Technische Universiteit Delft
Faculteit Elektrotechniek, Wiskunde en Informatica
Delft Institute of Applied Mathematics

Het prijzen van financiële opties met gebruik van stochastische collocatie en lokale volatiliteit.

(Engelse titel: The pricing of financial options using stochastic collocation and local volatility.)

Verslag ten behoeve van het
Delft Institute of Applied Mathematics
als onderdeel ter verkrijging

van de graad van

BACHELOR OF SCIENCE
in
TECHNISCHE WISKUNDE

door

JAN TIJINK

Delft, Nederland
Augustus 2016

BSc verslag TECHNISCHE WISKUNDE

"Het prijzen van financiële opties met gebruik van stochastische collocatie en lokale volatiliteit."

(Engelse titel: "The pricing of financial options using stochastic collocation and local volatility.)"

JAN TIJINK

Technische Universiteit Delft

Begeleider

Prof.Dr. C. W. Oosterlee

Overige commissieleden

Dr.ir. M. Keijzer

Prof.dr.ir. C. Vuik

Dr. R. J. Fokkink

Augustus, 2016

Delft

Contents

1 Introduction

During the past few years, financial markets have been a growing topic of discussion. There is a large variety of financial instruments being traded on these markets of which stocks and bonds are most generally known. Options are derivative products. The value of these products is derived from another asset, usually referred to as the underlying asset. When options become more complex and are not seen as "standard" they are referred to as exotic options. A specific type of these is the barrier-option. This means that an option either becomes worth nothing when the price of the underlying asset hits the value of the barrier, or that the option is worth nothing unless the underyling asset hits the value of the barrier.

In this thesis the goal is to find a model that allows for pricing of options while allowing control over the forward-start volatility and adhering to the market implied volatility. This is done by using the Collocating Local Volatility (CLV) model and using this with two different implementations. First it is explained how options are priced in a basic way. After this, the Black-Scholes model for options is derived. From this we see the Black-Scholes model does not incorporate the implied volatility, thus another model is required. The local volatility model is then looked at and derived. This model does not allow for control of the forward-start volatility so we turn to the Collocating Local Volatility model. The CLV model is derived and implemented using the Finite Difference Method (FDM) and Monte Carlo (MC) simulation. The different models are compared to see where they differ and a reasoning of when to choose which method is discussed.

2 Option price modelling and the Black-Scholes model

In this section we will discuss what is the basics of option pricing and from this determine the famous Black-Scholes model. We will look at the weaknesses of this model and look into different models that overcome some of these weaknesses.

2.1 What is an option?

To understand what an option is we first need to know what an asset is. The term asset describes an object which has a value, which is known at this moment, but might change in the future. Gold is an example of an asset, as are shares in a company.

We define a **European call option**: A European call option gives its owner the right, but not the obligation, to purchase a predetermined asset at a predetermined price at a predetermined future date. The price for which the asset can be bought in the future is known as the strike price, or excercise price. The date at which you can excercise the option is called the maturity time or expiry time.

Suppose your bank gives you a European call option that gives you, the holder, the right to buy 10 shares in Royal Dutch Shell for $K =$ € 100 six months from now. After six months you will choose whether or not to exercise your option. The sensible choice depends on the asset price:

1. If the value of 10 shares of Royal Dutch Shell is below € 100 you will not excercise your option, since you can buy these shares for less money in the market. We say that the option is out-of-the-money (OTM).

2. If the value of 10 shares of Royal Dutch Shell is above € 100 you will excercise your option, since you can now sell these shares for more money in the market and make a profit. We say that the option is in-the-money (ITM).

Because you cannot lose money when you have this option, while having a chance to earn money, the option must have a certain value and the bank will not give it to you for free. The question now becomes, what will you have to pay for the option? This can be rephrased to the question: What is the fair value of this option?

Define the asset price as $S(t)$ given the asset price at time t and K the strike (or excercise) price. Note that that for a European call option the payoff is $(S(T) - K)$ if $S(T) \geq K$ at expiry time and 0 if $S(T) \leq K$. Thus the value of the European call option V_C at maturity T is

$$V_C = \max(S(T) - K, 0).$$

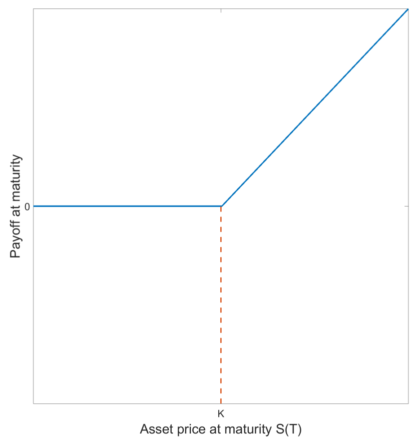


Figure 1: Payoff of a call option at the excercise time of the option.

A put option works the same as a call option, but has the possiblity of selling an option instead of buying an option. So a put option gives the holder the right to sell a predetermined asset, at a predetermined price at a predetermined future time. This means that the holder of a put option will excercise his/her option and sell an asset for strike price K if the asset value at expiry $S(T)$ is smaller than K. We find that the value of the European put option V_P at maturity is:

$$V_P = \max(K - S(T), 0).$$

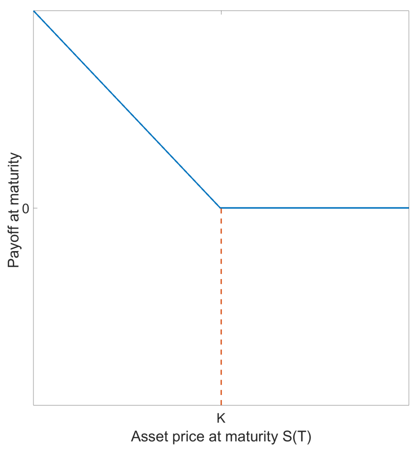


Figure 2: Payoff of a put option at the excercise time of the option.

2.2 Stochastic processes and martingales

Now we have established the basic idea about how an option works, we need to delve a little deeper into the underlying stochastic processes. The mathematical view of stochastic processes, brownian motions and martingales are presented which are used to construct models for asset pricing.

