_{i,j}^t(\hat{\sigma}_{i,t}) = \frac{a^t r_{i,t}^j}{\hat{\sigma}_{i,t}^2} - \frac{(r_{i,t}^j)^2}{2\hat{\sigma}_{i,t}^2} \right\} \\ &= \frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} e^{b_{i,t}(\hat{\sigma}_{i,t})} e^{c_{i,j}^t(\hat{\sigma}_{i,t})} e^{(\frac{1}{2}-\gamma)r_{i,t}^j}, \ \forall (i,t,j) \in \mathcal{I}_G \times \mathcal{J}_G \times \mathcal{I}^{i,t}, \end{aligned} \tag{86}$$

where r_f^t is the risk-free rate at time t. Then, (86) can be inserted in (85) to give the probability,

$$\begin{cases} \bar{p}_{i,t}^j = \dfrac{\frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} e^{b_{i,t}(\hat{\sigma}_{i,t})} e^{c_{i,j}^t(\hat{\sigma}_{i,t})} e^{(\frac{1}{2}-\gamma)r_{i,t}^j}}{\sum\limits_{k \in \mathcal{I}^{i,t}} \frac{1}{\sqrt{2u\hat{\sigma}_{i,t}^2}} e^{b_{i,t}(\hat{\sigma}_{i,t})} e^{c_{i,k}^t(\hat{\sigma}_{i,t})} e^{(\frac{1}{2}-\gamma)r_{i,t}^k}} = \dfrac{e^{c_{i,j}^t(\hat{\sigma}_{i,t})} e^{(\frac{1}{2}-\gamma)r_{i,t}^j}}{\sum\limits_{k \in \mathcal{I}^{i,t}} e^{c_{i,k}^t(\hat{\sigma}_{i,t})} e^{(\frac{1}{2}-\gamma)r_{i,t}^k}}, & \forall j \in \mathcal{I}^{i,t} \\ 0 & \forall j \notin \mathcal{I}^{i,t} \end{cases}, \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G. \tag{87}$$

6.3.1 Probability Preservation

The Kolmogorov axiom must be fulfilled for the assigned probabilities. The two axioms of concern are non-negativity and normalization, where the third, finite additivity, is true. The probabilities can be written as scalars,

$$p_{i,t}^j \equiv \bar{p}_{i,t}^j(1 + \epsilon_{i,t}^j) \quad \forall (i,t,j) \in \mathcal{I}_G \times \mathcal{J}_G \times \mathcal{I}^{i,t}. \tag{88}$$

Kolmogorov's first axiom postulates that all probabilities must be greater than zero,

$$p_{i,t}^j = \bar{p}_{i,t}^j(1 + \epsilon_{i,t}^j) \geq 0 \Leftrightarrow \begin{cases} \epsilon_{i,t}^j \geq -1 & \bar{p}_{i,t}^j > 0 \\ \epsilon \in \mathbb{R} & \bar{p}_{i,t}^j = 0 \end{cases} \quad \forall (i,t,j) \in \mathcal{I}_G \times \mathcal{J}_G \times \mathcal{I}^{i,t} \tag{89}$$

and Kolmogorovs second axiom postulates that the probability of the whole sample space is one,

$$\sum_{j \in \mathcal{I}^{i,t}} p_{i,t}^j = 1 \Leftrightarrow 1 + \sum_{j \in \mathcal{I}^{i,t}} \bar{p}_{i,t}^j \epsilon_{i,t}^j = 1 \Leftrightarrow \sum_{j \in \mathcal{I}^{i,t}} \bar{p}_{i,t}^j \epsilon_{i,j}^t = 0, \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G. \tag{90}$$

6.4 Adapting Probabilities by Moment Matching

The distribution needs to match the central moments to be appropriately similar to a specific distribution. Note however, that it is not always possible to match to moments given any sample space, see section 9.1.3.3 for conditions that must be satisfied. Furthermore, finding the probabilities $p^j_{i,t}$ between time steps for each nodes is done by moment matching for the first n_m central moments $\{m^c : c = 2, \dots n_m\}$ such that the discrete outcomes

$$m^c_{i,t} = E[(X - E[X])^c] = \sum_{j \in \mathcal{I}^{i,t}} p^j_{i,t} \cdot (r^j_{i,t} - E[X])^c \quad \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G, \tag{91}$$

where X is the stochastic variable for the returns. The transition probabilities in the moat are by construction

$$\begin{cases} p^j_{i,t} = 1 & j = i \quad i \in \mathcal{I}_{\tilde{G}} \setminus \mathcal{I}_G \\ p^j_{i,t} = 0 & j \neq i \quad i \in \mathcal{I}_{\tilde{G}} \setminus \mathcal{I}_G. \end{cases} \tag{92}$$

6.4.1 Special Case - Normal Distribution

Note that for a normally distributed variable the central moments m^c can be calculated as

$$m^c_{i,t} = \begin{cases} \hat{\sigma}^c_{i,t} \cdot \sqrt{\Delta_{T\,i}} \cdot c!! & c \textit{ is even} \\ 0 & c \textit{ is odd} \end{cases}, \ \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G. \tag{93}$$

where

$$c!! = \begin{cases} (c-1) \cdot (c-2)!! & c \geq 3 \\ 1 & c \leq 2. \end{cases} \tag{94}$$

The condition can then be written as

$$\sum_{j \in \mathcal{I}^{i,t}} \bar{p}^j_{i,t}(1 + \epsilon^j_{i,t}) \cdot (r^j_{i,t} - \hat{\mu}_t)^c = m^c_{i,t}, \quad \forall c \in \{1 ... n_c\}, \forall (i,t) \in \mathcal{I}_G \times \mathcal{J}_G. \tag{95}$$

6.5 Probability Adjustment

The central moments are not matched if the initial probabilities in (85) are chosen. Therefore, the probabilities are adjusted by minimization of change. For a cleaner look, the time notation is dropped, but this optimization problem is done for all nodes. Let

$$p^j_{i,t} \equiv \bar{p}^j_{i,t}(1 + \epsilon^j_{i,t}) \quad \forall (i,t,j) \in \mathcal{I} \times \mathcal{J} \times \mathcal{I}^{i,t}, \tag{96}$$

which is used in the next section for defining a minimization problem for one special case, $i \in \mathcal{I}_G$ and $j \in \mathcal{J}_G$.

6.5.1 Full Allocation Condition

The full allocation condition in the market portfolio does not necessarily hold by the definitions made. Therefore, additional constrains are required to ensure that the allocation, $\alpha = 1$. Thus, we need to find the conditions on the probabilities that maximizes the expected utility of wealth and $\alpha = 1$. We start from equation (55), with $\beta = 0,\ V_t|_{\beta=0} = W_t \left(\alpha \frac{W_T}{W_t} + \beta \frac{g_T}{g_t} + \gamma R \right) \Big|_{\beta=0} = W_t \left(\alpha \frac{W_T}{W_t} + \gamma R \right) \equiv V^*_t$.

$$\begin{aligned} \max_{\gamma, \alpha \in \mathbb{R}} \quad & \mathrm{E}_t[U(V^*_t)] \\ \text{s.t.} \quad & \alpha + \gamma = 1. \end{aligned} \tag{97}$$

With $\gamma = 1 - \alpha$, we can rewrite (97) as

$$\max_{\alpha \in \mathbb{R}} \mathrm{E}_t \left[U \left(W_t \left(\alpha \frac{W_T}{W_t} + (1 - \alpha) R \right) \right) \right] = \max_{\alpha \in \mathbb{R}} \mathrm{E}_t \left[U \left(\alpha W_T + W_t (1 - \alpha) R \right) \right]. \tag{98}$$

For a discrete sample space $\Omega = \left\{W_T^1, W_T^2, \dots, W_T^n\right\}$, where n is a finite number it is possible to write the expectation as

$$\max_{\alpha\in\mathbb{R}} \mathrm{E}_t\left[U\left(\alpha W_T + W_t(1-\alpha)R\right)\right] = \max_{\alpha\in\mathbb{R}} \sum_{i=1}^{n} U(W_T^i\alpha + W_t(1-\alpha)R)p_i \equiv \max_{\alpha\in\mathbb{R}} f(\alpha). \tag{99}$$

Furthermore, the problem is one-dimensional and unconstrained. We will show that there exists an unique global optima with $f'' < 0$ and $f' = 0$. The first and second derivative can be written as

$$f'(\alpha) = \sum_{i=1}^{n} U'(W_T^i\alpha + W_t(1-\alpha)R)(W_T^i - RW_t)p_i \tag{100}$$

and

$$f''(\alpha) = \sum_{i=1}^{n} U''(W_T^i\alpha + W_t(1-\alpha)R)(W_T^i - RW_t)^2 p_i. \tag{101}$$

A special case of the second derivative is when U is the log-utility function. The expression (101) has three factors, the third, p_i is by assumption in the interval $[0,1]$[2]. The second factor is greater or equal than zero since it is quadratic. The first factor is the second derivative of the log-utility function,

$$U''(x) = -\frac{1}{x^2} < 0,\ \forall x \in \left\{\mathbb{R}\setminus\{0\}\right\}, \tag{102}$$

where $\alpha W_T + W_t(1-\alpha)R > 0$, since the logarithmic function U is no defined for non-positive values. Therefore the singularity is no problem. Furthermore, we conclude that $f'' \leq 0$, and if the sum includes more than one term the inequality is strict, which are the case in practice. Hence, any extreme point, α^*, is an unique optima, $f(\alpha^*) > f(\bar{\alpha}),\ \forall\bar{\alpha} \in \mathbb{R}\setminus\{\alpha^*\}$.

Since an extreme point for $\alpha = 1$, where $f'(\alpha) = 0$, is optimal, we set

$$\begin{aligned} f'(\alpha)|_{\alpha=1} = 0 \Leftrightarrow \sum_{i=1}^{n} U'(W_T^i\alpha + W_t(1-\alpha)R)p_i\left(W_T^i - RW_t\right)\Bigg|_{\alpha=1} = 0 \Leftrightarrow \\ \sum_{i=1}^{n} U'\left(W_T^i\right)\left(W_T^i - RW_t\right)p_i = \left\{\tilde{d} := U'\left(W_T^i\right)\left(W_T^i - RW_t\right)\right\} = \sum_{i=1}^{n} \tilde{d}_i p_i = 0. \end{aligned} \tag{103}$$

In equation (96) we have $p_i = (1+\epsilon_i)\bar{p}_i$ and with the definition $\tilde{d} := \frac{1}{W_T^i}\left(W_T^i - RW_t\right)$ (given log-utility) rewrite equation (103) to

$$\begin{aligned} \sum_{i=1}^{n} \tilde{d}_i(1+\epsilon_i)\bar{p}_i = \sum_{i=1}^{n} \tilde{d}_i\bar{p}_i + \tilde{d}_i\bar{p}_i\epsilon_i = 0 \Leftrightarrow \left\{d_i = \tilde{d}_i\bar{p}_i\right\} \Leftrightarrow \\ \sum_{i=1}^{n} d_i = -\sum_{i=1}^{n} d_i\epsilon_i \Leftrightarrow \mathbf{d}^T\mathbf{1} = -\mathbf{d}^T\boldsymbol{\epsilon}. \end{aligned} \tag{104}$$

[2] This property is assured by another constraint in the optimization problem.

6.5.2 Matrix Representation

If the inequality constraint (89) is relaxed then only the equality constraints (104), (90) and (95) remain. This equation system can be written in matrix form as

$$X = \begin{bmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_{n_c}^T \\ \mathbf{p}^T \\ \mathbf{d}^T \end{bmatrix}, \quad \text{and } \mathbf{y} = \begin{bmatrix} m_1 - \mathbf{x}_1^T \mathbf{1} \\ m_2 - \mathbf{x}_2^T \mathbf{1} \\ \vdots \\ m_{n_c} - \mathbf{x}_{n_c}^T \mathbf{1} \\ 0 \\ -\mathbf{d}^T \mathbf{1}. \end{bmatrix}, \tag{105}$$

where

$$\mathbf{x}_c = \begin{pmatrix} \bar{p}_{i,t}^{\min(\mathcal{I}^{i,t})} \cdot (r_{i,t}^{\min(\mathcal{I}^{i,t})} - \hat{\mu}_{i,t})^c \\ \bar{p}_{i,t}^{\min(\mathcal{I}^{i,t})+1} \cdot (r_{i,t}^{\min(\mathcal{I}^{i,t})+1} - \hat{\mu}_{i,t})^c \\ \vdots \\ \bar{p}_{i,t}^{\max(\mathcal{I}^{i,t})-1} \cdot (r_{i,t}^{\max(\mathcal{I}^{i,t})-1} - \hat{\mu}_{i,t})^c \\ \bar{p}_{i,t}^{\max(\mathcal{I}^{i,t})} \cdot (r_{i,t}^{\max(\mathcal{I}^{i,t})} - \hat{\mu}_{i,t})^c \end{pmatrix}, \forall c = 1, \ldots, n_c. \tag{106}$$

These expressions can now be inserted in equation (73), $\boldsymbol{\epsilon} = X^T(XX^T)^{-1}\mathbf{y}$.

7 Pricing Formula Computation

With the utility based framework with optimal growth, in a discrete grid with adequate transition probabilities and distribution in place, the pricing formula can be defined. The function $\boldsymbol{g}$ is used to price the spectrum of put and call options. The pricing formula computation is defined and subsequently the components are derived. The function $\boldsymbol{g}$ is used to price the spectrum of put and call options and is defined as

$$\boldsymbol{g} = \Xi R_f^{-1}\mathbf{q}, \tag{107}$$

where R_f^{-1} is a discount factor, Ξ is the sample space of the option payoff and $\mathbf{q}$ is the risk-neutral distribution. The payoff and the discount factor is presented in this section, and the derivation of the risk-neutral probability is derived in the next section.

The payoff function for a call and put option are per definition

$$\xi_c(S_T, K) = \max\left(S_T - K, 0\right) \tag{108}$$

$$\xi_p(S_T, K) = \max\left(K - S_T, 0\right), \tag{109}$$

where S_T is the price of the underlying at the maturity time T and K is the option's strike price. If the sample space of the asset price, in time τ, is $\mathcal{S}_\tau = \{S_{1,\tau}, \ldots, S_{n_\tau,\tau}\}$, the sample space of the payoff becomes

$$\boldsymbol{\xi}_c\left(\mathcal{S}_\tau, K\right) = \big(\max\left(S_{1,\tau} - K, 0\right), \ldots, \max\left(S_{n_\tau,\tau} - K, 0\right)\big)^T \tag{110}$$

$$\boldsymbol{\xi}_p\left(\mathcal{S}_\tau, K\right) = \big(\max\left(K - S_{1,\tau}, 0\right), \ldots, \max\left(K - S_{n_\tau,\tau}, 0\right)\big)^T, \tag{111}$$

for a specific call and put option respectively. For a set of options with different strike prices and maturity times a matrix can be constructed,

$$\Xi = \begin{pmatrix} \boldsymbol{\xi}(\mathcal{S}_{\tau_1}, K_1^{\tau 1}) & & & \\ \vdots & & & \\ \boldsymbol{\xi}(\mathcal{S}_{\tau_1}, K_m^{\tau 1}) & & & \\ & \boldsymbol{\xi}(\mathcal{S}_{\tau_2}, K_1^{\tau 2}) & & \\ & \vdots & & \\ & \boldsymbol{\xi}(\mathcal{S}_{\tau_2}, K_m^{\tau 2}) & & \\ & & \ddots & \\ & & & \boldsymbol{\xi}(\mathcal{S}_{\tau_{n_\tau}}, K_1^{\tau n_\tau}) \\ & & & \vdots \\ & & & \boldsymbol{\xi}(\mathcal{S}_{\tau_{n_\tau}}, K_m^{\tau n_\tau}) \end{pmatrix}, \tag{112}$$

where l denotes the number of maturity times and $K_j^{\tau_i}$ is the jth strike level for the maturity time τ_i that has an option. The vector $\mathbf{q}$ contains the distributions for the time spectrum,

$$\mathbf{q} = \begin{pmatrix} \mathbf{q}_{\tau_1} \\ \mathbf{q}_{\tau_2} \\ \vdots \\ \mathbf{q}_{\tau_{n_\tau}} \end{pmatrix}. \tag{113}$$

Lastly the discount factors for a specific time can be written as $e^{-r_f^{T_{\tau_i}, T_{\tau_j}}(T_{\tau_i} - T_{\tau_j})}$ and the discount factor matrix, R_f^{-1}, can be written as

$$R_f^{-1} = \begin{pmatrix} e^{-T_{\tau_0}^{\tau 1} \cdot r_f^{T_{\tau_0}, T_{\tau_1}}} & & & & & & \\ & \ddots & & & & & \\ & & e^{-T_{\tau_0}^{\tau 1} \cdot r_f^{T_{\tau_0}, T_{\tau_1}}} & & & & \\ & & & e^{-T_{\tau_0}^{\tau 2} \cdot r_f^{T_{\tau_0}, T_{\tau_2}}} & & & \\ & & & & \ddots & & \\ & & & & & e^{-T_{\tau_0}^{\tau 2} \cdot r_f^{T_{\tau_0}, T_{\tau_2}}} & \\ & & & & & & \ddots \\ & & & & & & & e^{-T_{\tau_0}^{\tau n} \cdot r_f^{T_{\tau_0}, T_{\tau_n}}} \end{pmatrix}, \tag{114}$$

where $T_{\tau_0}^{\tau_1} \equiv T_{\tau_1} - T_{\tau_0}$, and expression (107) can be formulated.

7.1 Discrete Pricing Formula

Given an optimal allocation ($\alpha = 1$), the probabilities $p_{i,j}^t$, the risky returns $r_{i,j}^t$ and the risk-free rate $r_f^{T_t,T_{t+1}}$ the price of the derivative can be calculated for the time T_t. The pricing formula holds for all initial wealths W_t, and can be showed to be independent of it. The price of a derivative, g in time t, with the underlying's price S_k is g_t^k. If the probabilities and α given by the current node (S_k, T_t) is known and the log-utility function is used, then the price is given by

$$\begin{aligned} g_t^i = \frac{\mathrm{E}_t[U'(W_{t+1})g_{t+1}]}{\mathrm{E}_t[U'(W_{t+1})]e^{r_f^{T_t,T_{t+1}}\Delta T_t}} &= \frac{\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{g_{t+1}^j}{\left(\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}\right)W_t}}{e^{r_f^{T_t,T_{t+1}}\Delta T_t}\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{1}{\left(\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}\right)W_t}} \\ &= \frac{\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{g_{t+1}^j}{\left(\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}\right)}}{e^{r_f^{T_t,T_{t+1}}\Delta T_t}\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{1}{\left(\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}\right)}},\ \forall(i,t)\in\mathcal{I}\times\mathcal{J} \end{aligned} \tag{115}$$

which is completely independent of W_t, since we know that α is independent of W_t. By recursive calculations one finally get the current price of the derivative, g_0^k where $\{k : S_k = S_{ini}\}$.