2.2.1 Stochastic processes and Brownian motions

A stochastic process $X(t)$ is a collection of random variables which are indexed by a time variable t. Suppose we have a collection of time points $T_1, T_2, \ldots, T_N$ and we are at time point T_k with

$T_1 < T_k < T_N$. Untill now we have observed values $X(T_1), X(T_2), \ldots, X(T_k)$. We need to use some mathematical tools to show we have knowledge of a stochastic process upto a certain time T_k. We do this with the help of a filtration. To construct this we order σ-algebras.

$$\mathcal{F}(T_k) = \sigma \{X(T_i) : 1 \leq k \leq i\}$$

Note that this gives us the property that $\mathcal{F}(T_1) \subseteq \mathcal{F}(T_2) \subseteq \cdots \subseteq \mathcal{F}(T_k)$. Thus the information that is available to us at time T_k is given by $\mathcal{F}(T_k)$. We call a process $\mathcal{F}(t)$-measurable when we know the value of this process $\forall s \leq t$. A simple example of this is the stock price process, we can find historical values of the stock price process, thus we know the realizations, but we do not know what is going to happen to the stock price in the future.

A Brownian motion (also known as a Wiener process) is often used to construct a Stochastic Differential Equation(SDE) and is a fundamental stochastic process. A process $W(t)$ is a Brownian motion if

1. $W(t_0) = 0$ almost surely (which means $\mathbb{P}[W(t_0) = 0] = 1$)
2. $W(t)$ is almost surely continous (which means $\mathbb{P}\left[\lim_{s \to t} W(s) = W(t)\right] = 1$)
3. $W(t)$ has independent Gaussian increments which means that $\forall t_1 < t_2 < t_3 < t_4$ we know that $W(t_2) - W(t_1) \perp W(t_4) - W(t_3)$. The distribution of the increment $W(t) - W(s) \sim \mathcal{N}(0, t-s)$ for $s \in [t_0, t)$.

A brownian motion is the continous time version of a random walk.

2.2.2 Martingales and Itô's lemma

Let us consider a probability space $(\Omega, \mathcal{F}, \mathbb{Q})$, where Ω denotes the collection of all possible outcomes, $\mathcal{F}$(t) is the σ-algebra and $\mathbb{Q}$ is the probability measure. Assume the process $X(t)$ is right continous and its left limits exist for $t \in [t_0, T]$. We call $X(t)$ a martingale with respect to the filtration $\mathcal{F}(t)$ under the measure $\mathbb{Q}$ if $\forall t < \infty$ the following holds:

1. $\mathbb{E}[|X(t)|] < \infty$
2. $\mathbb{E}[X(t)|\mathcal{F}(s)] = X(s)$ for $s < t$.

This implies that the expected future value for a martingale is its current value. The notation $\mathbb{E}[\cdot|\mathcal{F}]$ is the conditional expectation under the measure $\mathbb{Q}$ where $\mathcal{F}$ is the filtration described previously. Note that the Brownian motion is a martingale.
One of the assumptions of the Black-Scholes model which will be discussed in chapter 2.4 is that the asset price $S(t)$ follows a Geometric Brownian Motion (GBM). This means that it satisfies the Stochastic Differential Equation (SDE)

$$\mathrm{d}S(t) = \mu S(t)\ \mathrm{d}t + \sigma S(t)\ \mathrm{d}W^{\mathbb{P}}(t)$$

with $S(t_0) = S_0$. The Brownian motion in this is $W(t)$ under the $\mathbb{P}$ measure, which is the so called real-world measure.
We will now state Itô's lemma which will later be used to derive models for option pricing.

Theorem 1 (Itô's lemma)**.** *Suppose process $X(t)$ satisfies the SDE*

$$\mathrm{d}X(t) = \mu(t, X(t))\ \mathrm{d}t + \sigma(t, X(t))\ \mathrm{d}W(t)$$

with $X(t_0) = X_0$. Suppose also that for all $x, y \in \mathbb{R}$ there exists a positive N such that

$$|\mu(t,x) - \mu(t,y)|^2 + |\sigma(t,x) - \sigma(t,y)|^2 \le N\,|x-y|^2$$
$$|\mu(t,x)|^2 + |\sigma(t,x)|^2 \le N\left(1+|x|^2\right).$$

Let $g(t,X), X = X(t)$ be a function of X and t with continous partial derivatives $\frac{\partial g}{\partial t}, \frac{\partial g}{\partial X}, \frac{\partial^2 g}{\partial X^2}$. Then the stochastic variable $Y = g(t,X)$ is also a stochastic process governed by the same Brownian motion $W(t)$ with dynamics:

$$\mathrm{d}Y(t) = \left(\frac{\partial g}{\partial t} + \mu(t,X)\frac{\partial g}{\partial X} + \frac{1}{2}\sigma^2(t,X)\frac{\partial^2 g}{\partial X^2}\right)\,\mathrm{d}t + \frac{\partial g}{\partial X}\sigma(t,X)\,\mathrm{d}W(t).$$

The proof can be found in several books on stochastic processes such as [2] and is not of interest for this thesis.

2.3 Option pricing principles

It is now clear that options have value, thus if you trade them, you must agree on a price. One of the most important concepts of option valuation is the **no arbitrage principle**. This principle says that there is never the opportunity to make a risk-free profit that gives a greater return than the interest gained by depositing money in a bank. Essentially this means that if you want to make more money than the interest gained by putting money on a bank account, you have to take more risk. Assume there exists a risk-free interest rate r at which we can lend and borrow for any amount.If € 100 is deposited now, the value in one year will be € $100 \cdot e^r$. The other way around, if we have € 100 in a year, the value now is € $100 \cdot e^{-r}$. From this we find a relation between the European call and put options. Consider a call option and a put option on the same asset with strike price K and expiry date T. We will now consider two portfolios:

1. Π_A: one call option V_C and $K \cdot e^{-rT}$ cash
2. Π_A: one put option V_P and one asset S.

At expiry, the first portfolio's worth is

$$\max(S(T)-K,0) + K(e^{-rT}\cdot e^{rT}) = \max(S(T)-K,0) + K = \max(S(T),K).$$

At expiry the second portfolio has value:

$$\max(K-S(T),0) + S(T)(e^{-rT}\cdot e^{rT}) = \max(K-S(T),0) + S(T) = \max(K,S(T)).$$

Since these two portfolios always give the same payoff, it is logical that they must have the same value at any time between now and expiry. Therefore we can conclude that

$$V_C + Ke^{-rT} = V_P + S.$$

This relation is called **put-call parity**. Note that if the values of portfolio A and B differ, we can make a risk-free profit, thus allowing arbitrage. If the value of $\Pi_A > \Pi_B$ we would sell the call option and borrow the money and buy Π_B. This would directly give us the value $\Pi_A - \Pi_B > 0$, thus a risk-free profit instananeously. If $\Pi_B > \Pi_A$ we get a risk-free profit in a similar manner.

One of the most important assumptions made in pricing an option is the Efficient Market Hypothesis, which states that asset prices(in the market) fully reflect all available information. This essentially means that if we discount asset prices by the interest rate this is a martingale.

So under the risk-neutral measure the discounted stock price is a martingale. Since we can construct a replicating portfolio for an option, we see that the discounted value of this must also be a martingale. So

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{S(t)}{M(t)}\middle|\mathcal{F}(t_0)\right] = \frac{S(t_0)}{M(t_0)} = S(t_0)$$

for all $t \in [t_0, T]$. This also holds for options, so the discounted option price is a martingale. Because we know the terminal condition for European options, we can express the option value now as the expectation of the discounted payoff:

$$\mathbb{E}^{\mathbb{Q}}\left[\frac{V(t)}{M(t)}\middle|\mathcal{F}(t_0)\right] = \frac{V(t_0)}{M(t_0)} = V(t_0)$$

2.4 Black-Scholes model

Fisher Black and Myron Scholes published their article [3] in 1973 explaining a model on option and stock dynamics. It is to this day one of the most important models in the pricing of derivatives and Myron Scholes and Robert Merton (who gave the mathematical understanding for this model) received the Nobel prize for this. The Black-Scholes equation translates the problem of pricing a European option to solving a stochastic partial differential equation (stochastic PDE) with a certain final condition. In this section we will see how the PDE is derived and what its solution is.

2.4.1 Derivation of the Black-Scholes PDE

The Black-Scholes model is based on the following assumptions:

1. There is a constant risk-free interest rate r at which we can borrow and lend money in any amount.
2. The asset price follows a geometric brownian motion with constant drift and volatility.
3. The stock pays no dividend.
4. There are no arbitrage possibilities.
5. It is possible to buy and sell every amount of stock, including fractions and short selling.
6. There are no transaction costs on trading.

Since we assume that the asset price follows a geometric brownian motion we see that

$$\mathrm{d}S(t) = rS(t)\ \mathrm{d}t + \sigma\ \mathrm{d}W^{\mathbb{P}}(t)$$

Consider the money-savings account $M(t)$ with $\mathrm{d}M(t) = rM(t)\ \mathrm{d}t$ with $M(t_0) = 1$. We know that the discounted value of options and assets are martingales. So

$$V(t,S) = M(t)\mathbb{E}^{\mathbb{Q}}\left[\frac{V(T,S)}{M(T)}\middle|\mathcal{F}(t_0)\right].$$

We assume there exists a continously differentiable function $\Pi_V \equiv \Pi_V(t,S)$ such that

$$\Pi_V(t,S) = \mathbb{E}^{\mathbb{Q}}\left[\frac{V(T,S)}{M(T)}\middle|\mathcal{F}(t_0)\right] = \frac{V(t,S)}{M(t)}.$$

Since the discounted option value is a martingale we see that:

$$\mathrm{d}\Pi_V = \mathrm{d}\left(\frac{V}{M}\right) = \frac{1}{M}\,\mathrm{d}V - \frac{V}{M^2}\,\mathrm{d}M = \frac{1}{M}\,\mathrm{d}V - r\frac{V}{M}\,\mathrm{d}t.$$

Looking at an infinitely small change in V we find using Itô's lemma:

$$\mathrm{d}V = \left(\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t + \sigma S\,\mathrm{d}W^{\mathbb{Q}}.$$

This gives us

$$\begin{aligned}\mathrm{d}\Pi_V &= \frac{1}{M}\left(\left(\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2}\right)\mathrm{d}t + \sigma S\,\mathrm{d}W^{\mathbb{Q}}\right) - r\frac{V}{M}\,\mathrm{d}t \\ &= \frac{1}{M}\left(\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV\right)\mathrm{d}t + \frac{\sigma S}{M}\,\mathrm{d}W^{\mathbb{Q}}.\end{aligned}$$

Since Π_V must be a martingale, the dynamics of Π_V cannot contain any $\mathrm{d}t$ terms(since the expectation of a martingale is not time dependant). Therefore

$$\frac{\partial V}{\partial t} + rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} - rV = 0.$$

This is known as the Black-Scholes PDE. For the solution to be well-posed we require a final condition, which logically is the payoff function at time T.

2.4.2 Solution to the Black-Scholes PDE

The solution to the Black-Scholes PDE is known analytically for European call and put options. From an economical perspective it is easy to see that the value of an asset must always be greater than or equal to zero, and the option satisfies its dynamics from the moment of being issued to the expiration time. This means we have $S \in [0, \infty)$ and $t \in [0, T]$. For a call option we know that

$$V_C(S, T) = \max(S(T) - K, 0).$$

Furthermore, if the asset price is zero, thus the asset has no value, this means the option has no value. Thus

$$V_C(0, t) = 0, \quad \forall t \in [0, T].$$

If we now have a very large asset price, so large that it is extremely unlikely that it will go out-of-the money, we can see that

$$V_C(S, t) = S - K \cdot e^{-rT}, \text{for } S \gg K.$$

This holds because the option value is the discounted expected payoff. These 3 conditions make sure a unique solution exists for the value of a call option:

$$V_C(S, t) = SF_{\mathcal{N}(0,1)}(d_1) - Ke^{-r(T-t)}F_{\mathcal{N}(0,1)}(d_2).$$

Where $\mathcal{N}(0, 1)$ is the standard normal distribution and F denotes the Cumulative Distribution Function (CDF).

$$d_1 = \frac{\log(\frac{S}{K}) + (r - \frac{1}{2}\sigma^2)(T - t)}{\sigma\sqrt{T - t}},$$

$$d_2 = d_1 - \sigma\sqrt{(T - t)}.$$

Using put-call parity we can now also determine the price of a European put option:

$$\begin{aligned}V_P(S, t) &= V_C(S, t) + Ke^{-r(T-t)} - S \\ &= Ke^{-r(T-t)}(1 - F_{\mathcal{N}(0,1)}(d_2)) + S(F_{\mathcal{N}(0,1)}(d_1) - 1) \\ &= Ke^{-r(T-t)}F_{\mathcal{N}(0,1)}(-d_2) - SF_{\mathcal{N}(0,1)}(-d_1).\end{aligned}$$

2.4.3 Barrier options under the Black-Scholes model

In the previous section a formula was derived to price European options, options whose payoff depend on the strike price and the asset price at expiry. There is a large variety of other types of options, referred to as **exotic** options. There are usually two features that make an option more complex:

1. Path dependency: the way in which the payoff depends on the asset path $S(t)$ for $t \in [0, T]$,
2. Early excercise is allowed.