7.2 Risk-neutral Probabilities Computation

The classic approach of pricing derivatives is the risk-neutral valuation principle which states that

$$g_t = \frac{\mathrm{E}_t^Q[g_{t+1}]}{e^{\int_{T_t}^{T_{t+1}} r(u)du}} \tag{116}$$

or the discrete case

$$g_t^i = e^{-r_f^{T_t,T_{t+1}}\Delta T_t} \sum_{j\in\mathcal{I}^{i,t}} q_{i,j}^t g_{t+1}^j \quad \forall(i,t)\in\mathcal{I}\times\mathcal{J} \tag{117}$$

where $q_i = Q(g_{t+1} = g_{t+1}^i|\mathcal{F}_t)$. Equating the corresponding utility based pricing formula (115) with (117) gives

$$\sum_{j\in\mathcal{I}^{i,t}} q_{i,t}^j g_{t+1}^j = \frac{\sum_{j\in\mathcal{I}^{i,t}} \frac{p_{i,t}^j}{\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}} g_{t+1}^j}{\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{1}{\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}}}, \quad \forall(i,t)\in\mathcal{I}\times\mathcal{J}. \tag{118}$$

This must hold for all instruments. If the instrument is an Arrow-Debreu security it has a payoff of one if the underlying coincides exactly with the strike at the given maturity, otherwise it expires worthless. The expression for $q_{i,j}^t$ is then found by identification in equation (118),

$$q_{i,t}^j = \frac{\frac{p_{i,t}^j}{\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^t\Delta t_t}}}{\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^j \frac{1}{\alpha e^{r_{i,t}^j}+(1-\alpha)e^{r_f^t\Delta t_t}}}, i\in\mathcal{I}^{\tilde{G}}, \quad \forall j\in\mathcal{I}^{i,t},\ \forall t=0,\ldots,n_T-1 \tag{119}$$

The expression (119) can be rewritten into a simplified and more direct form. We start with the cases $j \in \mathcal{I}^{i,t}$ and after that the trivial case of $j \notin \mathcal{I}^{i,t}$.

$$
\begin{aligned}
q_{i,t}^{j} &= \frac{\frac{p_{i,t}^{j}}{\alpha e^{r_{i,t}^{j}}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}}}{\sum_{j\in\mathcal{I}^{i,t}} p_{i,t}^{j}\frac{1}{\alpha e^{r_{i,t}^{j}}+(1-\alpha)e^{r_f^{T_t,T_{t+1}}\Delta T_t}}} = \left\{\alpha = 1, p_{i,t}^{j} = \bar{p}_{i,t}^{j}(1+\epsilon_{i,t}^{j})\right\} \\
&= \frac{\bar{p}_{i,t}^{j}(1+\epsilon_{i,t}^{j})e^{-r_{i,t}^{j}}}{\sum_{j\in\mathcal{I}^{i,t}}\left(\bar{p}_{i,t}^{j}(1+\epsilon_{i,t}^{j})e^{-r_{i,t}^{j}}\right)} = \{(87)\} = \frac{\left(\frac{e^{c_{i,j}^{t}(\hat{\sigma}_{i,t})}e^{(\frac{1}{2}-\lambda)r_{i,t}^{j}}}{\sum\limits_{k\in\mathcal{I}^{i,t}} e^{c_{i,k}^{t}(\hat{\sigma}_{i,t})}e^{(\frac{1}{2}-\lambda)r_{i,t}^{k}}}\right)(1+\epsilon_{i,t}^{j})e^{-r_{i,t}^{j}}}{\sum_{l\in\mathcal{I}^{i,t}}\left(\frac{e^{c_{i,j}^{t}(\hat{\sigma}_{i,t})}e^{(\frac{1}{2}-\lambda)r_{i,t}^{l}}}{\sum\limits_{k\in\mathcal{I}^{i,t}} e^{c_{i,k}^{t}(\hat{\sigma}_{i,t})}e^{(\frac{1}{2}-\lambda)r_{i,t}^{k}}}\right)(1+\epsilon_{i,t}^{l})e^{-r_{i,t}^{l}}} \\
&= \frac{e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}(1+\epsilon_{i,t}^{j})e^{-(\frac{1}{2}+\lambda)r_{i,t}^{j}}}{\sum_{l\in\mathcal{I}^{i,t}} e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}(1+\epsilon_{i,t}^{j})e^{-(\frac{1}{2}+\lambda)r_{i,t}^{j}}} = \{\lambda = 0\} = \frac{(1+\epsilon_{i,t}^{j})e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}}{\sum_{l\in\mathcal{I}^{i,t}}(1+\epsilon_{i,t}^{j})e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}}, \forall(i,t)\in\mathcal{I}\times\mathcal{J}.
\end{aligned}
\tag{120}
$$

The second equality is given by the optimal portfolio, which is the market portfolio with $\alpha = 1$, see section 6.2.1. When $j \notin \mathcal{I}^{i,t}$ then $q_{i,j}^{t} = 0$ and therefore the complete case can be written as

$$
q_{i,t}^{j} = \begin{cases} \frac{(1+\epsilon_{i,t}^{j})e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}}{\sum_{j\in\mathcal{I}^{i,t}}(1+\epsilon_{i,t}^{j})e^{c_{i,t}^{j}(\hat{\sigma}_{i,t})}} & j \in \mathcal{I}^{i,t} \\ 0 & j \notin \mathcal{I}^{i,t} \end{cases}, \forall(i,t)\in\mathcal{I}\times\mathcal{J} \tag{121}
$$

which in vector form is

$$
\mathbf{q}_{i,t} = \left(q_{i,t}^{0}, q_{i,t}^{1}, \ldots, q_{i,t}^{n}\right)^{T} \tag{122}
$$

where $n = |\mathcal{I}_{\tilde{G}}|$.

7.3 Transition Matrices

The risk-neutral probabilities described in the previous section were *transition probabilities*. A special case of these are the *distribution probabilities*,

$$\mathbf{q}_d^{(t)} = \begin{pmatrix} q_{ini,0}^{0,t} \\ q_{ini,0}^{1,t} \\ \vdots \\ q_{ini,0}^{n,t} \end{pmatrix}, \tag{123}$$

which are the transition probabilities from time 0 to a future time t. Furthermore, distribution probabilities at different times, t and $t+1$ can be connected with a *transition matrices*, Q^t, as

$$\mathbf{q}_d^{(t_{n+1})} = Q^t \mathbf{q}_d^{(t_n)}, \tag{124}$$

where

$$Q^t = \left(\mathbf{q}_0^t, \dots, \mathbf{q}_n^t\right) = \begin{pmatrix} q_{0,t}^0 & q_{1,t}^0 & \cdots & q_{n,t}^0 \\ q_{0,t}^1 & q_{1,t}^1 & \cdots & q_{n,t}^1 \\ \vdots & \vdots & \ddots & \vdots \\ q_{0,t}^n & q_{1,t}^n & \cdots & q_{n,t}^n \end{pmatrix} \tag{125}$$

If applied iteratively we get

$$Q^{t_n} Q^{t_{n-1}} \cdots Q^{t_1} Q^{t_0} \mathbf{q}_d^{(t_0)} = Q^{t_n} Q^{t_{n-1}} \cdots Q^{t_1} \mathbf{q}_d^{(t_1)} = \ldots = Q^{t_n} \mathbf{q}_d^{(t_n)} = \mathbf{q}_d^{(t_{n+1})}. \tag{126}$$

The distribution in the reference time, t_0, is known, since the underlying has a certain observed price with probability 1. The distribution at the initial time can be expressed as,

$$\begin{cases} \mathbf{q}_i^{(t_0)} = 1 & , i : S_{i,0} = S_{ini} \\ \mathbf{q}_i^{(t_0)} = 0 & \text{otherwise} \end{cases}. \tag{127}$$

We now have all the information required to use the pricing expression (107).

7.4 Interpolation Option Pricing

In order to price option in intermediate points in the grid, interpolation pricing can be used, described in Barkhagen (2015, pp. 206-209). The interpolation is needed since the risk-neutral distribution is discrete[3].

The call and put price for options whose strike price are presented in the grid is priced as described as above for all strikes in the grid. These are theoretical options and do not, in general, have a market price. The intermediate call options, with strike price K_k and maturity time T_l in the grid can now be priced, at time t, as

$$\begin{aligned} g_k(\mathbf{q}) = \lambda_k \left(C(\bar{K}_k) + \frac{\partial C(\bar{K}_k)}{\partial K}\left(K_k - \bar{K}_k\right) + \frac{1}{2}\frac{\partial^2 C(\bar{K}_k)}{\partial K^2}\left(K_k - \bar{K}_k\right)^2 \right) + \\ (1-\lambda_k)\left(C(\bar{K}_{k+1}) + \frac{\partial C(\bar{K}_{k+1})}{\partial K}\left(K_k - \bar{K}_{k+1}\right) + \frac{1}{2}\frac{\partial^2 C(\bar{K}_{k+1})}{\partial K^2}\left(K_k - \bar{K}_{k+1}\right)^2 \right), \end{aligned} \tag{128}$$

where λ_k is defined as

$$\lambda_k = \frac{\bar{K}_{k+1} - K_i}{\bar{K}_{k+1} - \bar{K}_k}. \tag{129}$$

Furthermore $\bar{K}_k$ ($\bar{K}_{k+1}$) are the largest (smallest) of all smaller (larger) strike prices in the grid. The call options with these strikes have the price $C(\bar{K}_k)$ ($C(\bar{K}_k)$) respectively. Furthermore first and second derivative are given by

$$\frac{\partial C(\bar{K}_j)}{\partial K} = -D(t, T_l)\int_{\bar{K}_k}^{\infty} q_c(S_{t_l}, t_l) dS_{t_l} \approx -D(t, T_l)\sum_{j=k}^{N} q(K_j, T_l) \tag{130a}$$

[3] Interpolation of the distribution might also be a solution

$$\frac{\partial^2 C(\bar{K}_j)}{\partial K^2} = D(t,T_l)q(\bar{K}_j) \approx D(t,T_l)\frac{2q_{K_j,T_l}}{\bar{S}_{j+1}-\bar{S}_{j-1}}, \tag{130b}$$

where $D(\tau, t)$ is the discount factor from time t to τ and where N is the total number of strike prices in the grid.

The put options are priced analogously as

$$\begin{aligned} g_k(\mathbf{q}) = \lambda_k \left(P(\bar{K}_k) + \frac{\partial P(\bar{K}_k)}{\partial K}\left(K_k - \bar{K}_k\right) + \frac{1}{2}\frac{\partial^2 P(\bar{K}_k)}{\partial K^2}\left(K_k - \bar{K}_k\right)^2 \right) + \\ (1-\lambda_k)\left(P(\bar{K}_{k+1}) + \frac{\partial P(\bar{K}_{k+1})}{\partial K}\left(K_k - \bar{K}_{k+1}\right) + \frac{1}{2}\frac{\partial^2 P(\bar{K}_{k+1})}{\partial K^2}\left(K_k - \bar{K}_{k+1}\right)^2 \right), \end{aligned} \tag{131}$$

where $P(x)$ denotes the price of an put option with the strike price x. The first and second derivative are given by

$$\frac{\partial P(\bar{K}_j)}{\partial K} = D(t,T_l)\left(1 - \int_{\bar{K}_j}^{\infty} q_c(S_{t_l}, t_l)dS_{t_l}\right) \approx D(t,T_l)\left(1 - \sum_{j=k}^{N} q_{K_j,T_l}\right) \tag{132a}$$

$$\frac{\partial^2 P(\bar{K}_j)}{\partial K^2} = D(t,T_l)q_c(\bar{K}_j, t_l) = \frac{\partial^2 C(\bar{K}_j)}{\partial K^2}. \tag{132b}$$

8 Automatic Differentiation

To evaluate a differentiation it is preferred to have an analytical expression to evaluate all points of interest. It is sometimes infeasible or impossible to find such a expression and thus other approaches must be used. One common approach is to use the a numerical approach, for instance the *finite difference*

$$\frac{\partial f}{\partial x_i}(x) \approx \frac{f(x+\epsilon)-f(x)}{\epsilon}, \tag{133}$$

which is an approximation. An alternative, exact[4], approach will be presented in this chapter, *automatic differentiation*[5], which is neither numerical or symbolic differentiation; it is just another way of calculating derivatives.

We use automatic differentiation to compute gradients, or more specifically the gradient $\nabla_u(\boldsymbol{\psi}(\boldsymbol{\bar{\pi}}))$, which is used for the linearizion of $\boldsymbol{\psi}$. This is done since it is complex to compute this gradient analytically. Computing gradients with automatic differentiation is more efficient and more precise compared to a finite difference method, (Rall and Corliss, 1996, p. 2). Additionally, it computes the gradient and function value simultaneously which is more efficient than calculating the function and its derivate separately.

Rall and Corliss (1996, pp. 2-4 and pp. 9-11) and Neidinger (2010, pp. 547-551 and pp. 558-561)[6] describe two modes of automatic differentiation, *forward* and *reverse* mode. Both modes are constructed with the *chain rule*. Depending on the problem size the modes have different computational efficiency, but in general they are equally accurate.

The difference in computational efficiency depends on the characteristics of the evaluated function. Rall and Corliss state that it is only optimal to use one or another in extreme cases and in general it is optimal to combine the methods. However, finding the optimal mix results in an optimization problem, which seldom results in a computational efficiency improvement. Furthermore, Rall and Corliss, (1996, p. 11), present a guideline,

"The reverse mode is generally favored if $m >> q$",

where m is the number of independent variables (input values) and q is the number of dependent variables (output values). Given these guidelines we will use the reverse mode, since we may have a relative difference of ~ 100 between independent variables and dependent variables. Since the reverse mode is of greatest interest for the thesis it is presented before the forward mode.

The chapter is an introduction and overview of automatic differentiation, for a comprehensive presentation see (Griewank and Walther, 2008) and for some details used in this thesis see (Forth et al., 2004, pp. 266-277) and (Neidinger, 2010, pp.558-561).

8.1 Reverse Mode

The reverse mode consists of two *sweeps*, a forward and a reverse sweep. The forward sweep calculates the function value and also creates a *tape*, which is often represented by a matrix, L, (Neidinger, 2010, p. 560) and (Forth et al., 2004, p. 270). The reverse sweep calculates the derivatives, from which the gradient is defined, by rewinding (using) the tape. The rest of this section will consist of first an example to introduce automatic differentiation followed by a more general presentation. This section presents a mathematical description, while the implementation is briefly presented in section 9.2.2.

8.1.1 Example

Calculate the value and gradient of the function,

$$h(x,y) = 4x + 6\sin(y). \tag{134}$$

The first step is to separate the different operators into intermediate variables, showed in table 2. The next step is to calculate the derivatives of the intermediate variable in table 2, where the results are presented in table 3. From table 3, it is possible to see that the *chain rule* can be used, where all the necessary partial derivative are

[4] The machine precession will of course restrict the exactness.

[5] Automatic differentiation is also known as *algorithmic differentiation* or *computational differentiation.*

[6] Neidinger does not explicitly state that it is the forward mode approach.

Intermediate variables	Expressions in intermediate variables	Expression in independent variables
u_1	x	x
u_2	y	y
u_3	$4u_1$	$4x$
u_4	$\sin(u_2)$	$\sin(y)$
u_5	$6u_4$	$6\sin(y)$
u_6	$u_3 + u_5$	$4x + 6\sin(y)$

Table 2: The values of the intermediate variables and the respective expression in independent variables for the example.

Derivative intermediate variables
$\frac{\partial u_1}{\partial x} = 1$
$\frac{\partial u_2}{\partial y} = 1$
$\frac{\partial u_3}{\partial u_1} = 4$
$\frac{\partial u_4}{\partial u_2} = \cos(u_2)$
$\frac{\partial u_5}{\partial u_4} = 6$
$\frac{\partial u_6}{\partial u_3} = 1$
$\frac{\partial u_6}{\partial u_5} = 1$

Table 3: The partial derivative for the intermediate variables in the example.

calculated,

$$\frac{\partial h}{\partial x} = \frac{\partial u_6}{\partial x} = \frac{\partial u_6}{\partial u_3}\frac{\partial u_3}{\partial u_1}\frac{\partial u_1}{\partial x} = 1 \cdot 4 \cdot 1 = 4, \tag{135}$$

which oblivious are accurate and analogously

$$\frac{\partial h}{\partial y} = \frac{\partial u_6}{\partial y} = \frac{\partial u_6}{\partial u_5}\frac{\partial u_5}{\partial u_4}\frac{\partial u_4}{\partial u_2}\frac{\partial u_2}{\partial y} = 1 \cdot 6 \cdot \cos(u_2) \cdot 1 = \{u_2 = y\} = 6\cos(y), \tag{136}$$

which are also is correct. A more systematic approach, instead of the tables, is to use a matrix L. The matrix for this example is present in table 4, the interpretation of the matrix is that each row has a variable and the columns correspond to the partial derivatives of the other variables, except the diagonal elements. For example the 4 in row three and column one is the partial derivative, $\frac{\partial u_3}{\partial u_1} = 4$. (Forth et al., 2004, p. 270) derive the negative diagonal. It is possible to "loop" through the matrix and find all the derivatives such that solvable equation systems can be constructed.

	u_1	u_2	u_3	u_4	u_5	u_6
u_1	-1					
u_2	0	-1				
u_3	4	0	-1			
u_4	0	$cos(u_2)$	0	-1		
u_5	0	0	0	6	-1	
u_6	0	0	1	0	1	-1

Table 4: The matrix L, i.e. the tape for the example. Blank spaces are to be seen as zeros.

The equation system can be constructed so that different derivatives are calculated. Neidinger's approach (2010, pp. 558-561) is preferable if all partial derivatives of the type,

$$\bar{u}_i = \frac{\partial u_6}{\partial u_i}, \ \forall i = 1, \ldots, 6, \tag{137}$$

are of interest. These derivatives are received if a equation system are solved for $\bar{u}$, and the equation system is

$$\bar{u}^T L = \mathbf{b}^T \tag{138}$$

where $\bar{u}^T = [\bar{u}_1, \bar{u}_2, \dots, \bar{u}_6]$ and $\mathbf{b}^T = [0, 0, \dots, 0, -1]$, where

$$\bar{u}_i = \frac{\partial u_6}{\partial u_i} \forall i = 1, \dots, 6. \tag{139}$$

The size of $\mathbf{b}$ corresponds to the number of rows (columns) of L, where the last element is -1 and the rest are zero. It is also possible to calculate the derivative, e.g. $\frac{\partial u_j}{\partial u_i}, \forall i = 1, \dots, j$ as well. The tape is then cut, so that all rows and columns after row j and column j are removed and then the same algorithm is applied.

It can be of interest to find all partial derivatives for all the intermediate variables with respect to the *independent variables.* The approach used in (Forth et al., 2004, pp. 268-271) fits this purpose and will be referred to as *Forth's approach.* The equation system to solve, for D, is

$$LD = -P, \tag{140}$$

where L is the same tape as before. The right hand side,

$$P = \begin{bmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & -1 \\ 0 & 0 & 0 \\ \vdots & \vdots & \vdots \\ 0 & 0 & 0 \end{bmatrix}, \tag{141}$$

where the number of columns is the number of *independent variables* and the number of rows match L. The diagonal of the matrix should be -1, described in (Forth et al., 2004, p.268-271) and D has the form

$$D = \begin{bmatrix} \nabla u_1^T \\ \nabla u_2^T \\ \vdots \\ \nabla u_6^T \end{bmatrix}. \tag{142}$$

8.1.2 Forward Sweep - General

We have now seen an easy example of the automatic differentiation. In this section a more general presentation of the forward sweep is presented. Let the function be of the type $f : \mathbb{R}^n \mapsto \mathbb{R}$, i.e. that we have n *independent variables* and one *dependent variable* and a unknown number of *intermediate variables.* The first n intermediate variables are independent variables, $u_i = x_i, \ \forall i = 1, \dots, n$. Then the intermediate variables are defined as $u_i = \Phi_i(\{u\}_{j \prec i}), i > n$[7], i.e. that an intermediate variable can depend on previous calculated intermediate variables but can not depend on future intermediate variables.