For these exotic options we usually cannot find exact expressions for the option value, which means we must approximate the price. This is usually done with a Monte Carlo method and simulating many asset paths or numerically approximating the solution to the Black-Scholes PDE.

A barrier option is an option whose payoff switches either on or off if the asset price crosses a pre-determined level B. A down-and-out call option for example has the same payoff as a regular European call option, **unless** the asset price goes below a pre-determined barrier $B < S_0$ at any time in $[0, T]$.

For the value of a down-and-out call option we now write $V_{C,B}(S,t)$. If$B < K$ we can express the solution to the Black-Scholes PDE analytically by:

$$V_{C,B}(S,t) = V_C(S,t) - \left(\frac{S}{B}\right)^{1-\frac{2r}{\sigma^2}} V_C\left(\frac{B^2}{S}, t\right).$$

It is logical that the down-and-out call option is worth less than a regular European call option and this is reflected in the above expression. Other types of barrier options, such as an down-and-in put option, can be written in terms of call and put options as well.

2.5 Weaknesses of the Black-Scholes model

The Black-Scholes model is based on a number of assumptions, which do not always hold in the real world as can be seen in financial markets as can be read in [4]. Especially the assumption that the volatility is constant over time is not always realistic. Furthermore in markets there is evidence of a so called volatility smile/skew which will be discussed later. This has been shown extensively and this is why banks and other financial institutions will usually sell their derivatives for a different price than the Black-Scholes price.
Even though the Black-Scholes model has weaknesses it is widely used, because it is easy to calculate, a good approximation for further calibration and reversible. The reversibility will be discussed in the implied volatility section (3.2).

3 Implied volatility and the local volatility model

In this section different choices of volatility are presented and implied volatility will be explained. Futhermore the local volatility model will be derived and after this an alternative representation of the local volatility will be derived, which is useful since it puts the local volatility in terms of the implied volatility. The local volatility model is used to overcome the weakness of the Black-Scholes model of not allowing different volatilities for different strike prices at the same time to maturity.

3.1 Time dependent volatility

Suppose we want to extend the Black-Scholes model with a volatility parameter that is not constant, but time dependent. We now get:

$$\mathrm{d}S(t) = r(t)S(t)\ \mathrm{d}t + \sigma(t)S(t)\ \mathrm{d}W(t)$$

If we now take two stochastic processes $X(t), Y(t)$, that are constructed with the same Wiener process:

$$\mathrm{d}X(t) = \left(r - \frac{1}{2}\sigma^2(t)\right)\ \mathrm{d}t + \sigma(t)\ \mathrm{d}W(t), \qquad \mathrm{d}Y(t) = \left(r - \frac{1}{2}\sigma_Y\right)\ \mathrm{d}t + \sigma_Y\ \mathrm{d}W(t)$$

where σ_Y denotes a constant volatility for the process Y. If we look at the expectations we see:

$$\mathbb{E}\left[X(T)\right] = X_0 + \int_0^T \left(r - \frac{1}{2}\sigma^2(t)\right)\ \mathrm{d}t, \quad \mathbb{E}\left[Y(T)\right] = Y_0 + \left(r - \frac{1}{2}\sigma_Y^2\right)T.$$

We now look at the variances:

$$\begin{aligned}
&\mathbb{V}\mathrm{ar}\left[Y(T)\right] = \sigma_Y^2 T \\
&\mathbb{V}\mathrm{ar}\left[X(T)\right] = \mathbb{E}\left[X^2(T)\right] - \left(\mathbb{E}\left[X(T)\right]\right)^2 = \mathbb{E}\left[\int_0^T \sigma(t)\ \mathrm{d}W(t)\right]^2.
\end{aligned}$$

We know from Itô's isometry that

$$\mathbb{E}\left[\int_0^T g(t)\ \mathrm{d}W(t)\right]^2 = \int_0^T \mathbb{E}\left[g^2(t)\right]\ \mathrm{d}t.$$

Thus

$$\mathbb{E}\left[\int_0^T \sigma(t)\ \mathrm{d}W(t)\right]^2 = \int_0^T \mathbb{E}\left[\sigma^2(t)\right]\ \mathrm{d}t = \int_0^T \sigma^2(t)\ \mathrm{d}t.$$

So we find:

$$\mathbb{V}\mathrm{ar}\left[X(T)\right] = \int_0^T \sigma^2(t)\ \mathrm{d}t.$$

The variances of $X(T)$ and $Y(T)$ are equal if

$$\begin{aligned}
\mathbb{V}\mathrm{ar}\left[X(T)\right] &= \int_0^T \sigma^2(t)\ \mathrm{d}t = \sigma_Y^2 T = \mathbb{V}\mathrm{ar}\left[Y(T)\right] \\
\sigma_Y^2 &= \frac{1}{T}\int_0^T \sigma^2(t)\ \mathrm{d}t \\
\sigma_Y &= \sqrt{\frac{1}{T}\int_0^T \sigma^2(t)\ \mathrm{d}t}.
\end{aligned}$$

If we now take $X_0 = Y_0$ we see that $\mathbb{E}[X(T)] = \mathbb{E}[Y(T)]$. This ensures the first two moments of these stochastic processes match. This is enough to ensure equality for European options under the Black-Scholes model. Thus for European options when considering a model with a volatility parameter that only depends on time, we can find the matching volatility for the Black-Scholes model which gives these options the same price at every time.

3.2 Implied volatility

Suppose we have a model that computes or approximates the price of an option V. For this option the strike price, the asset price at this moment, the time to maturity and the risk free interest rate are known. With the Black-Scholes model we can determine the option price using these parameters and the volatility. Therefore, we can find the volatility for which, under the Black-Scholes model, the option value is equal to V. Thus we seek the volatility σ which makes the option value under the Black-Scholes model equal to the option price given by a different model. This is what we call the implied volatility. The same can of course be done for option prices quoted in the market. It is actually quite normal to quote options in terms of their Black-Scholes implied volatility instead of their price.
To determine the Black-Scholes implied volatility (from now on shortened simply to implied volatility) belonging to the price of a call option in the market $V_{C,\text{market}}$, we find the value σ_{imp} such that the following equation holds:

$$V_{C,\text{market}} = V_C(t_0, S, T, K, \sigma_{\text{imp}}, r)$$

For a time-dependent volatility parameter as in the previous section we see that the implied volatility is simply:

$$\sigma_{\text{imp}} = \sqrt{\frac{1}{T} \int_0^T \sigma^2(t) \, \mathrm{d}t}.$$

When we look at options on the same stock with the same expiry date but different strike prices in the market, we often see a so called volatility smile or skew. This means that the volatility is not constant for different strike prices, which is not possible under the Black-Scholes model or with a volatility parameter that is only time-dependent. This gives us a reason to look for a different model which allows the volatility parameter to differ for different strike prices.