Then let a $u_m = \phi(u_k, u_l)$ where $k < m$ and $l < m$, and with a current tape L^{m-1} it is then possible to calculate the new tape matrix as,

$$L^m = \left(\begin{array}{cccccc} & & L^{m-1} & & & \mathbf{0}_{m-1} \\ \mathbf{0}_{k-1}^T & \frac{\partial \phi(u_k, u_l)}{\partial u_k} & \mathbf{0}_{l-k-1}^T & \frac{\partial \phi(u_k, u_l)}{\partial u_l} & \mathbf{0}_{m-1-k-l}^T & -1 \end{array} \right), \tag{143}$$

the function ϕ can depend on a arbitrary number of intermediate variables, from one to $m-1$ the choice of two is just chosen for visibility. This process is then repeated for all the operations until the function f is calculated. The value of the function of the is also calculated as usual. To handle a more general function, $f : \mathbb{R}^n \mapsto \mathbb{R}^k, k > 1$, the forward sweep is unchanged but the reverse sweep must be changed.

[7] The notation is chosen to match the notation in (Forth et al., 2004)

8.1.3 Reverse Sweep - General

Depending if $k > 1$ or $k = 1$ in $f : \mathbb{R}^n \mapsto \mathbb{R}^k$ the tape must be re-winded with different methods. We start with the first case, $k = 1$. Neidinger's approach needs a right hand side $\mathbf{b}^T$, which has the size $1 \times n_c$ where n_c is the number of columns in the matrix L, all elements is zero except the last which is -1, and it is possible to solve the equation system, (138).

For Forth's approach, the matrix P must be constructed. The number of columns equals the number of *independent variables* and the number of rows equals the number of rows in L. The value -1 is then assigned to the diagonal and the equation system, (140), is solvable.

For functions with $k > 1$ the tape can be solved by cutting the tape and repeating the rewinding. One important observation is that the different dependent variables are different rows in the matrix L. This means that we need to keep track of the rows where different dependent variables are stored. With Forth's approach nothing is changed for the equation system except that more than one element in D is of interest.

Neidinger's approach requires more modification. In the example above, a method to find other derivatives for intermediate variables, was described. Let the tape matrix, L, have the dimension, $n \times n$ and that the intermediate variable u_k is a dependent variable, which corresponds to row k in the tape L. To find the derivative of this variable, the tape must be cut. The cut matrix, L^k, is the first k rows and columns of L. The regular algorithm is then applied to the cut tape, L^k. This procedure is then repeated for all interesting intermediate variables.

8.2 Forward Mode

The forward mode consists of only one sweep, a *forward sweep*, which is different from the reverse mode's forward sweep. The forward mode is constructed around a pair, value and derivative, of values and not a *tape*. Function operate parallel on the pair, which give that the value and the derivative are computed in parallel.

Let x be an independent variable and noted as a pair $x = \{x_v, x_d\} = \{x_0, 1\}$. Also, let $g : \mathbb{R} \mapsto \mathbb{R}$ and we can now evaluate the function in x and get $g(x) = \{g(x_v), g'(x_v)x_d\}$. Additionally, let $y = \{y_v, y_d\}$ and $z = \{z_v, z_d\}$ and that both are dependent on the same independent variable. Evaluate a two argument function, $h(w_1, w_2)$, in the point (y, z), $h(y, z) = \{h(y_v, z_v), \left.\frac{\partial h}{\partial w_1}\right|_{w_1=y_v} \cdot z_d + \left.\frac{\partial h}{\partial w_2}\right|_{w_2=z_v} \cdot y_d\}$.

Example 1. *Let $x = \{3, 1\}$ and evaluate the function $f \equiv f(x) = \sin(x) \cdot e^x$. First are some intermediate variables computed, $y = \sin(x) = \{\sin(3), \cos(3)\cdot 1\}$ and $z = e^x = \{e^3, e^3 \cdot 1\}$. Finally $h = y\cdot z = \{\sin(3)e^3, \cos(3)e^3 + \sin(3)e^3\}$. Alternative we use derivate the function $f'(x) = \cos(x)e^x + \sin(x)e^x|_{x=3} = \cos(3)e^3 + \sin(3)e^3$.*

An advantage of the forward method is that all the intermediate variables do not need to be saved after they have been used, and therefore the memory requirement is lower than the reverse mode.

8.3 Combination of Forward and Reverse Mode

The reverse mode is memory expensive, by combining the forward mode and reverse mode, the memory requirement can be lowered. Furthermore the cost of rewinding the tape can also be substantially lowered by combining the two modes. The reverse mode is the general mode in the thesis and this will be complemented by the forward mode. The forward mode will be used when the functions have the form $f : \mathbb{R} \mapsto \mathbb{R}^k$. The *forward dependent variables* are made intermediate variables in the reverse mode and then written to the tape. The tape is extended with one row (and one column), which only contains the intermediate variable and its derivative. All operations regarding the new intermediate variable are summarized in just one row. Thus, the combined method is competitive both with respect to speed and memory requirement

9 Implementation

The mathematical concepts and derivations are not trivial to realize as a computer algorithm. Therefore, this chapter is dedicated to explain how some of these theoretical concepts can be implemented in practice. The following section will describe the construction of the grid, the implementation of the automatic differentiation and the selection of options. When code commands are referred, `teletypefont` will be used.

9.1 Grid Construction

There exists a time and strike spectrum for a set of options, $\mathcal{O}$. The options strike price creates the set $\mathcal{K}_{\mathcal{O}}$ and the maturity times create $\mathcal{T}_{\mathcal{O}}$. There are also variance spanning sets, $\mathcal{K}_c$ and $\mathcal{T}_T$ for the central spanning strike levels and maturity spanning times respectively. It is sought after that both $\mathcal{K}_c \cap \mathcal{K}_{\mathcal{O}} = \emptyset$ and $\mathcal{T}_T \cap \mathcal{T}_{\mathcal{O}} = \emptyset$. If the intersect is non-empty the mutual elements are removed from $\mathcal{K}_{\mathcal{O}}$, $\mathcal{T}_{\mathcal{O}}$. $\mathcal{K}_{\mathcal{O}}$ and $\mathcal{T}_{\mathcal{O}}$. Furthermore, these must have certain properties, which are presented below.

For convenience, the strikes are fixed to an equidistant logarithmic grid, for all $t \in \mathcal{J}$. The difference between the levels are Δ_c and the index price at the reference time is S_c. From these we can define the central spanning strike levels

$$\mathcal{K}_c = \left\{S_c e^{-l\Delta_c},\ \ldots,\ S_c e^{-2\Delta_c},\ S_c e^{-\Delta_c},\ S_c,\ S_c e^{\Delta_c},\ S_c e^{2\Delta_c},\ \ldots,\ S_c e^{u\Delta_c}\right\}, \tag{144}$$

where $S_u \equiv S_c e^{u\Delta_c}$ and $S_l \equiv S_c e^{-l\Delta_c}$ is the highest and lowest strike level respectively in the unextended grid. The grid construction can be divided into several parts, given the size of the unextended grid, Δ_c, the number of branches, moat construction and definition of $\mathcal{I}^{i,t}$, $\forall i \in \mathcal{I},\ t \in \mathcal{J}$. Furthermore, there are several limits that need to be decided without mathematical motivation. These limits will be tested in the implementation in order to find parameters that are both proper and still computable.

In order to avoid changing the grid in every optimization step, the expected return and volatility used to create the grid are chosen to be constant. These could for instance be chosen as an average of the historical volatility or as the average of the implied volatility of the quoted options. This volatility is also a reasonable (flat) starting guess for the local volatility surface.

9.1.1 Unextended Grid Size

The grid must span a strike spectrum greater than the option implied, $S_u > \max\left(\mathcal{K}_{\mathcal{O}}\right)$ and $S_l < \min\left(\mathcal{K}_{\mathcal{O}}\right)$ and sufficiently large such that the effect on accuracy is negligible. This can be achieved by ensuring that the probability of reaching the strike boundaries are small,

$$\mathrm{P}(S^{T_{n_T}} \geq S_u | S^{T_0} = S_c) < \epsilon_u \tag{145}$$

$$\mathrm{P}(S^{T_{n_T}} \leq S_l | S^{T_0} = S_c) < \epsilon_l, \tag{146}$$

where S is the underlying price process in discrete time. Hence,

$$\begin{gathered}\mathrm{P}\left(S^{T_{n_T}} > S_u | S^{T_0} = S_c\right) < \epsilon_u \Leftrightarrow \mathrm{P}\left(\frac{R-\mu}{\sigma} > \frac{\ln\left(\frac{S_u}{S_c}\right) - \mu}{\sigma}\right) < \epsilon_u \Leftrightarrow \\ 1 - \Phi\left(\frac{\ln\left(\frac{S_u}{S_c}\right) - \mu}{\sigma}\right) < \epsilon_u \Leftrightarrow \left\{S_u = S_c e^{\bar{u}\Delta_c}\right\} \Leftrightarrow \Phi^{-1}(1-\epsilon_u)\sigma + \mu < \bar{u}\Delta_c \Rightarrow \\ u = \left\lceil \frac{\Phi^{-1}\left(1-\epsilon_u\right)\sigma + \mu}{\Delta_c} \right\rceil\end{gathered} \tag{147}$$

and analogously

$$l = \left\lceil -\frac{\Phi^{-1}\left(\epsilon_l\right)\sigma + \mu}{\Delta_c} \right\rceil. \tag{148}$$

If equation (145) (equation (146)) is unsatisfied, S_u (S_l) must be adjusted to a greater (lower) value.

9.1.2 Resolution

The strike resolution is given by Δ_c, where a smaller Δ_c adds accuracy at the cost of increased computational complexity. A criterion is formed to restrict the amount of (transition) probability assigned to a single branch. From a given node there always exists an outcome of the price process where the price is unchanged and concurrently there exists $b_u^{i,t} + b_l^{i,t}$ branches to adjacent nodes, where $b_u^{i,t}$ and $b_l^{i,t}$ are the number of upward and downward branches, respectively. The set of events, A, are constructed as

$$A_k^{i,t} = \begin{cases} \left[b_l^{i,t}\Delta_c, b_l^{i,t}\Delta_c - \frac{\Delta_c}{2}\right] & k = -b_l^{i,t} \\ \left[\frac{\Delta_c}{2} - \Delta_c \cdot k, \frac{\Delta_c}{2} - \Delta_c \cdot (k-1)\right] & k = -(b_l^{i,t} - 1), \ldots, -1 \\ \left[-\frac{\Delta_c}{2}, \frac{\Delta_c}{2}\right] & k = 0 \\ \left[\frac{\Delta_c}{2} + \Delta_c \cdot (k-1), \frac{\Delta_c}{2} + \Delta_c \cdot k\right] & k = 1, \ldots, b_u^{i,t} \\ \left[b_u^{i,t}\Delta_c - \frac{\Delta_c}{2}, b_u^{i,t}\Delta_c\right] & k = b_u^{i,t}, \end{cases} \tag{149}$$

where $S_i^{T_t} e^{b_u^{i,t}\Delta_c} \leq S_u$, $S_l \leq S_i^{T_t} e^{-b_l^{i,t}\Delta_c}$ and $[c,d]$ denotes the interval between c and d. Additionally, to achieve a sufficient fragmentation of the probability the events must also fulfill

$$\max_{k \in \{-b_l^{i,t}, \ldots, 0, \ldots, b_u^{i,t}\}} P\left(r_{i,i+k}^t \in A_k^{i,t}\right) < \epsilon, \ \forall i \in \mathcal{I}, \ \forall t \in \mathcal{J}. \tag{150}$$

If any event $A_k^{i,t}$ violates (150), the difference Δ_c is to large. Furthermore, if Δ_c is set very small the condition (150) will hold, but if b_u and b_l are assigned with too low integers most of the sample space is not covered by the events, which affects the adequacy of the pricing. This is addressed as conditions in posterior paragraphs.

9.1.3 Choice of Branches

This section focuses on computational efficiency, central moment spanning and coverage of the sample space. We start with a definition, which is a special case of definition 8.

Definition 10. *The **lowest** and **greatest growth factor** attainable for node (S_i, T_t) is denoted as $r_{i,t}^{min}$ and $r_{i,t}^{max}$ respectively.*

Necessary, but not sufficient, intervals for $r_{i,t}^{min}$ and $r_{i,t}^{max}$ $\forall i \in \mathcal{I}, \ \forall t \in \mathcal{J}$ are derived below. The conditions are based on computational efficiency, central moments matching and coverage of probabilities.

9.1.3.1 Computational Efficiency

To include extreme returns with very low probabilities cost much computation power in relation to the increased accuracy. Let $0 \leq \epsilon_{min} \leq 1$ denote this limit. The normal distribution is utilized here, but the approach for other distributions is similar. The limits are given by

$$\begin{gathered} \epsilon_{min} < P\left(X \leq r_{i,t}^{min}\right) = P\left(\frac{X - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}} \leq \frac{r_{i,t}^{min} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}}\right) = \\ \left\{\frac{X - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}} \text{ is standard normal}\right\} = \Phi\left(\frac{r_{i,t}^{min} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}}\right) \Leftrightarrow \\ \frac{r_{i,t}^{min} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}} > \Phi^{-1}(\epsilon_{min}) \Leftrightarrow r_{i,t}^{min} > \sigma\sqrt{\Delta_{T_i}}\Phi^{-1}(\epsilon_{min}) + \mu_{i,t}\Delta_{T_i} \end{gathered} \tag{151}$$

and

$$\begin{gathered} \epsilon_{max} < P(X > r_{i,t}^{max}) = P\left(\frac{X - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}} > \frac{r_{i,t}^{max} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}}\right) = \\ \left\{\frac{X - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}} \text{ is standard normal}\right\} = 1 - \phi\left(\frac{r_{i,t}^{max} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}}\right) \Leftrightarrow \\ \epsilon_{max} < 1 - \Phi\left(\frac{r_{i,t}^{max} - \mu_{i,t}\Delta_{T_i}}{\sigma_{i,t}\sqrt{\Delta_{T_i}}}\right) \Leftrightarrow r_{i,t}^{max} < \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}\left(1 - \epsilon_{max}\right), \end{gathered} \tag{152}$$

which are summarized to

$$\begin{aligned} r_{i,t}^{min} &> \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}(\epsilon_{min}) \\ r_{i,t}^{max} &< \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}\left(1-\epsilon_{max}\right). \end{aligned} \tag{153}$$

Hence, branches that represent greater (lower) returns than r_{max} (r_{min}) will not add to the accuracy in relation to the computational cost.

9.1.3.2 Probability Coverage

The equations (151) and (152) can be used to derive conditions for the coverage of the sample space. By replacing ϵ with a corresponding ϵ', the inequalities can be reversed with an analogues derivation. Thus we can ensure that only a tiny fraction of the variance is not adequately addressed.

$$\begin{aligned} r_{i,t}^{min} &< \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}(\epsilon'_{min}) \\ r_{i,t}^{max} &> \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}\left(1-\epsilon'_{max}\right). \end{aligned} \tag{154}$$

9.1.3.3 Central Moment Spanning

The algorithm that assigns branches with probability uses central moment matching, see equation (95). The conditions below are necessary but not sufficient to guarantee that the matching can be done. We assume, that there exists a branch with zero growth, which is generally true. Furthermore, the only interesting range, where $\mu \geq 0$ is studied.

The approach is different for even and odd orders of central moments, c and is therefore divided. For an even c we study if the two most extreme outcomes are sufficient to reach the central spanning moments,

$$\begin{gathered} \max_{p\in[0,1]} \mathrm{E}[(X-\mathrm{E}[X])^c] = \max_{p}\,(1-p)(r_{i,min}^t - \mu\Delta_{T_i})^c + p(r_{i,max}^t - \mu\Delta_{T_i})^c \geq m_c \Rightarrow \\ \{\mu\Delta_{T_i} > 0 \Rightarrow p = 0\} \Rightarrow (r_{i,min}^t - \mu\Delta_{T_i})^c > m_c \Leftrightarrow \{m_c \geq 0\} \Leftrightarrow |(r_{i,min}^t - \mu\Delta_{T_i})| > (m_c)^{(1/c)} \Leftrightarrow \\ \left\{r_{i,min}^t < 0\right\} \Leftrightarrow r_{i,min}^t < \mu\Delta_{T_i} - (m_c)^{(1/c)}, \end{gathered} \tag{155}$$

where X is the stochastic returns for the time increment Δ_{T_i}. The first implication follows since c is even and $r_{i,min}^t < 0$, which also implies that $(r_{i,min}^t - \mu\Delta_{T_i})^c > (r_{i,min}^t - \mu\Delta_{T_i})^c$ and that all probability should be allocated to $(1-p)$ in the extreme case.

The odd central moments can in them self be divided into two cases, m_c negative and m_c positive,

$$\begin{cases} \max_{p\in[0,1]} \mathrm{E}[(X-\mathrm{E}[X])^c] = (r_{i,max}^t - \mu_{i,t}\Delta_{T_i})^c > m_c \Leftrightarrow r_{max} > \mu_{i,t}\Delta_{T_i} + (m_c)^{1/c} & \text{if } m_c > 0 \\ \min_{p\in[0,1]} \mathrm{E}[(X-\mathrm{E}[X])^c] = (r_{i,min}^t - \mu_{i,t}\Delta_{T_i})^c < m_c \Leftrightarrow r_{min} < \mu_{i,t}\Delta_{T_i} - (-m_c)^{1/c} & \text{if } m_c < 0. \end{cases} \tag{156}$$

For the normal distribution all odd central moments are zero and therefore we can summarize to,

$$\begin{cases} r_{i,max}^t > \mu_{i,t}\Delta_{T_i} \\ r_{i,min}^t < \mu_{i,t}\Delta_{T_i}. \end{cases} \tag{157}$$

In the implementation it is reasonable to assume that $\epsilon'_{min} < 0.5$ and $\epsilon'_{max} < 0.5$, since the adjustment is generally very close to zero. This gives that the constraints in (154) implies the constraints (157).

9.1.3.4 Number of branches

Two conditions for $r^t_{i,max}$ and $r^t_{i,min}$ have been derived and can be rewrite from equation (153), (154) and (155) to

$$\mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}(\epsilon_{max}) < r_{i,t}^{min} < \min\Big(\mu_{i,t}\Delta_{T_i} - (m_c)^{1/c}, \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}(\epsilon'_{min})\Big) \quad (158)$$

$$\mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}\left(1 - \epsilon'_{max}\right) < r^t_{i,max} < \mu_{i,t}\Delta_{T_i} + \sigma_{i,t}\sqrt{\Delta_{T_i}}\Phi^{-1}(1 - \epsilon_{max}). \quad (159)$$

The left (right) inequality for $r^t_{i,max}$ ($r^t_{i,min}$) can be broken since it just is a efficiency condition, if the other inequality is broken it becomes impossible to find probabilities that matches the central moments. From these it is possible to determine the number of branches that must be used for spanning a sufficient sample space. It is not assured by definition that $r^t_{i,min}$ and $r^t_{i,max}$ are prefect divisible with Δ_c so we get the formula

$$\begin{cases} b^t_{i,u} &= \min\left(\left\lceil \frac{r^{max}_{i,t}}{\Delta_c} \right\rceil, \left\lceil \frac{1}{\Delta_c} \ln\left(\frac{S_u}{S^t_i}\right)\right\rceil\right) \\ b^t_{i,l} &= \min\left(\left\lceil \frac{-r^{min}_{i,t}}{\Delta_c} \right\rceil, \left\lceil \frac{1}{\Delta_c} \ln\left(\frac{S^t_i}{S_l}\right)\right\rceil\right) \end{cases} \forall i \in \mathcal{I},\ \forall j \in \mathcal{J}. \quad (160)$$

9.1.4 Grid Moat Construction

The *moat* is constructed such that all probability outside the initial grid is addressed. This moat can be constructed in many ways. One solution is to use several strike levels in the extended grid and another is to only use one level sufficiently far away from the initial grid to be able to match the moments. Since the probability is accumulated in the grid, it is essential that the probability of entering the grid is very small.

Independent of the approach, the boarders of the initial grid needs to be within the extended grids, with no overlap. Thus, the necessary conditions,

$$\min \bar{\mathcal{M}}_K > \max_{i\in\mathcal{I}_G, t\in\mathcal{T}_G} S^t_i \cdot e^{r^{max}_{i,t}}, \quad (161)$$

$$\max \underline{\mathcal{M}}_K < \min_{i\in\mathcal{I}_G, t\in\mathcal{T}_G} S^t_i \cdot e^{r^{min}_{i,t}}, \quad (162)$$

where $\bar{\mathcal{M}}_K$ and $\underline{\mathcal{M}}_K$ is the upper and lower moat respectively. We can construct

$$\mathcal{K}_{\tilde{G}} = \mathcal{K}_c \cup \mathcal{K}_{\mathcal{O}} \cup \bar{\mathcal{M}}_K \cup \underline{\mathcal{M}}_K. \quad (163)$$

9.1.5 Time Discretization

The set $\mathcal{T}_{\tilde{G}}$ contains the time spectrum, $\mathcal{T}_{\mathcal{O}}$, which is a nice property, since all maturities can then be matched exactly and the payoffs do not need to be mapped to other points in time. Furthermore, the size of Δ_T does impact the price of put and call options since the risk-free rate, dividend yield and local volatility are time dependent. Since an accurate local volatility surfaced is desirable, a smaller Δ_T is better but more computationally complex. In the implementation, it is convenient with an equidistant time discretization which coincides with all quoted option maturities and thus Δ_T can be chosen to a fraction of a business day or approximately $\frac{1}{252}$ year. However, since the methodology does not require the time increments to be equidistant, it might be feasible to increase the increments, whilst making sure that the quoted option maturities are matched.

9.1.6 Child Nodes

This section describes how this index set is constructed. Firstly the set $I_{\tilde{G}} = \{0, \dots, |\mathcal{K}_{\tilde{G}}| - 1\}$ and $I_G = \{1, \dots, |\mathcal{K}_{\tilde{G}}| - 2\}$ and the index that corresponds to the set $\mathcal{K}_c \cup \bar{\mathcal{M}}_K \cup \underline{\mathcal{M}}_K$ and $\mathcal{K}_{\mathcal{O}}$ defines the central moment and option spanning set,

$$\mathcal{I}_c \subset \mathcal{I}_{\tilde{G}} \tag{164}$$

$$\mathcal{I}_o \subset \mathcal{I}_{\tilde{G}}, \tag{165}$$

respectively, where $\mathcal{I}_c \cap \mathcal{I}_o = \emptyset$ and $\mathcal{I}_c \cup \mathcal{I}_o = \mathcal{I}_{\tilde{G}}$. Furthermore, nodes should cover the strike spectrum between $r^t_{i,min}$ and $r^t_{i,max}$ $\forall i \in \mathcal{I},\ t \in \mathcal{J}$. We can now define

$$\begin{aligned}\mathcal{I}^{i,t}_c = &\left\{j \in \mathcal{I}_c : r^{min}_{i,t} \leq r^t_{i,j} \leq r^{max}_{i,t}\right\} \\ &\cup \left\{j \in \mathcal{I}_c : j = \underset{i}{\operatorname{argmin}} \left(r^j_{i,t} - r^{max}_{i,t}\right) \text{ s.t. } r^j_{i,t} - r^{max}_{i,t} \geq 0\right\} \\ &\cup \left\{i \in \mathcal{I}_c : i = \underset{i}{\operatorname{argmax}} \left(r^j_{i,t} - r^{min}_{i,t}\right) \text{ s.t. } r^j_{i,t} - r^{min}_{i,t} \leq 0\right\} \quad \forall t \in \mathcal{J},\ \forall i \in \mathcal{I}_G,\end{aligned} \tag{166}$$

From $\mathcal{I}^{i,t}_c$ we can derive $\mathcal{I}^{i,t}_o \subset \mathcal{I}_o$, as

$$\mathcal{I}^{i,t}_o = \left\{j \in \mathcal{I}_o : \min(\mathcal{I}^{j,t}_c) < j < \max(\mathcal{I}^{j,t}_c)\right\}, \tag{167}$$

and these can be combined to

$$\mathcal{I}^{i,t} = \begin{cases}\mathcal{I}^{i,t}_o \cup \mathcal{I}^{i,t}_c & \forall i \in \mathcal{I}_G \\ i & \forall i \notin \mathcal{I}_G,\end{cases} \tag{168}$$

where $\{i \notin \mathcal{I}_G\} = \bar{\mathcal{M}}_K \cup \underline{\mathcal{M}}_K$. From these sets $\mathcal{I}^{i,t}$ the outgoing branches for each node (S_i, T_t) are determined and the probabilities are assigned.

9.2 Implementation of Automatic Differentiation

The implementation is presented briefly and should be seen as a concept. The idea for both reverse and forward mode are very similar. Both implementations are based on *operator overloading* and *classes*. The operator overloading gives functionality to an operator for a new class that already has functionality for existing classes.

9.2.1 Forward Mode

The class `revAD` has two properties, `val` and `der`. The property `val` and `der` is the value and the derivative of the object respectively. The `der` is a row vector whose length equals the number of independent variables. The i-th element corresponds to the i-th independent variable and thus, the value of the i-th element is the object's derivative with respect to the i-th independent variable.

Example 2. *Let the number of independent variables be 3 and* $\mathtt{x} = \mathtt{forAD}(\mathtt{x}_0, [1,0,0])$ *and* $\mathtt{y} = \mathtt{forAD}(\mathtt{y}_0, [0,1,0])$ *and we can want to calculate* $z = x \cdot y|_{x=x_0, y=y_0} = x_0 y_0$. *The operator overloading for the multiplication,* `mtimes`*, is made as*

$$\mathtt{w} = \mathtt{u} * \mathtt{v} = \mathtt{forAD}(\mathtt{u.val} * \mathtt{v.val}, \mathtt{u.der} * \mathtt{v.val} + \mathtt{v.val} * \mathtt{u.der})$$

and we receive $\mathtt{z} = \mathtt{forAD}(\mathtt{x}_0 * \mathtt{y}_0, [\mathtt{y}_0, \mathtt{x}_0, 0])$*, i.e. the value is* $x_0 \cdot y_0$ *and the derivatives are* $\frac{\partial z}{\partial x}\big|_{x=x_0, y=y_0} = y_0$ *and* $\frac{\partial z}{\partial x}\big|_{x=x_0, y=y_0} = x_0$.

9.2.2 Reverse Mode

The class `revAD` has three properties, `val`, `id` and `tape`. The property `val` is the value of the object, `id` is the index for its *intermediate variable*, the corresponding row in the tape. The last property, `tape`, is not determined if it is a property to a specific instance of a global variable, which all instances have accesses to. The operator overloading handles the value in the same manner as for `forAD`. The handle of the derivative is different, we will use the same example as above but with reverse automatic differentiation.

Example 3. *Let the number of independent variables be 3 and* $\mathtt{x} = \mathtt{revAD}(\mathtt{x_0}, 1)$ *and* $\mathtt{y} = \mathtt{revAD}(\mathtt{y_0}, 2)$ *and we can want to calculate* $z = x \cdot y|_{x=x_0, y=y_0} = x_0 y_0$. *The initial tape is a diagonal* 3×3 *matrix whose elements are* (-1). *The operator overloading for the multiplication,* `mtimes`*, is made in two steps. Firstly a new variable is created as*

$$\mathtt{w} = \mathtt{u} * \mathtt{v} = \mathtt{revAD}(\mathtt{u.val} * \mathtt{v.val}, \mathtt{size}(\mathtt{tape}, 1) + 1)$$

and the `tape` *is extended with one new last row and column. The last row in the extended tape is than assigned with* `v.val` *in position* `u.id` *and* `u.val` *in position* `v.id` *and a* (-1) *in the last position. The value is directly attainable from* `z.val` *and the derivatives can be found by solving a equation system* (138) *or* (140).

9.3 Option Selection

A dense time and strike spectrum is also essential for extracting an adequate surface. An example of a proper underlying is the S&P500. Since there should not exist any arbitrage the put call parity must hold and thus it is equivalent to use put and call options. The correct price will be in the bid-ask interval, which will be approximated with the average (mid) price. It is inadequate to use last prices, since these are generally asynchronous.

10 Evaluation

This chapter contains a short description of how the evaluation will be performed. The evaluation consists of two parts. *Firstly*, the evaluation of the optimization algorithm itself, with respect to speed and numerical stability. *Secondly*, the evaluation of the surfaces that are extracted from the algorithm.

Comparison of the values of the objective function between iterations is one possible method that can be used for evaluation. Since the problem is a minimization problem the objective functions magnitude should be monotonically decreasing, which should be verified. Another evaluation of the algorithm is the speed, where for instance the average iteration time or the total extraction time is measured. A further evaluation could be to examine how the surface evolves for each iteration.

The extraction was made with respect to pricing error and smoothness, thus these are two essential properties to evaluate. One first, naive, evaluation is to construct a surface plot and visually inspect for abnormalities, such as roughness. The pricing error can be found by comparison between the market quoted prices and prices implied by the local volatility surface. These pricing errors can be presented in a matrix or in a surface. Furthermore, the pricing errors will be investigated in specific areas, and if the penalties can be adjusted to address this occurrence.

A further step is to price options, which are not used in the optimization, which are denoted as *out of sample* options. A common problem is *over-fitting*, where the local volatility surface is adapted to to match the options *in sample* too much and therefore an unrealistic surface might be extracted.

This can be investigated by pricing options *out of sample* and determine if the errors are distinguished from the in pricing errors in sample. This is a very important evaluation, since a future application of the surfaces is to price new options, which might not exist in the market.

The evaluation and implementation is somewhat an iterative process. After an evaluation it is important to check if the optimization problem can be refined for better performance. After these alterations it is essential to verify that these changes actually improved the model.

11 Results

The implementation has resulted in numerous different volatility surfaces and results, where the most relevant will be presented, and then discussed in the next chapter.

When applicable, the parameters used for the extraction are presented along with the results. The selection of results is done such that for instance the dynamics of the optimization problem is shown as well as the pricing accuracy and surface roughness.

11.1 Data series for Evaluation

In the remainder of the thesis the following four data series will be central for the evaluation and discussion in the next chapter.

The "long" data series used in the results was from 2015-04-24, and is the 6th series from that day. The number of options in the series are 444, with strikes between 1600 and 2500 at maturities up to 236 working days.

The "reference" series is from 2015-05-07, and is the third series from that day. The number of options in that series are 248, with strikes between 2000 and 2300 at maturities up to 77 working days.

The "reference for convergence" series is from 2015-04-24, with series number 3. The number of options in that series are 221, with strikes between 2000 and 2300 at maturities up to 86 working days.

The "subsequent" series is from 2015-05-06 to 2014-05-08, with series numbers 2,3 and 1 respectively. The set of options is constant over these days and the intermediate day coincides with the reference case. However, the time to maturity naturally varies by a day compared to the reference.

11.2 General Appearance

The local volatility surface for a circa 1 year long time series is presented in figure 3, where the parameters for the extraction are shown in table 5. The time series for this result are from 2015-04-24, where series 6 was used to extract the surface and series 7, 8 and 9 were used for out of sample pricing. The absolute pricing errors (in USD) are presented in figure 4 and the absolute pricing errors (in implied volatility) are presented in figure 5.

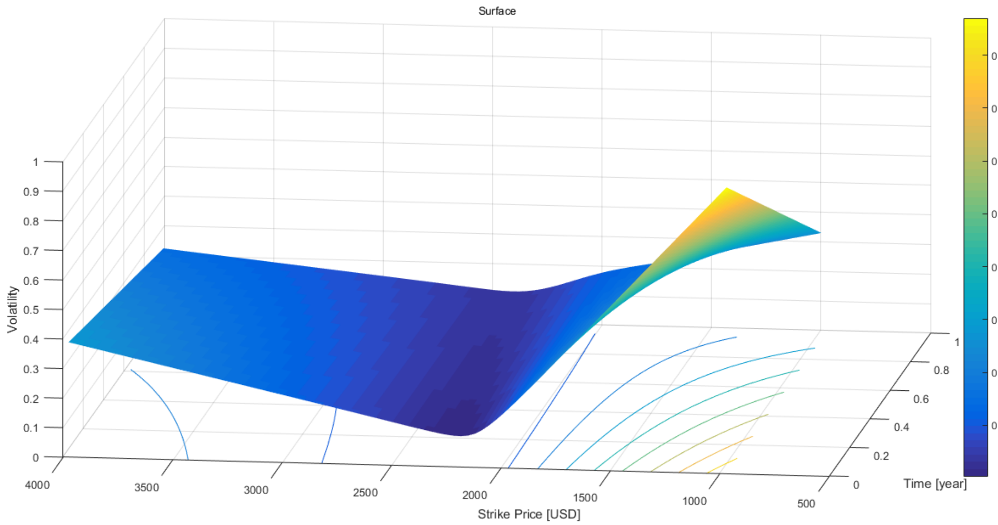


Figure 3: The general appearance of an extracted LVS.

Parameter	a^T	a^K	a^{TT}	a^{KK}	a^{KT}	E	F
Value	$1 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$5 \cdot 10^{0}$	$3 \cdot 10^{+7}$	$1 \cdot 10^{-12}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Table 5: The extraction parameters.

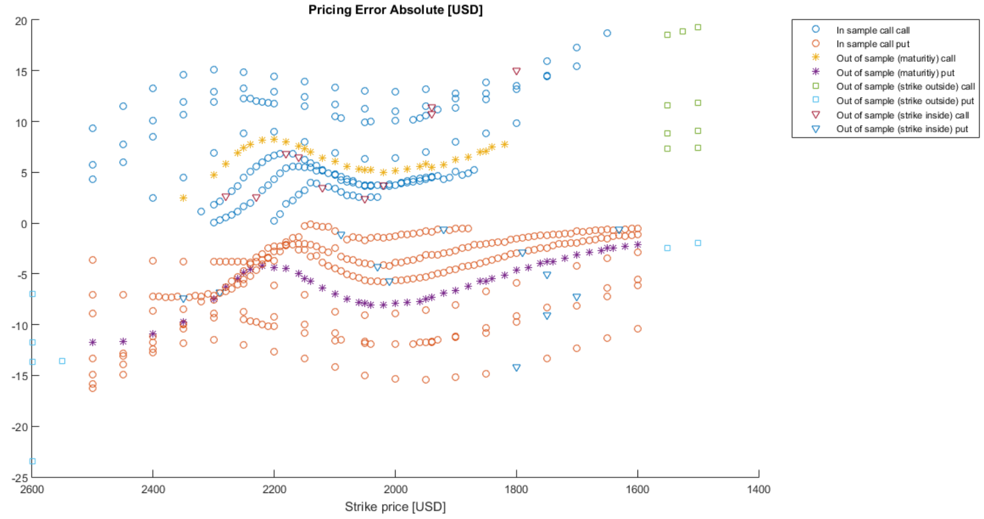


Figure 4: The values for the pricing errors in [USD]

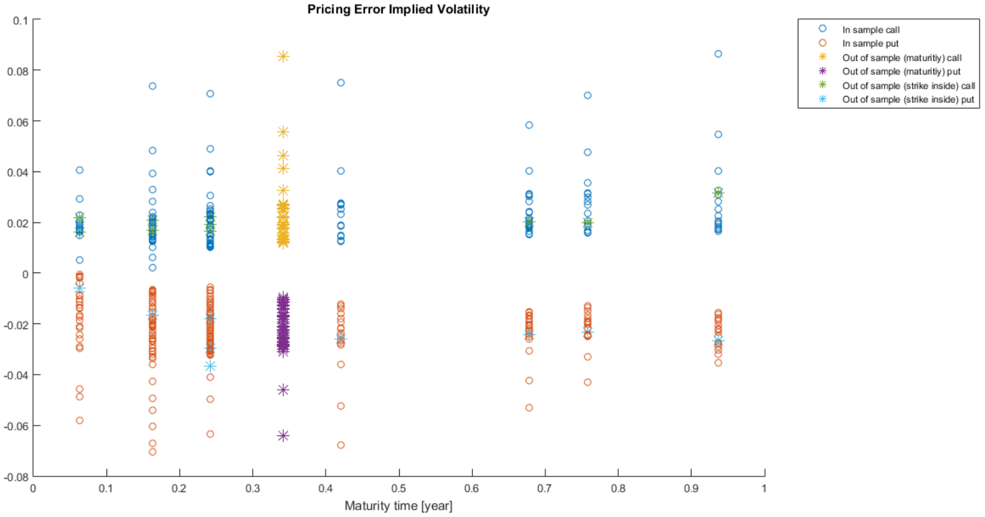
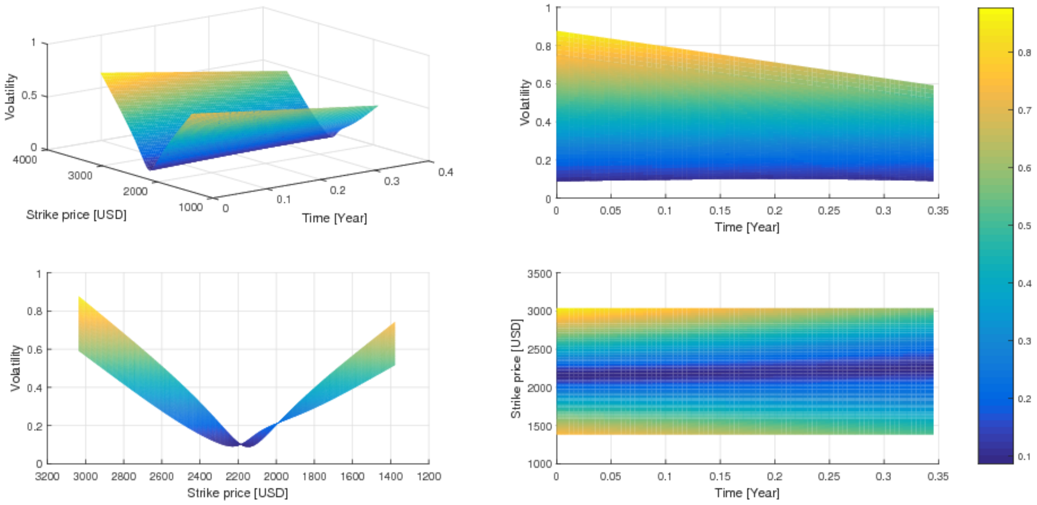


Figure 5: The values for the pricing errors in implied volatility

11.3 Reference Case

In order to draw conclusions from the results in different aspects, a reference case is extracted with corresponding parameters. The surface will be central in many of the following properties tests. The surface of the reference case and its parameters are presented in figure 6.