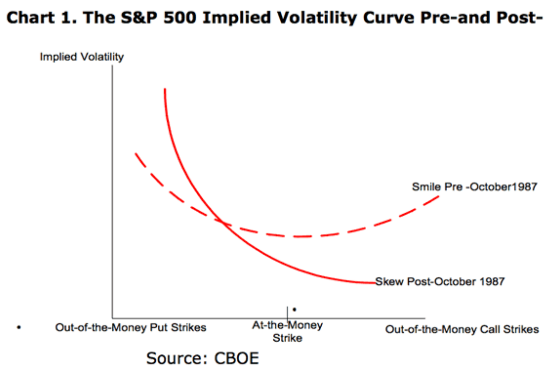


Figure 3: Example of the volatility smile

We will now look into more complicated models. Note that more complicated models often require calibration to the market and it is important that these models are able to replicate the values for European options.

3.3 Deriving the local volatility model

We will now determine the price of a call option with **local volatility**. The local term is an indicator that the volatility is a function of the stock. The classical model for local volatility is given by:

$$\mathrm{d}S(t) = S(t)r\ \mathrm{d}t + S(t)\bar{\sigma}(t, S(t))\ \mathrm{d}W(t), \quad S(t_0) = S_0.$$

Where r represents the constant interest rate.
For a call option $V(t, S)$ when assuming no arbitrage, we get (due to Itô's lemma) the following PDE:

$$\frac{\partial V}{\partial t} + \frac{1}{2}\bar{\sigma}(t, S)S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0$$
$$V(T, S) = \max(S(T) - K, 0).$$

This is also known as the Kolmogorov-backwards equation, which describes a probability distribution at a time $s < t$. We will link this to the Fokker-Planck PDE later in this section.
First we derive the stock density f with the help of call and put options. We then use this to find the function $\bar{\sigma}(t, S)$ in terms of market quotes. If we have a standard European call option $V_C(t_0, S_0, K)$ and we assume the interest rate is constant we know the value is:

$$\begin{aligned} V_C(t_0, S_0, K) &= e^{-r(T-t_0)}\mathbb{E}^{\mathbb{Q}}\left[\max(S(T) - K, 0)|\mathcal{F}(t_0)\right] \\ &= e^{-r(T-t_0)}\int_K^\infty (x - K)f(T, x)\ \mathrm{d}x. \end{aligned}$$

Where f is the density of the stock process. If we now write $F(t_0, T) = e^{-r(T-t_0)}$ and differentiate the option with respect to the strike price K we find:

$$\begin{aligned} \frac{\partial}{\partial K}V_C(t_0, S_0, K) &= F(t_0, T)\frac{\partial}{\partial K}\int_K^\infty (x - K)f(T, x)\ \mathrm{d}x \\ &= F(t_0, T)\int_K^\infty \frac{\partial}{\partial K}(x - K)f(T, x)\ \mathrm{d}x = -F(t_0, T)\int_K^\infty f(T, x)\ \mathrm{d}x. \end{aligned}$$

Differentiating again with respect to K gives us:

$$\frac{\partial^2}{\partial K^2}V_C(t_0, S_0, K) = F(t_0, T)f(T, K).$$

Thus we can describe the stock density f in terms of derivatives of call options (or put options).
If we differentiate with respect to the time to maturity T we find:

$$\begin{aligned} \frac{\partial}{\partial T}V_C(t_0, S_0) &= \frac{\partial}{\partial T}\left(e^{-r(T-t_0)}\int_K^\infty (x - K)f(T, x)\ \mathrm{d}x\right) \\ &= -rV_C(t_0, S_0) + F(t_0, T)\int_K^\infty (x - K)\frac{\partial}{\partial T}f(T, x)\ \mathrm{d}x. \end{aligned}$$

We can describe the evolution of a probability density with the help of the Fokker-Planck PDE which usually models the probablility density function by a Dirac delta function δ at starting time and describes the evolution of this probability density in time.

Theorem 2 (Fokker-Planck PDE)**.** *The transition density $f(t, S) \equiv f(t_0, S_0, t, S)$ associated to the SDE:*

$$\mathrm{d}S(t) = \bar{\mu}(t, S)\ \mathrm{d}t + \bar{\sigma}(t, S)\ \mathrm{d}W(t), \quad S(0) = S_0$$

for $S(t)$ with $t_0 \leq t \leq T$ satisfies the Fokker-Planck PDE:

$$\frac{\partial}{\partial t}f(t, S) + \frac{\partial}{\partial S}\bar{\mu}(t, S)f(t, S) + \frac{1}{2}\frac{\partial^2}{\partial S^2}\bar{\sigma}(t, S)S^2 f(t, S) = 0$$
$$f(t_0, S_0) = \delta(S_0).$$

The idea behind the local volatility model is to match the density of our asset price with the Fokker-Planck PDE. When considering the Fokker-Planck PDE for our problem we can rewrite this to:

$$\frac{\partial}{\partial t}f(t,S) = -r\frac{\partial}{\partial S}Sf(t,S) + \frac{1}{2}\frac{\partial^2}{\partial S^2}\bar{\sigma}(t,S)S^2 f(t,S).$$

We use this to rewrite the integral found in the derivative of the call option with respect to T and see:

$$\int_K^\infty (x-K)\frac{\partial}{\partial T}f(T,x)\ \mathrm{d}x = \int_K^\infty (x-K)\left[-\frac{\partial}{\partial x}rxf(T,x) + \frac{1}{2}\cdot\frac{\partial^2}{\partial x^2}\bar{\sigma}(t,x)x^2 f(t,x)\right]\ \mathrm{d}x \quad (1)$$

$$= -r\int_K^\infty (x-K)\frac{\partial}{\partial x}xf(T,x)\ \mathrm{d}x + \quad (2)$$

$$\frac{1}{2}\int_K^\infty (x-K)\frac{\partial^2}{\partial x^2}\bar{\sigma}^2(T,x)x^2 f(T,x)\ \mathrm{d}x. \quad (3)$$

Assuming $\lim_{x\to\infty} xf(T,x) = 0$ (thus the density decays faster than x grows) we find:

$$\begin{aligned}\int_K^\infty (x-K)\frac{\partial}{\partial x}xf(T,x)\ \mathrm{d}x &= (x-K)xf(T,x)|_{x=K}^\infty - \int_K^\infty xf(T,x)\ \mathrm{d}x \\ &= -\int_K^\infty xf(T,x)\ \mathrm{d}x.\end{aligned}$$

We use the second derivative of the call option to express this in terms of the density:

$$\begin{aligned}\int_K^\infty (x-K)\frac{\partial}{\partial x}xf(T,x)\ \mathrm{d}x &= \frac{-1}{F(t_0,T)}\int_K^\infty x\frac{\partial^2}{\partial x^2}V_C(T,x)\ \mathrm{d}x \\ &= \frac{-1}{F(t_0,T)}\left[x\frac{\partial}{\partial x}V_C(T,x)\bigg|_K^\infty - \int_K^\infty \frac{\partial}{\partial x}V_C(T,x)\ \mathrm{d}x\right] \\ &= \frac{1}{F(t_0,T)}\left[K\frac{\partial}{\partial K}V_C(T,K) - V_C(T,K)\right].\end{aligned}$$

The second integral in (3) can likewise be written as:

$$\begin{aligned}\int_K^\infty (x-K)\frac{\partial^2}{\partial x^2}\bar{\sigma}^2(T,x)x^2 f(T,x)\ \mathrm{d}x &= \bar{\sigma}^2(T,K)K^2 f(T,K) \\ &\equiv \frac{1}{F(t_0,T)}\bar{\sigma}^2(T,K)K^2\frac{\partial^2}{\partial K^2}V_C(T_0,T,S_0,K).\end{aligned}$$

So we see that

$$\begin{aligned}\frac{\partial}{\partial T}V_C(t_0,S_0) &= -rV_C + F(t_0,T)\left[\frac{-r}{F(t_0,T)}K\frac{\partial V_C}{\partial K} + \frac{rV_C}{F(t_0,T)} + \frac{1}{2}\frac{1}{F(t_0,T)}\bar{\sigma}^2(T,K)K^2\frac{\partial^2 V_C}{\partial K^2}\right] \\ &= -rV_C + \left[-rK\frac{\partial V_C}{\partial K} + rV_C + \frac{1}{2}\bar{\sigma}^2(T,K)K^2\frac{\partial^2 V_C}{\partial K^2}\right].\end{aligned}$$

From this it is easy to rewrite this to express $\bar{\sigma}^2$ in terms of the derivatives of call options:

$$\bar{\sigma}^2(T,K) = \frac{\frac{\partial}{\partial T}V_C(t_0,S_0,K) + rK\frac{\partial}{\partial K}V_C(t_0,S_0,K)}{\frac{1}{2}K^2\frac{\partial^2}{\partial K^2}V_C(t_0,S_0,K)}. \quad (4)$$

So the volatility $\bar{\sigma}(T,K)$ is described by the data available from the market and perfectly fits these datapoints, while not introducing any parametrization of the volatility, which means we do not have to perform a calibration procedure.

3.4 Local volatility in terms of implied volatility

We see that the formula for the local volatility is expressed in terms of derivatives of call option prices. However, in practice not all derivatives can be obtained from the market data, and thus we have to approximate these derivatives. We will now derive the local volatility in terms of the Black-Scholes implied volatility. From the Black-Scholes model we know that for a European call option the value is given by:

$$V_C(t,K) = S_0 F_{\mathcal{N}(0,1)}(d_1) - Ke^{-r(t-t_0)} F_{\mathcal{N}(0,1)}(d_2),$$

where

$$d_1 = \frac{\log\left(\frac{S_0}{K}\right) + \left(r + \frac{1}{2}\bar{\sigma}^2_{\text{imp}}(t,K)\right)(t-t_0)}{\bar{\sigma}_{\text{imp}}(t,K)\sqrt{t-t_0}}, \quad d_2 = d_1 - \bar{\sigma}_{\text{imp}}(t,K)\sqrt{t-t_0}$$

and $\bar{\sigma}_{\text{imp}}(t,K)$ is the Black-Scholes implied volatility with strike K at time t. We now use a transformation of variables:

$$y = \log\left(\frac{K}{P(t_0,t)}\right) = \log\left(\frac{K}{S_0}\right) - r(t-t_0), \quad \omega = \bar{\sigma}^2_{\text{imp}}(t,K)(t-t_0),$$

where P is the usual notation for the forward price: $P = S_0 e^{r(t-t_0)}$. We will now define the price of the call option in terms of a function $c(y,\omega)$:

$$c(y,\omega) = S_0\left[F_{\mathcal{N}(0,1)}(d_1) - e^y F_{\mathcal{N}(0,1)}(d_2)\right] = V_C(t,K).$$

We can also express d_1 and d_2 in terms of these variables:

$$d_1 = -\frac{y}{\sqrt{\omega}} + \frac{1}{2}\sqrt{\omega}, \quad d_2 = d_1 - \sqrt{\omega}.$$

When differentiating to K we get:

$$\frac{\partial V_C}{\partial K} = \frac{\partial c}{\partial y}\frac{\partial y}{\partial K} + \frac{\partial c}{\partial \omega}\frac{\partial \omega}{\partial K} = \frac{\partial c}{\partial y}\frac{1}{K} + \frac{\partial c}{\partial \omega}\frac{\partial \omega}{\partial K}.$$

The second derivative with respect to K:

$$\frac{\partial^2 V_C}{\partial K^2} = \frac{1}{K^2}\left(\frac{\partial^2 c}{\partial y^2} - \frac{\partial c}{\partial y}\right) + \frac{2}{K}\frac{\partial \omega}{\partial K}\frac{\partial^2 c}{\partial \omega \partial y} + \frac{\partial^2 \omega}{\partial K^2}\frac{\partial c}{\partial \omega} + \left(\frac{\partial \omega}{\partial K}\right)^2 \frac{\partial^2 c}{\partial \omega^2}.$$

The derivative with respect to $(t-t_0)$:

$$\frac{\partial V_C}{\partial (t-t_0)} = -r\frac{\partial c}{\partial y} + \frac{\partial c}{\partial \omega}\frac{\partial \omega}{\partial t}.$$