Parameter	a^T	a^K	a^{TT}	a^{KK}	a^{KT}	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{0}$	$5 \cdot 10^{+5}$	$1 \cdot 10^{0}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Figure 6: The solution surface for the reference starting volatility (25%).

11.4 Convergence

Different initial flat volatilities, showed in table 6, were used to extract the local volatility surface. The solutions for these volatilities are presented in figure 6 and 7. Figure 6 shows the reference surface with the solution for the starting volatility 25 % and figure 7 shows the absolute and relative difference for the solutions for the starting volatilities 10% and 65% against the reference solution.

Starting Value (%)
10
25
65

Table 6: Starting volatilities for the convergence test.

The for the three cases constant parameters are presented in table 6. Figure 8 shows the norm of the gradient for the function L, in the optimization problem (16).

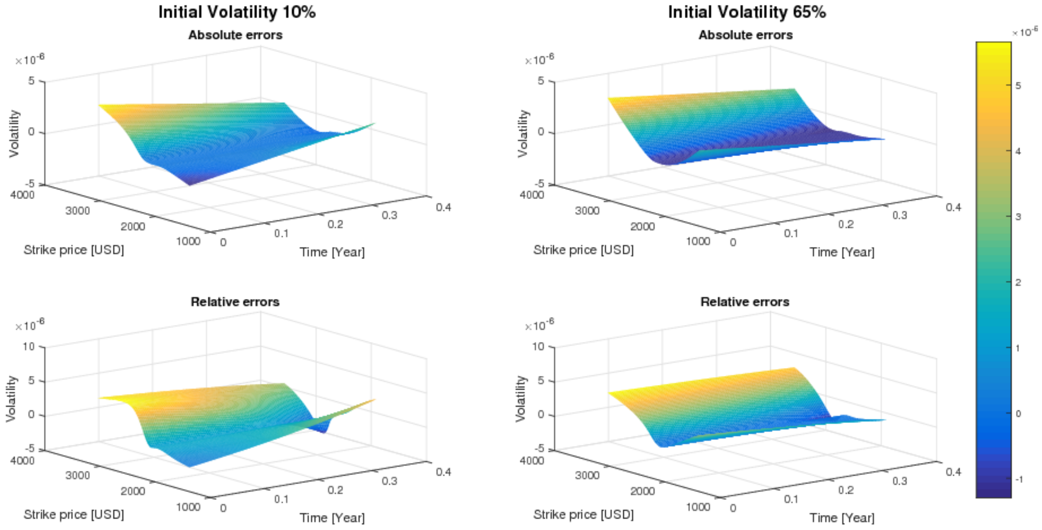


Figure 7: Absolute and relative errors for 10 % and 65% start volatility compared to the reference at 25%.

The left (right) column had 10% (65 %) as starting volatility. The first (second) row is the absolute (relative) errors with respect to the reference surface present in figure 6.

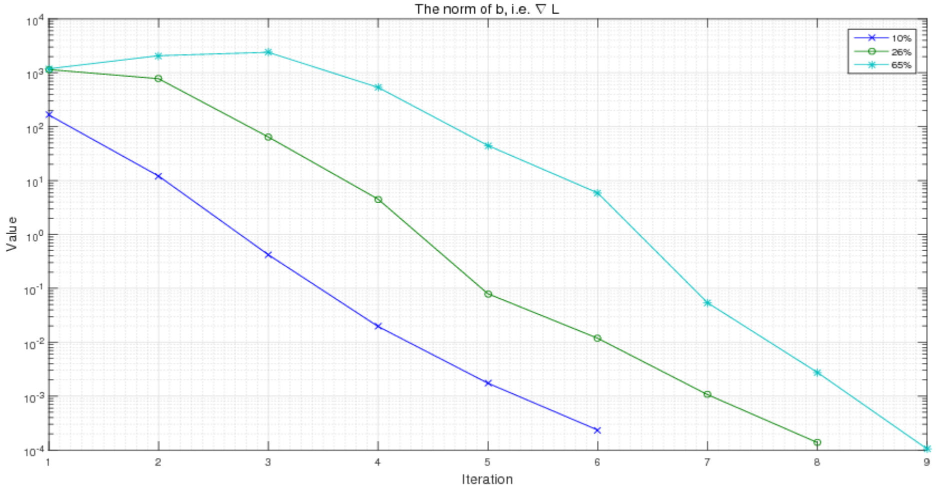
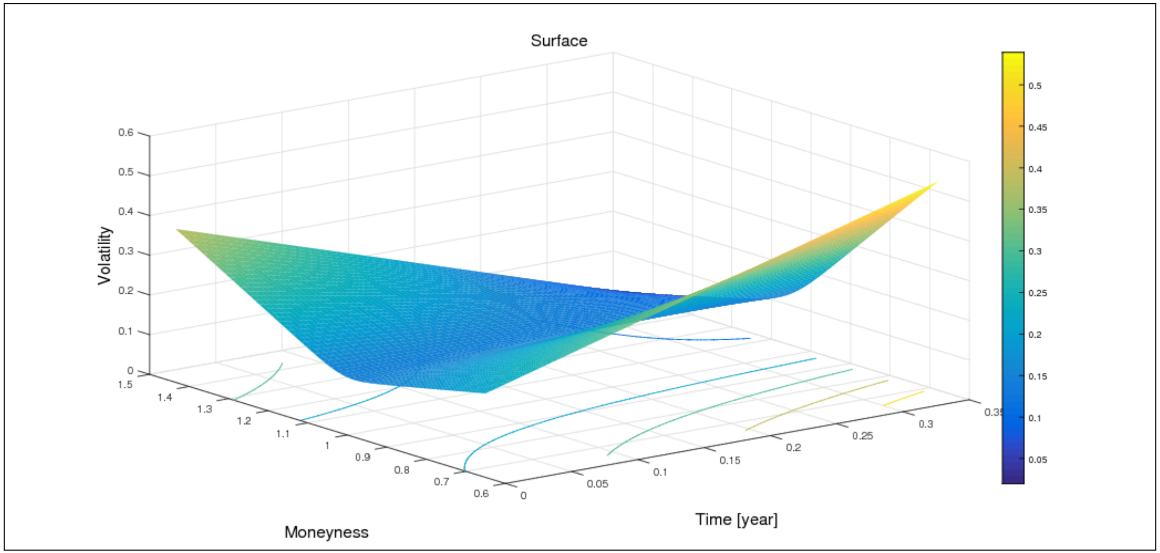


Figure 8: The rate of convergence for the norm of ∇L.

The value of the norm of ∇L is presented in the figure for each iteration, where the first value is after one iteration.

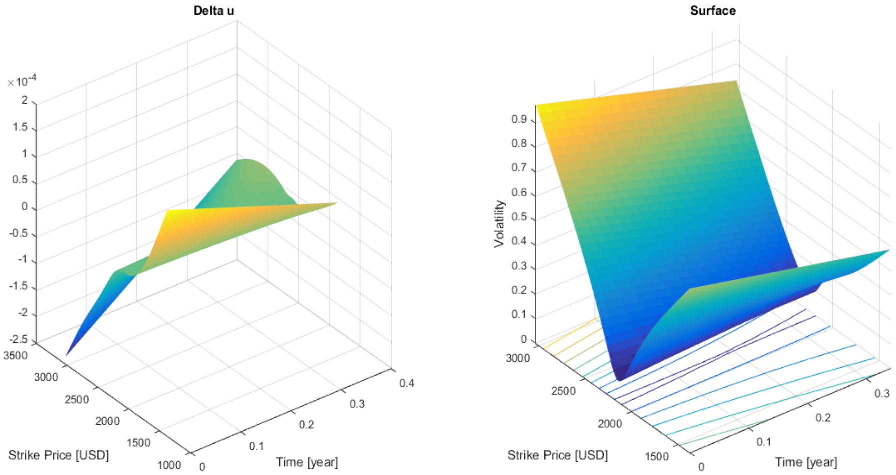
11.5 Different Parameters

In this section the reference set's parameters are higher penalized one by one. The resulting local volatility extraction is presented along with the parameters.



Parameter	$\boldsymbol{a^T}$	$\boldsymbol{a^K}$	$\boldsymbol{a^{TT}}$	$\boldsymbol{a^{KK}}$	$\boldsymbol{a^{KT}}$	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{0}$	$3 \cdot 10^{+7}$	$1 \cdot 10^{0}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Figure 9: Parameters, with higher penalized second order of derivate w.r.t. strike.

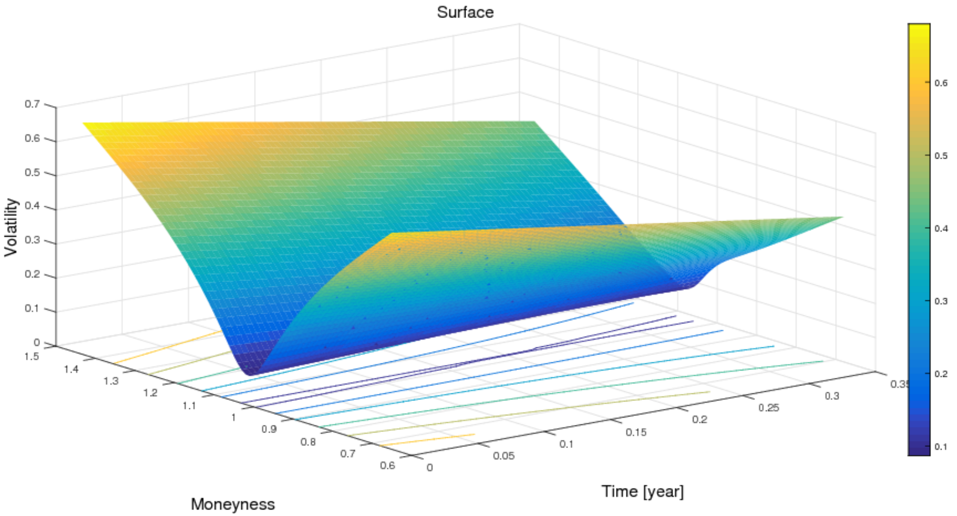


Parameter	$\boldsymbol{a^T}$	$\boldsymbol{a^K}$	$\boldsymbol{a^{TT}}$	$\boldsymbol{a^{KK}}$	$\boldsymbol{a^{KT}}$	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{2}$	$3 \cdot 10^{+5}$	$1 \cdot 10^{0}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Figure 10: Parameters, with higher penalized second order of derivative w.r.t. time. The algorithm did not converge and the step direction is showed to the left.

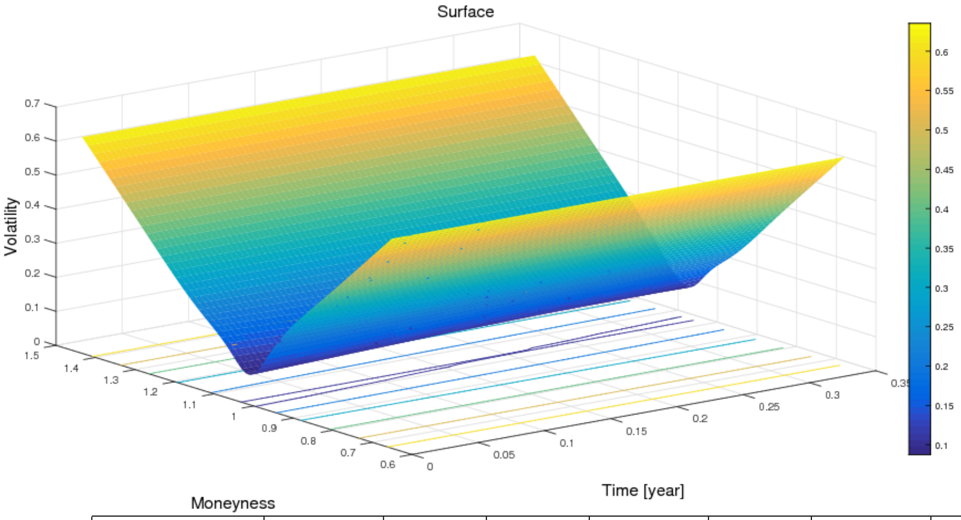
The second derivative for strikes is the most important for the general shape of the volatility surface. The other penalties do no effect 150424 series 3.

The roughness penalties $\boldsymbol{a^T}$, $\boldsymbol{a^K}$, $\boldsymbol{a^{TT}}$, $\boldsymbol{a^{KK}}$ and $\boldsymbol{a^{TK}}$ and pricing error E, have been varied. To increase comparability, the reference case, 7, will be used. Furthermore, the extraction results are presented in figure 6, 7, 9, 10, 11, 12 and 13.



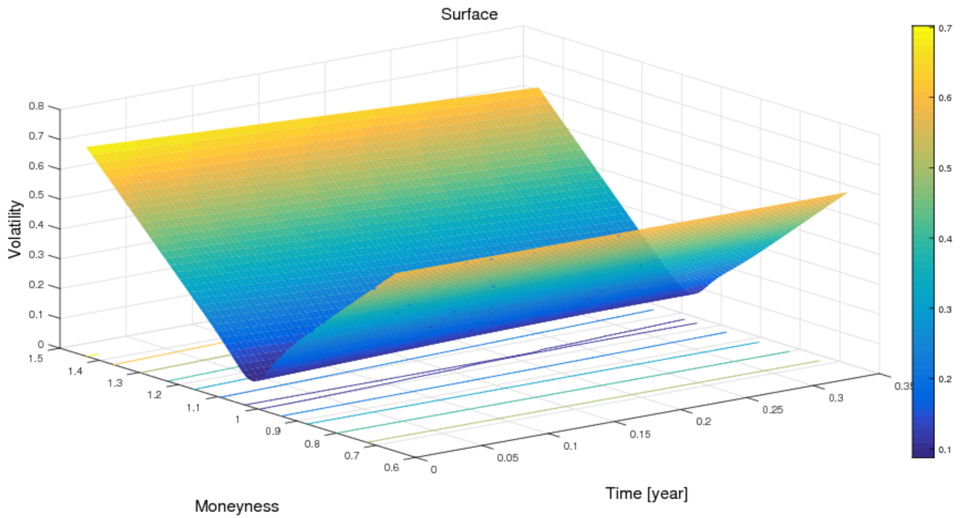
Parameter	$\boldsymbol{a^T}$	$\boldsymbol{a^K}$	$\boldsymbol{a^{TT}}$	$\boldsymbol{a^{KK}}$	$\boldsymbol{a^{KT}}$	E	F
Value	$1 \cdot 10^0$	$1 \cdot 10^{-3}$	$5 \cdot 10^0$	$3 \cdot 10^{+5}$	$1 \cdot 10^0$	$1 \cdot 10^{-2}$	$1 \cdot 10^0$

Figure 11: Parameters, with higher penalized first order of derivate w.r.t. strike.



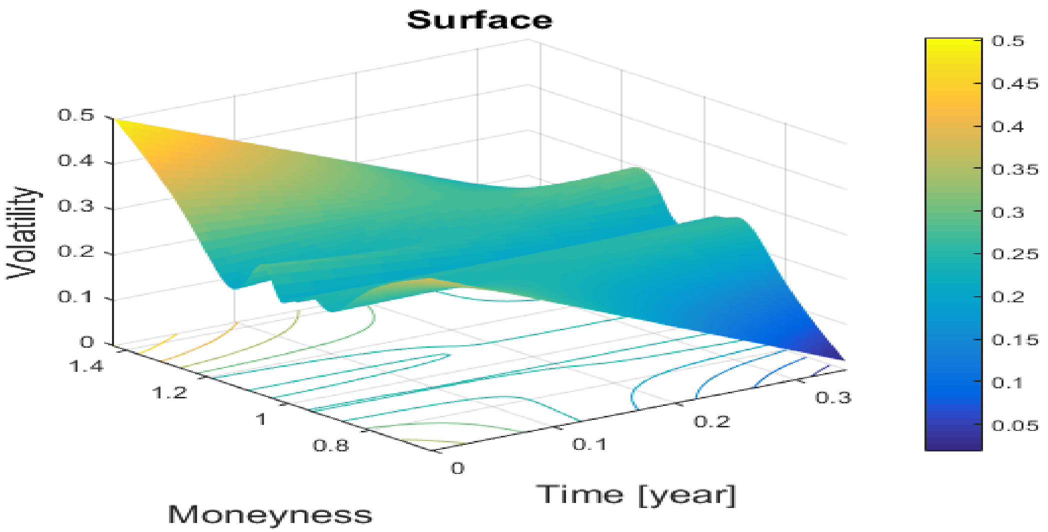
Parameter	$\boldsymbol{a^T}$	$\boldsymbol{a^K}$	$\boldsymbol{a^{TT}}$	$\boldsymbol{a^{KK}}$	$\boldsymbol{a^{KT}}$	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^0$	$5 \cdot 10^0$	$3 \cdot 10^{+5}$	$1 \cdot 10^0$	$1 \cdot 10^{-2}$	$1 \cdot 10^0$

Figure 12: Parameters, with higher penalized first order of derivate w.r.t. time.



Parameter	a^T	a^K	a^{TT}	a^{KK}	a^{KT}	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{0}$	$3 \cdot 10^{+5}$	$1 \cdot 10^{+2}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Figure 13: Parameters, with higher penalized second order of derivate w.r.t. the mixed second derivative.



Parameter	a^T	a^K	a^{TT}	a^{KK}	a^{KT}	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{0}$	$3 \cdot 10^{+5}$	$1 \cdot 10^{0}$	$1 \cdot 1$	$1 \cdot 10^{0}$

Figure 14: Parameters, with higher penalized pricing error

11.5.1 Interest Rate and Dividend Yield Effects

The appearance of the surface does change if the risk free rate and dividend yields are changed. However, the general curvature remains almost the same. Since data was not available, the interest rate and dividend yield were constant in the extractions. The surface from these are presented in the figures 15, 16 and 17. The absolute errors for these three cases are presented in figure 18, 19 and 20. The parameters where the same as the reference case above but here the risk free rate and the dividend yield was non zero, table 7 gives the information. Two options are removed from the absolute error plot because they had a extreme errors (compared) to the others.

Case	risk-free interest (%)	dividend yield (%)
1	1	0
2	0	2.5
3	1	2.5

Table 7: Interest rate and dividend test cases

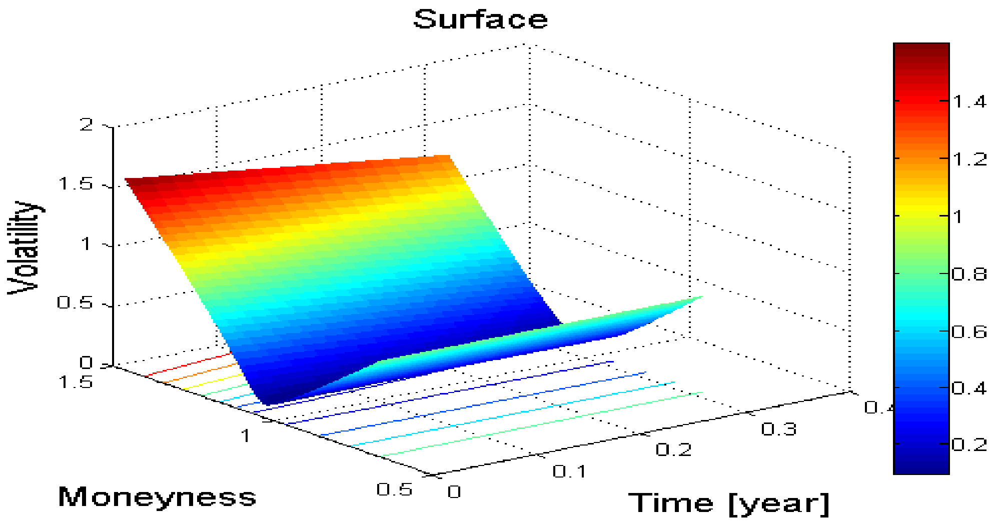


Figure 15: Surface with a risk free interest rate.

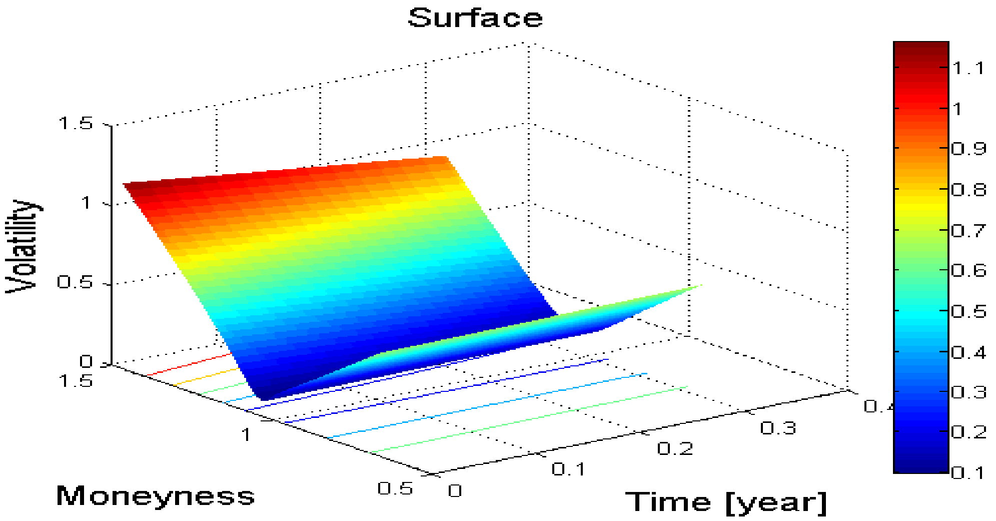


Figure 16: Surface with dividend yield.

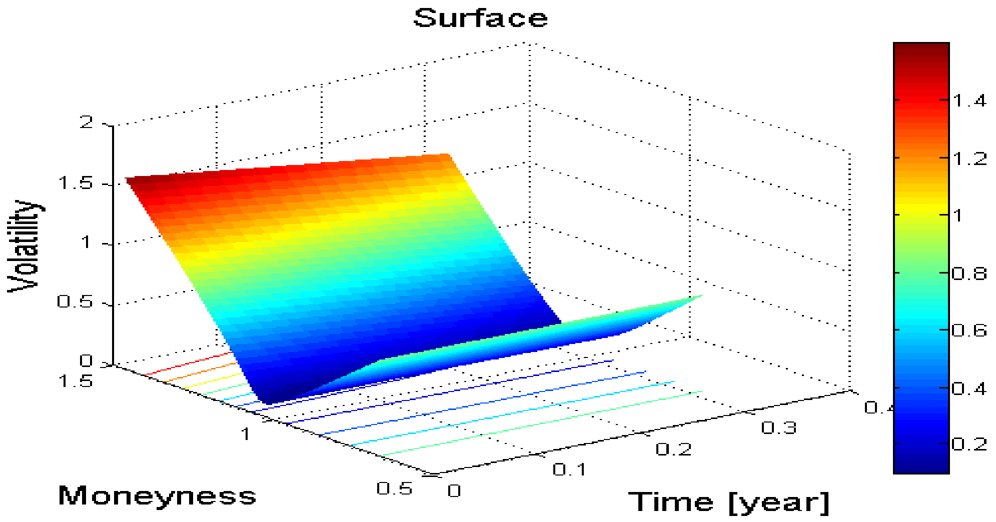


Figure 17: Surface with both the risk free rate and dividend yield.

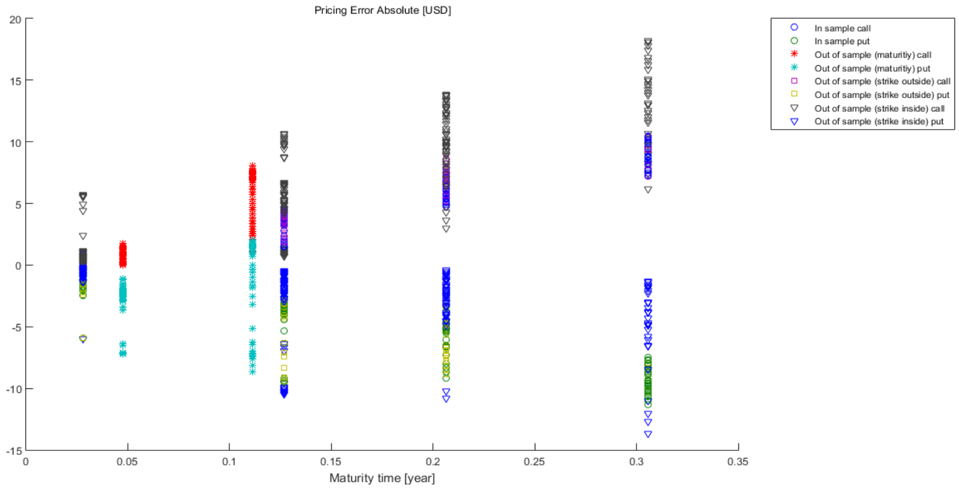


Figure 18: Absolute errors for the risk free interest rate modeled.

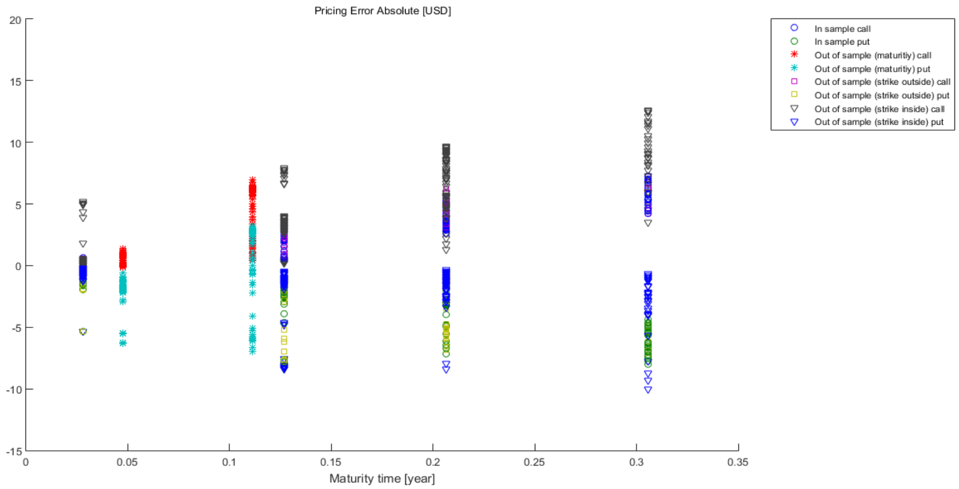


Figure 19: Absolute errors for the dividend yield modeled.

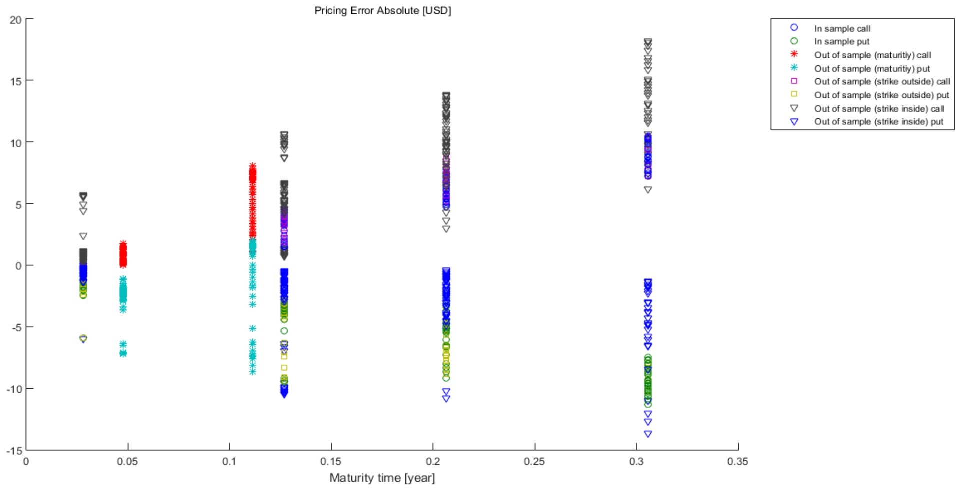


Figure 20: Absolute errors for the risk free interest rate and dividend yield modeled.

11.6 Out of Sample Option Pricing

In order to test the consistency of the pricing with the market quoted prices, out-of-sample options not used in the extraction of the surface were priced, for the reference case. Parameters that yielded adequate surfaces was used to extract a surface. The test is done to ensure that the model does not overfit the surface to the options in-sample. It also investigates how adequate the method can price options in between quoted maturities, which can have an exotic payoff. If the method's surface is overadapted then the in-sample pricing errors should be significantly lower than the errors for the out of sample options. The in-sample options are represented by circles in the figures and the out-of-sample options are represented by other symbols.

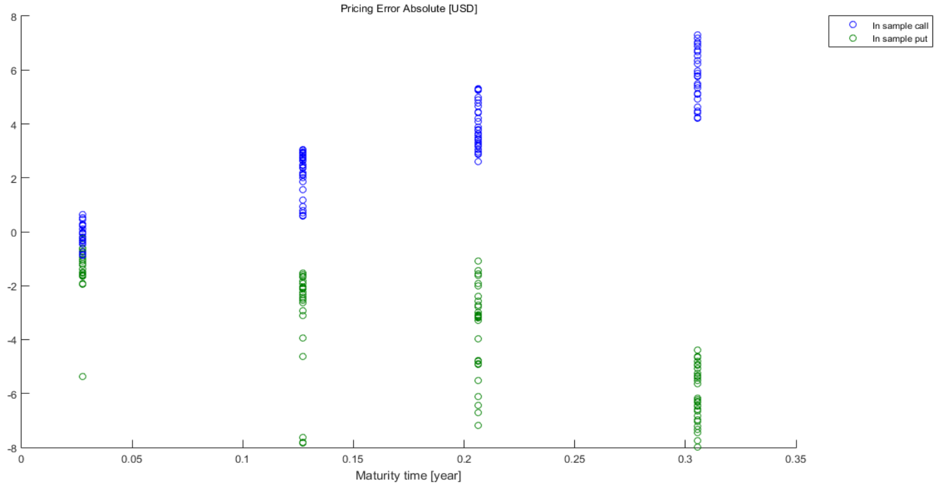


Figure 21: Absolute in-sample pricing error for the reference case.

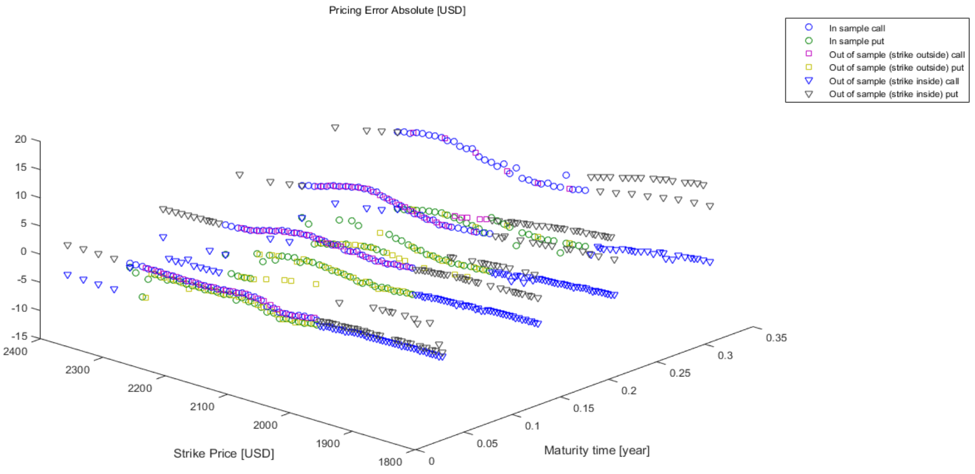


Figure 22: Absolute pricing errors for in and out-of-sample reference case options at their corresponding maturities. The in and out-of-sample options have the same maturities but different strikes. The strikes for the out-of-sample are partly intermediate strikes to the in-sample options and partly outside of the strike price range for the in-sample.

Absolute and relative errors are displayed for the options at their maturities. The errors are given in both units of money (USD) and in implied volatility. Furthermore, most surfaces have increasing absolute pricing errors for higher time to maturities.

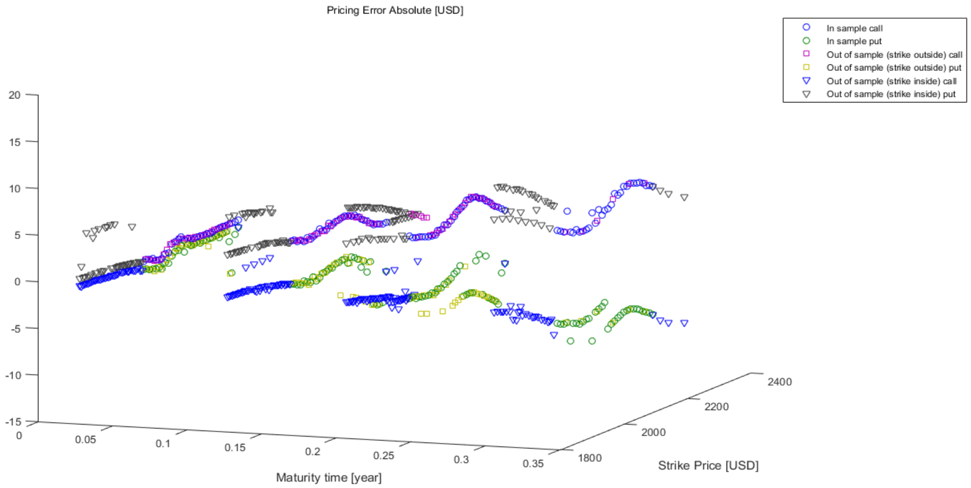


Figure 23: Absolute in and out-of-sample pricing errors w.r.t. time and strike for the reference case

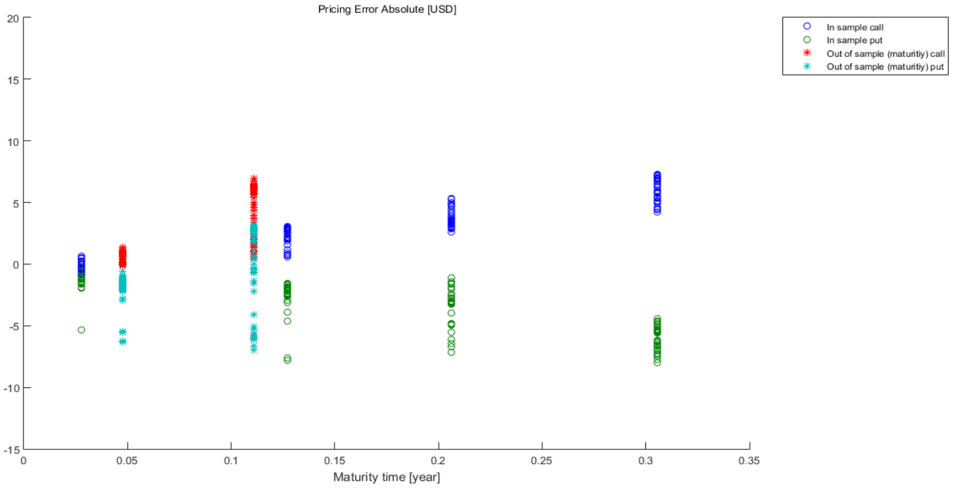


Figure 24: Absolute in and out-of-sample pricing errors. The out-of-sample options have different time to maturities than the in-sample-options.

11.7 Risk-neutral Distribution

A typical risk neutral distribution can be seen in figure 26 and figure 25. Figure 26 is a surface plot for the risk neutral distributions for the whole grid. Figure 25 is the seven specific option maturities, for the long data series, with 236 work days. Additionally, there is also a small "hump" in the distribution's left tail, since the grid was too small on the lower end.

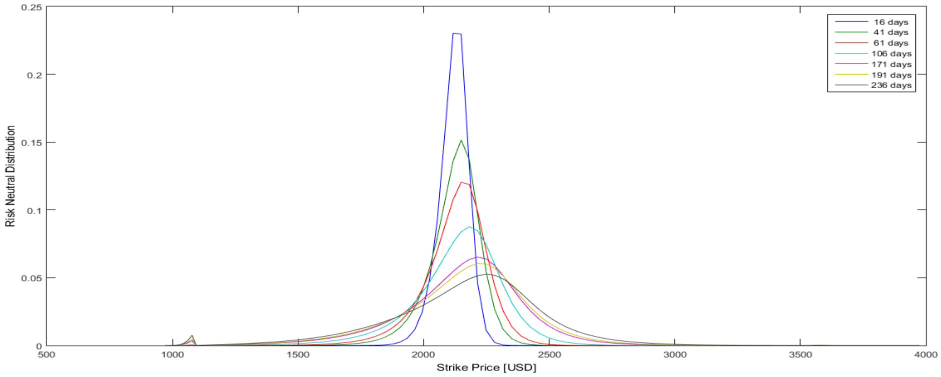


Figure 25: Typical risk-neutral distribution

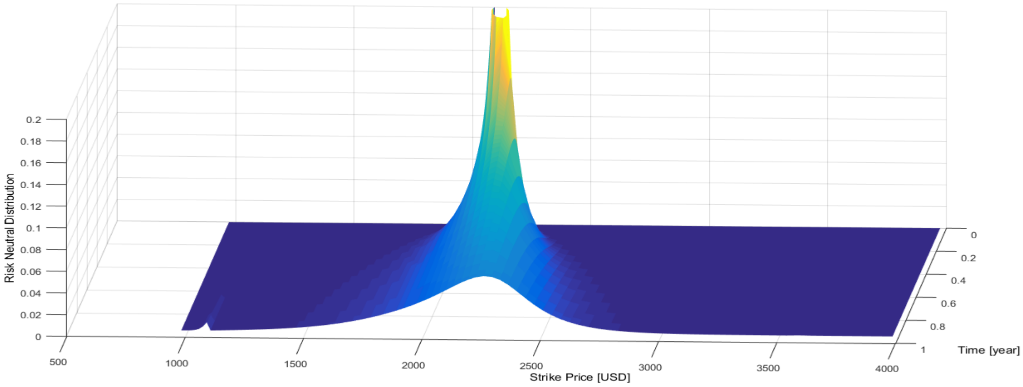


Figure 26: Typical surface of risk-neutral distributions.