If we now substitute these into (4) we find:

$$\bar{\sigma}^2(T,K) = \frac{\frac{\partial c}{\partial \omega}\frac{\partial \omega}{\partial (t-t_0)} + rK\frac{\partial c}{\partial \omega}\frac{\partial \omega}{\partial K}}{\frac{1}{2}\left(\frac{\partial^2 c}{\partial y^2} - \frac{\partial c}{\partial y}\right) + K\frac{\partial \omega}{\partial K}\frac{\partial^2 c}{\partial \omega \partial y} + \frac{1}{2}K^2\left[\frac{\partial^2 \omega}{\partial K^2}\frac{\partial c}{\partial \omega} + \left(\frac{\partial \omega}{\partial K}\right)^2\frac{\partial^2 c}{\partial \omega^2}\right]}. \tag{5}$$

We can simplify this by writing out the derivatives of c. Note that for a standard normally distributed stochastic variable Z the following holds:

$$F'_Z(x) = f(x)x'$$
$$f'_Z(x) = -xf_Z(x)x',$$

Furthermore:

$$\begin{aligned} f_Z(d_1) &= \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(d_2+\sqrt{\omega})^2} = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}(d_2^2+\omega+2d_2\sqrt{\omega})} \\ &= f_Z(d_2) e^{-d_2\sqrt{\omega}-\frac{1}{2}\omega} = f_Z(d_2) e^{-\left(\frac{-y}{\sqrt{\omega}}-\frac{1}{2}\sqrt{\omega}\right)\sqrt{\omega}-\frac{1}{2}\omega} = f_Z(d_2)e^y. \end{aligned}$$

We now see:

$$\begin{aligned} \frac{\partial c}{\partial \omega} &= S_0 \left[f_Z(d_1)\frac{\partial d_1}{\partial \omega} - e^y f_Z(d_2)\frac{\partial d_2}{\partial \omega}\right] = S_0 \left[f_z(d_2)e^y\frac{\partial d_1}{\partial \omega} - e^y f_Z(d_2)\frac{\partial d_2}{\partial \omega}\right] \\ &= S_0 \left[f_z(d_2)e^y\left(\frac{\partial d_2}{\partial \omega} + \frac{1}{2\sqrt{\omega}}\right) - e^y f_Z(d_2)\frac{\partial d_2}{\partial \omega}\right] \\ &= S_0 \left[e^y f_z(d_2)\frac{1}{2\sqrt{\omega}}\right] = \frac{1}{2}S_0 e^y \left[f_z(d_2)\frac{1}{\sqrt{\omega}}\right]. \end{aligned}$$

We use this to determine the second derivative:

$$\begin{aligned} \frac{\partial^2 c}{\partial \omega^2} &= \frac{1}{2}S_0 e^y \left[f_Z'(d_2)\frac{1}{\sqrt{\omega}} - \frac{1}{2}f_Z(d_2)\frac{1}{\sqrt{\omega^3}}\right] = \frac{1}{2}S_0 e^y \left[-d_2 f_Z(d_2)\frac{\partial d_2}{\partial \omega}\frac{1}{\sqrt{\omega}} - \frac{1}{2}f_Z(d_2)\frac{1}{\sqrt{\omega^3}}\right] \\ &= \frac{1}{2}S_0 e^y f_Z(d_2)\frac{1}{\sqrt{\omega}}\left[-d_2\frac{\partial d_2}{\partial \omega} - \frac{1}{2\omega}\right] = \frac{\partial c}{\partial \omega}\left[\left(\frac{y}{\sqrt{\omega}}+\frac{1}{2}\sqrt{\omega}\right)\left(\frac{1}{2}\frac{y}{\sqrt{\omega^3}} - \frac{1}{4\omega}\right) - \frac{1}{2\omega}\right] \\ &= \frac{\partial c}{\partial \omega}\left[-\frac{1}{8} + \frac{y}{4\omega} + \frac{y^2}{2\omega^2} - \frac{1}{4\omega} - \frac{1}{2\omega}\right] = \frac{\partial c}{\partial \omega}\left[-\frac{1}{8} + \frac{y^2}{2\omega^2} - \frac{1}{2\omega}\right]. \end{aligned}$$

Now we look at:

$$\begin{aligned} \frac{\partial^2 c}{\partial \omega \partial y} &= S_0 \frac{1}{2\sqrt{\omega}}\frac{\partial}{\partial y}\left(e^y f_Z(d_2)\right) = S_0\frac{1}{2\sqrt{\omega}}\left[e^y f_Z(d_2) + e^y f_Z'(d_2)\right] \\ &= S_0\frac{1}{2\sqrt{\omega}}\left[e^y f_Z(d_2) - e^y d_2 f_Z(d_2)\frac{\partial d_2}{\partial y}\right] = S_0\frac{1}{2\sqrt{\omega}}e^y f_Z(d_2)\left[1 - d_2\frac{\partial d_2}{\partial y}\right] \\ &= \frac{\partial c}{\partial \omega}\left[1 - d_2\frac{-1}{\sqrt{\omega}}\right] = \frac{\partial c}{\partial \omega}\left[1 - \frac{y}{\omega} - \frac{1}{2}\right] = \frac{\partial c}{\partial \omega}\left[\frac{1}{2} - \frac{y}{\omega}\right] \end{aligned}$$

and we see:

$$\begin{aligned} \frac{\partial c}{\partial y} &= S_0\left[f_Z(d_1)\frac{\partial d_1}{\partial y} - e^y F_Z(d_2) - e^y f_Z(d_2)\frac{\partial d_2}{\partial y}\right] \\ &= S_0 e^y\left[f_Z(d_2)\frac{\partial d_1}{\partial y} - F_Z(d_2) - f_Z(d_2)\frac{\partial d_2}{\partial y}\right] \\ &= -S_0 e^y F_Z(d_2), \end{aligned}$$

giving us:

$$\begin{aligned} \frac{\partial^2 c}{\partial y^2} &= -S_0\left[e^y F_Z(d_2) + e^y f_Z(d_2)\frac{\partial d_2}{\partial y}\right] \\ &= -S_0 e^y F_Z(d_2) + S_0 e^y f_Z(d_2)\frac{1}{\sqrt{\omega}} \\ &= \frac{\partial c}{\partial y} + 2\frac{\partial c}{\partial \omega}. \end{aligned}$$

Substituting this into (5) gives us the following result:

$$\bar{\sigma}^2(t,K) = \frac{\frac{\partial \omega}{\partial t} + rK\frac{\partial \omega}{\partial K}}{1 + K\frac{\partial \omega}{\partial K}\left(\frac{1}{2} - \frac{y}{\omega}\right) + \frac{1}{2}K^2\frac{\partial^2\omega}{\partial K^2} + \frac{1}{2}K^2\left(\frac{\partial \omega}{\partial K}\right)^2\left(-\frac{1}{8} - \frac{1}{2\omega} + \frac{y^2}{2\omega^2}\right)}. \tag{6}$$