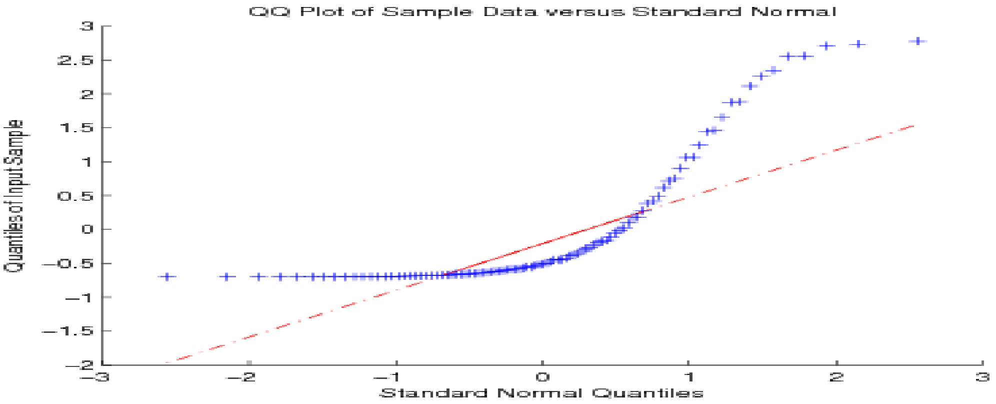


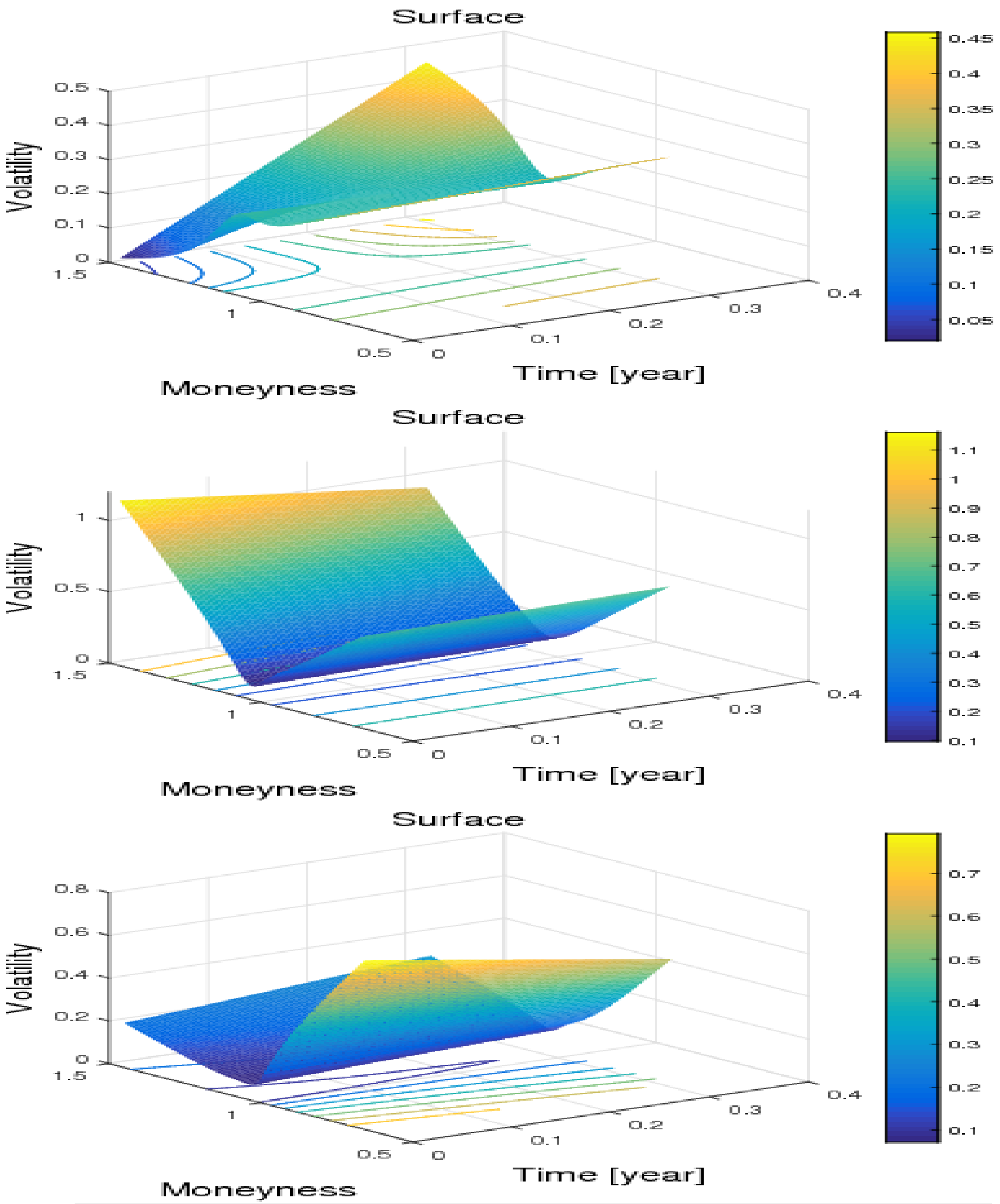
Figure 27: QQ-plot for the risk neutral distribution with the longest maturity (236 days)

11.8 Stability

The following section covers the results regarding the parameter stability.

11.8.1 Same options over time

A range of options were chosen at 2015-05-06 and data for the same option set was collected the two following two days, the data was retrieved at 16:00 CET. A surface was extracted for the three days with the same parameters. The results are presented in figure 28.



Parameter	a^T	a^K	a^{TT}	a^{KK}	a^{KT}	E	F
Value	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$	$5 \cdot 10^{0}$	$5 \cdot 10^{+5}$	$5 \cdot 10^{0}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{0}$

Figure 28: The extracted surfaces from the subsequent case (with the same options and parameters).

11.8.2 Different Number of Options

The reference case was used for this test. The upper and lower range of option strikes were unchanged, but the number of intermediate options were changed.

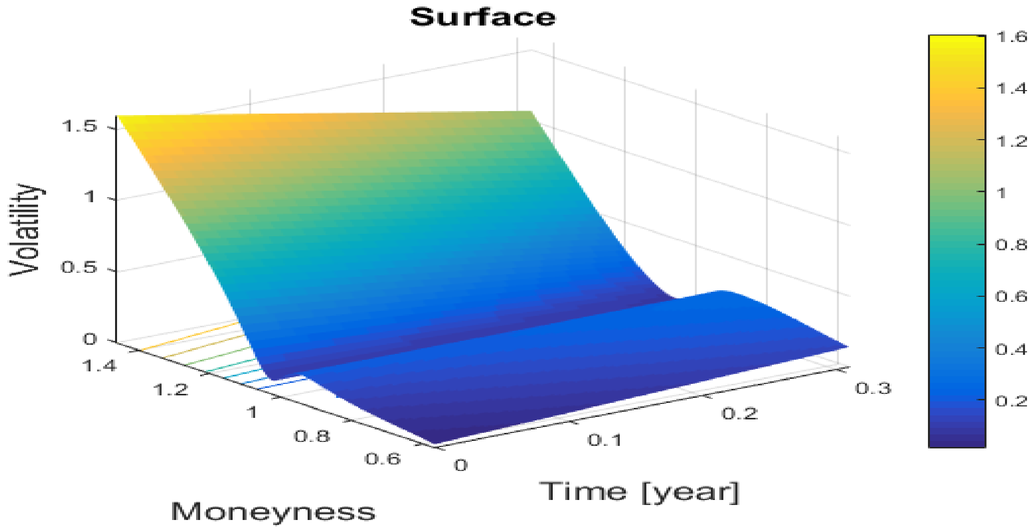


Figure 29: The strikes are closer together than the reference case in the previous sections

12 Discussion

The discussion is based on the results presented in the previous section, presented in the order seen fit.

12.1 Convergence

This section is divided into two parts, convergence of the problem and the rate of convergence. The definition we used for convergence is that independent on the starting volatility, which is assumed to be flat, the same surface should be extracted. The problem if the optimization do not converge is that is hard to find an optima. The reference case was an extraction with 25 % as starting volatility, which is presented in 12.2.1reffig:Convergence - three surfaces. The extraction was repeated with different starting volatilities and in figure 7, where the absolute and relative errors to the reference surface are presented. Both the relative and absolute errors are small and therefore they are considered to have converged.

The convergence of the method is measured by ∇L, where a low value is better than a high. The ∇L for a method with a high rate of converge has a value that decrease fast, which is wanted. The value of ∇L is in figure 8 plotted against the iteration. It is possible to observe that the decrease is exponential (linear for a logarithmic scale). Here, an interesting phenomena was present, with the starting volatility of 65 % the norm became greater before it started to decrease. This was present for more than one starting volatility, but a higher starting volatility gave a slower rate of convergence. However, the method does not converge for all parameters setups. For instance, the case when a^{TT} was scaled 100 times higher than the reference case the method failed to converge. The optimizer reached a value of the norm and after that point it did not improve the solution, but rather change it without any effect in the objective function. Many times when a^{TT} was too large, the method failed to converge, i.e. it failed to stop on a stop condition. We have not been able to find the underlying issue.

Furthermore, if the price error penalty, E is too large, then the surface will have big peaks, and these peaks cannot be negative in the implementation. Hence, the norm of the gradient will not be sufficiently small for convergence, and instead the optimization will halt, since the step length is too small. We believe that this problem arises when the search direction is too "extreme", an example can be seen in figure 14, where there are some peaks that are problematic. These peaks can sometimes come close to zero, and the current design of the optimizer is to shorten the step length such that negative volatilities never exist. A possible, ad hoc, solution is to let the new surface be negative temporarily and then change all negative points to the lowest allowed volatility. This surface is not a good solution since the next iteration will probably want an almost identical search direction, which results in the same problem again. The solution is also mathematically troublesome, since the search direction is in fact changed. A better solution is to change the roughness penalties as well, to still ensure a smooth surface.

An even better solution, which is much harder, is to use the second constraint in the optimization problem (14), $U \geq U_l$. This constraint was relaxed in (49), to be able to solve the problem easier. A possible solution is to rewrite (14) with a barrier function. The constraint,

$$U \geq U_l \Leftrightarrow U - U_l \geq 0 \Leftrightarrow C \geq 0, \tag{169}$$

where $C \equiv U - U_l$ is rewritten. Then the constraint can be replaced by a adding a term[8],

$$-\sum_{i=0}^{n_k}\sum_{j=0}^{n_t} \mu_{i,j} C_{i,j}. \tag{170}$$

This term will prevent the problem from violating the constraint, but the trade-off is that numerical problems can arise that need to be considered while solving the problem, for instance numerical issues with the logarithm close to zero and not letting the barrier to remove much of the solution space.

12.2 Parameters

The roughness parameters $\boldsymbol{a^T}$, $\boldsymbol{a^K}$, $\boldsymbol{a^{TT}}$, $\boldsymbol{a^{KK}}$ and $\boldsymbol{a^{TK}}$ and the pricing parameter E were varied, but not F. F was constant since all penalty combinations can be achieved by varying only E and thus have the sought-after

[8] There exists other possible barrier functions, this is merely an example.

effect on the solution. A high value for E can be represented by a low value for F and thus it was set constant to 1.

It is rather obvious that the most important parameters to assign the adequate magnitude to is $\boldsymbol{a^{KK}}$ and E, to require the correct shape and low price errors. An inadequately large value, extracts a flatter surface as can be seen in figure 9. Furthermore, the second derivative with respect to time is a sensitive parameter. Extracting a surface, with an inadequately large $\boldsymbol{a^{TT}}$, is difficult since the optimizer will then have convergence issues. Furthermore, it is not of great importance to see the absolute values of the parameters, the greatest concern is the relative values to each other. So to change all parameters with a same scale factor do not effect the problem[9]. Hence, it is essential to understand the dynamics of the parameters, and how they effect the optimization problem. Additionally, how these parameters should be changed together in order to get the sought for behavior. In the current implementation, these parameters are restrained to be constant.

12.2.1 Variable Parameters

The parameters were constant for the entire grid and all options, e.g. high curvature for short times are equally penalized as curvature for long times or a pricing of x units of money are equally penalized, independent of the market price of the options. The advantage of constant parameters is that the human interference is reduced. An alternative that we believe can be of great interest is to use different penalties for different options and for different parts of the grid. A clear example of this is when more options are used, figure 29. This surface was extracted with the same options as the reference case but had intermediate strikes as well, i.e. more information is present for the optimization. A comparison between figure 29 and 6 is that the solution is not close to having the same shape, and for the case with more options the surface does not have a realistic shape.

The aim of the thesis was not to find the optimal parameters. The parameters that are presented in the results and discussion should therefore not be seen as the optimal parameters, but merely as one possible set of parameters that work for the specific setup. These might however act as a guidance for refining the model. The current approach of constant parameters do we not believe to be the final solution. Furthermore, we think that the model must be much more dynamic and self-adjusting to different setups. The solution that we think would be a good is to use this method as a function in another optimization problem. An optimization problem that itself optimizes the parameters for this method. The new optimization problem should change the parameters and settings of this method, and its solution is the best possible parameters for this method. The problem with such a solution is that it is really theoretical and computationally strenuous.

12.2.2 Number of Options

Increasing the number of options also increases the importance of the pricing error in the objective function, and therefore the optimization. In this case the number of options have been approximate doubled and since they are intermediate the value of their pricing errors are almost the same as the existing, which means that the value of the pricing errors have been doubled. This effect is not proper, since adding (adequate) information should make the solution better. Therefore, the pricing penalties will be needed to be adjusted as more options are added. How to scale E according to the number of total options is not trivial. Linear scaling was implemented where, n_1 number of options in the reference had E_1 as roughness penalty and the number of options in the new case was n_2 with the pricing error parameter scaled to $E_2 = E_1 \cdot \frac{n_1}{n_2}$. Furthermore, the parameters were also squared, $E_2 = E_1 \cdot \left(\frac{n_1}{n_2}\right)^2$, neither of these approaches were enough to solve the problem.

Furthermore, just to consider the number of options is not enough. A extreme scenario, to emphasis the point, is that the out-of-money (in-the-money) side has very few option in relation to at-the-money and in-the-money (out-of-money) side. Then the side with few options must have higher penalties to correct the unbalance, otherwise one side will have to much weight. The problem with this approach is that the degrees of freedom will growth very fast. Firstly, how many different areas the options will be divided into and also how much the penalties should be changed. It is also very plausible that the human interference will be greater since the problem can ultimately be reformed to yield any result. The problem is not only the human interference, but it will also be impossible to use it as a function in a greater optimization problem, since the new optimization problem is very restricted to these choices made.

[9] If the factor is “unfeasibly” small or large computational problems can arise, which affects the solution.

12.2.3 Curvature

As already mentioned, penalizing curvature with respect to strike price with a constant penalty is not the best approach. The sought after behavior is that the curvature should be higher for shorter maturities compared to longer. The approach that can be used is have different curvature penalties for different time. The problems are the same as for E, how to divided the time and what should the relation between the different penalties be.

12.2.4 Parameter Stability

The method was designed to have stable parameters for small changes, i.e. they should at least work for the subsequent days. And if a large history of options for a given underlying is available, very good parameters could be possible to extract. If the parameters change drastically over time the practical use of the model is much more limited, since much computational power will be used for extracting new surfaces.

The results in figure 28 show that the same parameters did not give the same surface appearance for the three subsequent days. This does not prove that the parameters are changed much between different days, since we not have found equivalent parameters, but if the parameters would have worked it is more likely that they had been stable. The different was so great that the extraction of the first day, 2015-05-06, did not converge and the other two did converge but with very different appearance. The only change between these three days is the time to maturities and the options' market prices. Therefore, it was probably a problem with the parameter a^{TT} or E. However, given better (optimal) parameters the stability over time will probably be better, and if this is solved as an optimization problem, the human interference is lower. Hence, if very good (non-constant and problem adapted) parameters are available, it is more likely that these should be less sensitive. The parameters we used could for instance work on average for the surface, but the proper parameters would most likely not be constant, and consider for example the amount of options in that area and at what maturity time this area is situated.

12.3 Risk free Interest Rate and Dividend Yield

The "reference", "long" and the subsequent" case and the were all extracted with zero risk free interest rate and dividend yield. The "reference case" was also extracted with risk-free interest rate and dividend yield.

We tried to use different types of constant dividend yields and risk-free rates for the entire period but with small improvements of the surface. The general shape of the surfaces were almost unchanged. Some option prices were improved by a adding the dividend yield and risk-free rate but other were less correctly priced. We thought that the estimate was to bad and to find better estimate was beyond the scope of our work and therefore no more testing was conducted.

The bad estimate of the dividend yield and interest rate (constant and usually zero), is probably the cause of the strange behavior of increasing pricing errors as time passes, and where the put call parity does not hold. This behavior can be seen in figure 23 and figure 24, where the absolute pricing errors increase for longer maturities.

If there exists an error in the estimate, it is likely that this will increase over time. since the maturities of longer times have been exposed to the same problem during longer time and should therefore be more effected. The only real solution to this problem is to use better estimates for the expected dividend and interest rate.

12.4 Non-equidistant Time and Strike Increments

One property that was sought after, but where the implementation gave bad properties, was to use non-equidistant time steps. Our (internal) supervisor Blomvall thought that is would be beneficial to use very short time steps for the first time steps, perhaps as low as 15 [minutes], which then would then gradually be increased, to larger time increments, until days was achieved. The best would be to have very short time steps for the entire time. The problem with this is that the computational cost is very high. This trade-off needs to be taken into consideration. The advantage of having very short time steps in the beginning is that tree will branch very fast which makes it possible to have more points close to the edges to control. The behavior that is sought after is that the points close to the edges would be very high. If the extraction is for long times it is possible to lower the computational cost by having longer time steps for times far in to the future, because the surface should be more stable by then.

The problem we believe is that penalties are constant, and cannot handle the different non-equidistant time increments. One problem is that a^{TT} does not have the same behavior as before, since the integrating factor does not work accordingly. Another problem is with a^{KK} because the roughness with respect to the second order derivative with respect to strike is increased for the same surface. A surface with a given second order derivative, with respect to strike price can be changed by the discretization. So merely by changing the discretization, the dynamic of the problem will be affected, and this needs to be mitigated by changing the parameters as well. Unfortunately, there was no time to derive a theory to refine the parameters when the discretization was changed.

The length between the strike prices have the same arguments as the for the time increments. If the distance between the strike prices are changed, then the dynamic of the problem is also changed.

Furthermore, we found that parameters that performed well for equidistant discretization, had bad characteristics for unequidistant time increments.

It seemed that the curvature along the price-direction was affected by the change. The time increments are much smaller and therefore the resulting penalty for the curvature in price is much higher. Previous a surface with a one day increment were effected in two ways[10]. If this increment is than divided into smaller steps, e.g. four 15 minutes segments, four 30 minutes segments, three 1 hour segment and three 6 hours segments, a graphical representation can be see in figure 30. The resulting behavior is that the same time period have fifteen parts that contribute to penalize the curvature. The same surface but with another discretization have different roughness measure, which is an unwanted behavior.

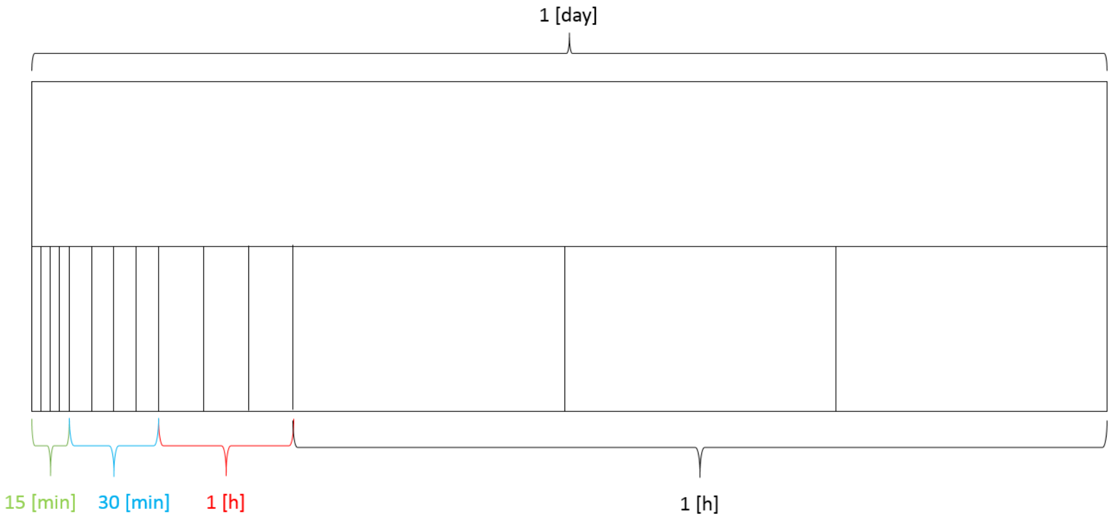


Figure 30: A graphical representation how different time increments for a one day time period.

Some attempts were made to address this problem. One, implemented, idea was to scale the parameter a^{KK}. For example if the old time period $[T_{t_o}, T_{t_o+1}]$, was split into m increments. In the new discretization, the old time period is represent by $[T_{t_n}, \dots, T_{t_n+(m-1)}]^{11}$, where $T_{t_o} = T_{t_n}$ and $T_{t_o+1} = T_{t_n+(m-1)}$. The parameter in the old discretization, ${}_oa^{KK}_{t_o,k}$, is then scaled to calculate the new parameters, ${}_na^{KK}_{\tau,k}$ as

$$ {}_na^{KK}_{\tau,k} = \frac{{}_oa^{KK}_{t_o,k}}{m}, \ \forall k = 0, \dots, n_k, \ \forall \tau = t_n, \dots, t_n + (m-1). \tag{171}$$

We also tried to use m^2, instead of m, as denominator in equation (171). This adjustments were not successful and the extracted surfaces were not improved much. The conclusion was that we have not been able to understand the dynamic enough to address this problem.

[10] It is also possible to argue that it is only affected by one side and that the next side belongs to the next increment. The difference will however not change the argument.

[11] The letter “o” denotes “old” and the letter “n” denotes “new”.

12.5 Risk-Neutral Distribution

The risk neutral distribution can be see in figure 25 and 26. The general shape of them seems legit, but there is a hump around 1000. This is an unwanted hump this exists because the grid has a two small down-side and hence the moat has accumulated too much distribution. If there is a hump in one of the sides the moat has absorbed some probability and therefore changed the distribution. The QQ-plot in figure 27 makes it clear that the risk-neutral distribution is not normally distributed.

12.6 Computational Issues

The computational problems are divided in three areas. The first is that the memory (8GB) is too small, and can not handle too large problems, which arises if the grid is to big(either too fine discretization or too large price spectrum). Furthermore, the next and biggest problem is the solving of equation system (138), that needs to be solved several times. This is the part of the extraction that take the most time. We do not have any solution to this problem we can only conclude that this is the most costly operation.

Additionally, we have learned that reversing the automatic differentiation can be very costly to compute if implemented in a bad way. The first approach was to overload all functions needed for the problem, the problem is that it creates a huge overhead. It is of great importance to minimize the overhead and try to get just the derivative from a function and not nestled in several variables.

In the moment matching, the first moment and the full allocation criterion was found to be almost linearly dependent, with the result that the matrix is singular. Since the information is more or less the same in these equations, the first moment was removed from the equation system. However, the expression will approximately be true from the full allocation criterion. For the other parts of methodology it is more important that the full allocation criterion is true and hence it remained in the equation system.

12.7 Pricing Errors

A general behavior of the pricing is that the errors increase as time to maturity increases. Furthermore, the in-sample errors tend to be smaller than the out-of-sample errors, which is reasonable since the in-sample options ere used for the extraction and should thus fit the surface better. However, the out-of-sample options seem to follow the in sample options. If maturities are excluded the pricing error mainly depends on what maturity these have, since the general behavior of pricing errors of in sample options, which increase for longer maturities. This might be due to that options with longer time to maturity affect shorter maturities and therefore the pricing of these options is better. Since there is more information on the local volatility (more options depend on these points in the grid) these might be more adequately priced. Another idea is that since we used constant interest rate and dividends, the error in these compared to the market are larger as the time to maturity increases.

Furthermore, a general behavior, both for out of sample and in sample options is that the call options are over-priced and the puts are under-priced with the method. We believe that this problem is partly a problem that we do not take risk free rate and dividend yield to account, and that these have been set fix to an average of the expected interest rate and dividends, respectively.

It is essential that the surface performs well for out of sample options both with respect to time and price. Mainly since this allows any option with a given (exotic) payoff to be priced from the surface at any given point, such as OTC contracts. The out of sample options are not part of the optimization problem, but they have an important role for the practical use of the model in for it to be a real method. One way to continue the development of this method is to incorporate “out of sample” options in the optimization problem.

An idea for future analysis is to use other more exotic options in the extraction of the surface and also for the out of sample testing. Examples of these more exotic options could be barrier options (such as knock-out and knock-in). The problems is that the market price data not is available, because that they are over-the-counter contracts. They are also not as easy to price as the vanilla call and put options since barrier options are path-dependent, the risk-neutral distribution at the time to maturity is not enough. For example a *Up-and-out* call option has the same payoff at maturity as an vanilla call option if and only if the underlying has been below the barrier the whole maturity time, otherwise it is worth zero.

12.8 Conclusion

The surfaces that were extracted were smooth and priced the options according to the market quoted prices. It is also reasonable to believe, in our opinion, that the method itself have possibilities to extract surface that are good enough for the market, especially if refined parameters are used. However, these are not a trivial matter to determine, and they should also be adapted to the choice of discretization both with respect to time and price. The method has exponential rate of convergence, so if the bottle neck regarding the automatic differentiation is solved the method will be much more competitive with respect to time. The next crucial step is to make the method more dynamic and making it self adapting to the conditions.

A Details for Roughness

In section 3.1 the roughness measure was defined but only the first term, the rest of them are present in this Appendix.

A.1 Second Term

The block,

$$B_K = \begin{bmatrix} -1 & 1 & 0 & \dots & 0 \\ 0 & -1 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 1 \end{bmatrix}, \tag{172}$$

which has the dimension, $[n_K \times (n_K + 1)]$ and the matrix,

$$A_K = \begin{bmatrix} B_K & & \\ & \ddots & \\ & & B_K \end{bmatrix}, \tag{173}$$

which has the dimension, $[(n_T + 1)n_K \times (n_T + 1)(n_K + 1)]$ and it is possible to write

$$\frac{1}{2}\boldsymbol{u}^T A_K^T \operatorname{diag}(\boldsymbol{\hat{a}^K}) A_K \boldsymbol{u}, \tag{174}$$

where

$$\boldsymbol{\hat{a}^K} = \begin{pmatrix} \hat{a}^K_{0,0} & \dots & \hat{a}^K_{n_{K-1},0} & \hat{a}^K_{0,1} & \dots & \hat{a}^K_{n_{K-1},n_T} \end{pmatrix}^T. \tag{175}$$

A.2 Third Term

The block,

$$B_{TT} = \begin{bmatrix} \Delta_{T_1} & -\Delta_{\tilde{T}_0} & \Delta_{T_0} & & & \\ & \Delta_{T_2} & -\Delta_{\tilde{T}_1} & \Delta_{T_1} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Delta_{T_{n_T-1}} & -\Delta_{\tilde{T}_{n_T-2}} & \Delta_{T_{n_T-2}} \end{bmatrix}, \tag{176}$$

where $\Delta_{\tilde{T}_i} \equiv (\Delta_{T_i} + \Delta_{T_{i+1}}),\ \forall i = 0, \dots, n_T - 2$. B_{TT} has the dimension $[(n_T - 1) \times (n_T + 1)]$ and it is possible to write,

$$A_{TT} = \begin{bmatrix} B_{TT} & & \\ & \ddots & \\ & & B_{TT} \end{bmatrix}, \tag{177}$$

which has the dimension $[(n_K + 1)(n_T - 1) \times (n_K + 1)(n_T + 1)]$ and it is possible to write

$$\frac{1}{2}\boldsymbol{u}^T P^T A_{TT}^T \operatorname{diag}(\boldsymbol{\hat{a}^{TT}}) A_{TT} P \boldsymbol{u}, \tag{178}$$

where

$$\boldsymbol{\hat{a}^{TT}} = \begin{pmatrix} \hat{a}^{TT}_{0,1} & \dots & \hat{a}^{TT}_{0,n_T-1} & \hat{a}^{TT}_{1,1} & \dots & \hat{a}^{TT}_{n_K,n_T-1} \end{pmatrix}^T. \tag{179}$$

A.3 Fourth Term

The block,

$$B_{KK} = \begin{bmatrix} \Delta_{K_1} & -\Delta_{\tilde{K}_0} & \Delta_{K_0} & & & \\ & \Delta_{K_2} & -\Delta_{\tilde{K}_1} & \Delta_{K_1} & & \\ & & \ddots & \ddots & \ddots & \\ & & & \Delta_{K_{n_K-1}} & -\Delta_{\tilde{K}_{n_K-2}} & \Delta_{K_{n_K-2}} \end{bmatrix}, \tag{180}$$

where $\Delta_{\tilde{K}_i} \equiv (\Delta_{K_i} + \Delta_{K_{i+1}}),\ \forall i = o, \dots, n_T - 2$. B_{KK} has the dimension, $[(n_K - 1) \times (n_K + 1)]$, and several blocks,

$$A_{KK} = \begin{bmatrix} B_{KK} & & \\ & \ddots & \\ & & B_{KK} \end{bmatrix}, \tag{181}$$

which has the dimension, $[(n_T + 1)(n_K - 1) \times (n_T + 1)(n_K + 1)]$, and it is possible to write

$$\frac{1}{2}\boldsymbol{u}^T A_{KK}\, \mathrm{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KK}}) A_{KK}\boldsymbol{u}, \tag{182}$$

where

$$\hat{\boldsymbol{a}}^{\boldsymbol{KK}} = \begin{pmatrix} \hat{a}^{KK}_{1,0} & \dots & \hat{a}^{KK}_{n_K-1,0} & \hat{a}^{KK}_{1,1} & \dots & \hat{a}^{KK}_{n_K-1,n_T} \end{pmatrix}^T. \tag{183}$$

A.4 Fifth Term

The fifth term is different than the previous. The summand in the last term is

$$U_{k+1,t+1} - U_{k-1,t+1} - U_{k+1,t-1} + U_{k-1,t-1} = (U_{k+1,t+1} - U_{k-1,t+1}) - (U_{k+1,t-1} - U_{k-1,t-1}) \tag{184}$$

the matrix will be build in a special way first the term,

$$U_{k+1,t+1} - U_{k-1,t+1} \tag{185}$$

is constructed and than

$$U_{k+1,t-1} - U_{k-1,t-1} \tag{186}$$

the to resulting matrices will be subtracted. The first term can be written as

$$B_{KT} = \begin{bmatrix} -1 & 0 & 1 & 0 & \dots & 0 \\ 0 & -1 & 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & -1 & 0 & 1 \end{bmatrix}, \tag{187}$$

which has the dimension, $(n_K - 1) \times (n_K + 1)$ then

$$A_{KT_1} = \begin{bmatrix} O, \begin{bmatrix} B_{KT} & & \\ & \ddots & \\ & & B_{KT} \end{bmatrix} \end{bmatrix}, \tag{188}$$

where O is a zero matrix with the dimension, $[(n_K - 1)(n_T - 1) \times 2(n_K + 1)]$. The second term can be construed with the same method,

$$A_{KT_2} = \begin{bmatrix} \begin{bmatrix} B_{KT} & & \\ & \ddots & \\ & & B_{KT} \end{bmatrix}, O \end{bmatrix}. \tag{189}$$

The subtraction,

$$A_{KT} = A_{KT_1} - A_{KT_2} \tag{190}$$

makes it possible to write

$$\frac{1}{2}\boldsymbol{u}^T A_{KT}^T\, \mathrm{diag}(\hat{\boldsymbol{a}}^{\boldsymbol{KT}}) A_{KT}\boldsymbol{u}, \tag{191}$$

where

$$\hat{\boldsymbol{a}}^{\boldsymbol{KT}} = \begin{pmatrix} \hat{a}^{KT}_{1,1} & \dots & \hat{a}^{KT}_{n_K-1,1} & \hat{a}^{KT}_{1,2} & \dots & \hat{a}^{KT}_{n_K-1,n_T-1} \end{pmatrix}^T. \tag{192}$$

B Mathematical Conventions and Definitions

The appendix contains some mathematical definitions that can be find in common mathematical literature but reproduced here for convenience.

B.1 Hadamard Product

The hadamard product is a matrix operator. It is a "element-wise" multiplication. Two matrixis A and B with the same size, $n \times m$, so

$$A \circ B = C,$$

where C has the same size $n \times m$ and where the elements is

$$c_{i,j} = a_{i,j} b_{i,j} \ \forall i = 1, \ldots, n \ , j = 1, \ldots, m. \tag{193}$$

B.2 Vectorization

Definition 11. *Vectorization of a matrix is a linear transformation which converts the matrix into a column vector. Specifically, the vectorization of a $m \times n$ matrix*

$$A = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{194}$$

denoted by $vec(A)$ is a $mn \times 1$ column vector,

$$vec(A) = [a_{1,1}, \ldots, a_{m,1}, a_{1,2}, \ldots, a_{m,2}, \ldots, a_{1,n}, \ldots, a_{m,n}] \tag{195}$$

B.3 Submatrix

A submatrix of a matrix is often used.

Definition 12. *Let M be a $n \times m$ matrix,*

$$M = \begin{bmatrix} m_{1,1} & \cdots & m_{1,m} \\ \vdots & \ddots & \\ m_{n,1} & \cdots & m_{n,m} \end{bmatrix}. \tag{196}$$

A submatrix can be defined as

$$M_{[i,j],[k,l]} = \begin{bmatrix} m_{(i,k)} & m_{(i,k+1)} & \cdots & m_{(i,l)} \\ m_{(i+1,k)} & m_{(i+1,k+1)} & \cdots & m_{(i+1,l)} \\ \vdots & \vdots & \ddots & \vdots \\ m_{(j,k)} & m_{(j,k+1)} & \cdots & m_{(j,l)} \end{bmatrix}, \tag{197}$$

where $i \geq 1, j \leq n$ and $k \geq 1, l \leq m$.

Remark 3. *Note that in this document the indexation sometimes starts at zero and thus a matrix, M, with the dimension, $n \times m$ includes the rows from 0 to $n-1$ and the columns 0 to $m-1$.*

B.4 Kolmogorov Axioms

Definition 13. *Let $(\Omega, \mathcal{F}, P)$ be a measure space, with sample space Ω, event space $\mathcal{F}$ and probability measure P. Given this measure space the* ***Kolmogorov axioms*** *can be formulated, (Gut, 2009, p.4).*

Non-negativity

All events, $A \in \mathcal{F}$ have a non-negative probability, $P(A) \geq 0$.

Unity

The probability of the sample space is one, $P(\Omega) = 1$.

Countable Additivity

Let $\{A_n,\ n \geq 1\}$ be pairwise disjoint events, joint by the union A, then

$$P(A) = \sum_{n=1}^{\infty} P(A_n). \tag{198}$$

References

Barkhagen, M. Optimal decisions in the equity index derivatives markets using option implied information, 2015.

Black, F. and Scholes, M. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81(3): 637–654, May - Jun. 1973.

Blomvall, J. Financial optimization - lecture notes, 2007.

Bodie, Z., Marcus, A. J., and Kane, A. *Investments.* McGraw-Hill Education, New York,, 10 edition, 2014.

Cochrane, J. H. *Asset pricing.* Princeton Univ. Press, Princeton [u.a.], 2001.

Derman, E. and Kani, I. Riding on a smile. 7, 1994a.

Derman, E. and Kani, I. The volatility smile and its implied tree. *Goldman Sachs, The Volatility Smile and Its Implied Tree*, 1994b.

Derman, E. and Kani, I. Stochastic implied trees: Arbitrage pricing with stochastic term and strike structure of volatility. *Quantitative Strategies Technical Notes*, 1997.

Derman, E., Kani, I., and Zou, J. Z. The local volatility surface unlocking the information in index option prices. *Quantitative Strategies Research Notes*, 1995.

Dupire, B. Pricing with a smile. *Risk*, 7, 1994.

Ekblom, J. Lecture 2 & 3, tppe53 - financial valuation methodology, 2014.

Forth, S. A., Tadjouddine, M., Pryce, J. D., and Reid, J. K. Jacobian code generated by source transformation and vertex elimination can be as efficient as hand-coding. *ACM Trans. Math. Software*, 30:266, 2004.

Griewank, A. and Walther, A. *Evaluating Derivatives - Principles and Techniques of Algorithmic Differentiation.* Society for Industrial and Applied Mathematics, Philadelphia, 2nd edition, 2008.

Gut, A. *An Intermediate Course in Probability.* Springer, New York, 2nd edition, 2009.

Hull, J. C. *Options, Futures and Other Derivatives: Global Edition.* Pearson, 8 edition, 2011.

Kelly, J. L. A new interpretation of information rate. *Information Theory, IRE Transactions on Information Theory*, 2(3):185–189, 1956.

Latané, H. A. and Rendleman, R. J. J. Standard deviations of stock price ratios implied in option prices. *The Journal of Finance*, 31(2, Papers and Proceedings of the Thirty-Fourth Annual Meeting of the American Finance Association Dallas, Texas December 28-30, 1975):369–381, May 1976.

Long, J. The numeraire portfolio. *Journal of Financial Economics*, 26(1), 1990.

Luenberger, D. G. *Investment Science.* Oxford University Press, 1998.

Merton, R. and Samuelson, P. Fallacy of the log-normal approximation to optimal portfolio decision-making over many periods. *Journal of Financial Economics*, page 67.

Neidinger, R. D. Introduction to automatic differentiation and matlab object-oriented programming. *SIAM Review*, 52(3):545–563, 2010.

Platen, E. A benchmark approach to investing and pricing. *Working paper*, 2009.

Rall, L. B. and Corliss, G. F. An introduction to automatic differntiation. *Computational Differentiation: Techniques, Applications and Tools*, pages 1–17, 1996.

von Neumann, J. and Morgenstern, O. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition).* Princeton University Press, 60th anniversary commermorative edition edition, March 2007.

Upphovsrätt

Copyright