Putting the remaining derivatives in terms of the implied volatility yields:

$$\begin{aligned}
\frac{\partial \omega}{\partial t} &= \bar{\sigma}_{\text{imp}}^2 + 2\bar{\sigma}_{\text{imp}} \cdot (t-t_0)\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial t} \\
\frac{\partial \omega}{\partial K} &= 2\bar{\sigma}_{\text{imp}} \cdot (t-t_0)\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K} \\
\frac{\partial^2 \omega}{\partial K^2} &= 2(t-t_0)\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + 2\bar{\sigma}_{\text{imp}} \cdot (t-t_0)\frac{\partial^2 \bar{\sigma}_{\text{imp}}}{\partial K^2}.
\end{aligned}$$

If we substitute these into (6) we can express the local volatility in terms of the implied volatility and find:

$$\bar{\sigma}^2(t,K) = \frac{\bar{\sigma}_{\text{imp}}^2 + 2\bar{\sigma}_{\text{imp}}(t-t_0)\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial t} + r \cdot K \frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)}{\left(1 - \frac{K \cdot y}{\sigma_{\text{imp}}}\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\bar{\sigma}_{\text{imp}}(t-t_0)\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K} - \frac{1}{4}K\bar{\sigma}_{\text{imp}}(t-t_0)\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\frac{\partial^2 \bar{\sigma}_{\text{imp}}}{\partial K^2}\right)},$$

where $\sigma_{\text{imp}} = \sigma_{\text{imp}}(t,K)$ and $y = \log\left(\frac{K}{S_0 e^{r(t-t_0)}}\right)$. Transforming to $\tau = T - t$ we find:

$$\bar{\sigma}^2(\tau,K) = \frac{\bar{\sigma}_{\text{imp}}^2 + 2\tau\bar{\sigma}_{\text{imp}}\left(-\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial \tau} + r \cdot K \frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)}{\left(1 - \frac{K \cdot \log\left(\frac{K}{S_0 e^{r\tau}}\right)}{\sigma_{\text{imp}}}\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K} - \frac{1}{4}K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial \bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\frac{\partial^2 \bar{\sigma}_{\text{imp}}}{\partial K^2}\right)}.$$

Note that the implied volatility is in general not equal to the local volatility unless the implied volatility is constant.

The local volatility model enables a volatility smile/skew to exist, which is implied by the market. For European options and barrier options this is enough to be able to compute their prices correctly. However when looking at options that allow you to excercise the option early, the local volatility model falls short. There is no control over the forward volatility which is essential for these kinds of options. A forward start option is an option for which the starting price and starting time is set in the future. Note that models that are more complicated than the Black-Scholes model are derived for more complex options than European options, but they should agree on the price on a European option. These more complicated processes are better at capturing the dynamics of asset price movements which are needed for exotic options.

4 The Collocating Local Volatility Model

The Collocating Local Volatility model is introduced in [1] by Grzelak. Before we delve into the Collocating Local Volatility(CLV) model, some background on stochastic collocation is necessary, which will be given in the following section. Afterwards an overview of how to construct the CLV model is given. This is followed by a derivation of a PDE for this model. The CLV model allows for control over the forward-start implied volatility while ensuring a good fit to the market data.

4.1 Stochastic collocation

The idea behind stochastic collocation is to approximate a stochastic variable Y that is computationally expensive to compute with a stochastic variable X which is easy to compute. We determine a function g such that $g(X) \approx Y$. This is done by inverting the cumulative distribution function (CDF) of Y on a few points, the so called collocation points. On these points we compute a few "expensive" inversions:

$$y_n = F_Y^{-1}(F_X(\bar{x}_n)).$$

We thus seek a function $g(\cdot) = F_Y^{-1}(F_X(\cdot))$ such that $F_X(x) = F_Y(g(x))$, giving us $Y = g(X)$. We can therefore generate samples from Y without having to compute the expensive inversions of the CDF but rather use our mapping function g and the easy to compute X. This is possible because $F_X(X)$ and $F_Y(Y)$ are uniformly distributed on $[0, 1]$: take $u \in [0, 1]$. Now if we look at $\mathbb{P}[F_Y(Y) \leq u] = \mathbb{P}[Y \leq F_Y^{-1}(u)] = F_Y(F_Y^{-1}(u)) = u$ we see that this is exactly the CDF of the uniform distribution, therfore $F_Y(Y)$ is uniformly distributed. For $F_X(X)$ this goes analogous. Therefore we have $F_Y(Y) \stackrel{\mathrm{d}}{=} F_X(X)$ thus $Y = F_Y^{-1}(F_X(X))$. Note that although both $F_Y(Y)$ and $F_X(X)$ are uniformly distributed, this does not mean that X and Y have the same distribution.

4.2 Constructing the Collocating Local Volatility model

The CLV method is constructed in the following manner. Consider an underlying asset $S(t)$ and a process $X(t)$ wich will be used to generate the process $S(t)$. We can choose $X(t)$ freely but require the moments to exist. We relate $X(t)$ and $S(t)$ by:

$$S(t) = g(t, X(t)),$$

where g is a deterministic function. We want to build the function $g(t, x)$ such that the volatilities implied by the market correspond to those generated by the model.

$$\begin{aligned} S(t) &= g(t, X(t)) \\ \mathrm{d}X(t) &= \mu(X(t))\ \mathrm{d}t + \sigma(X(t))\ \mathrm{d}W^{\mathbb{Q}}(t) \end{aligned}$$

with $X(t_0) = S(t_0)$. Although it is not necessary to have $X(t_0) = S(t_0)$ we will use this because if we let $t \to 0$ for $g(t, x)$ this will be more practical. To approximate this function g we consider the arbitrary process $X(t)$ with given marginal CDF and finite moments. We build a projection function $g(t, x)$ such that for a number of collocation points $x_j(T_i) = x_{i,j}$ (once again chosen by the method in [7]) we have that:

$$F_{X(T_i)}(x_{i,j}) = F_{\hat{S}(T_i)}(g(T_i, x_{i,j})) = F_{\hat{S}(T_i)}(s_{i,j})$$

Note that we can obtain the collocation values $s_{i,j}$ by computing $F_{\hat{S}(T_i)}^{-1}\left(F_{X(T_i)}(x_{i,j})\right)$.

The collocation points $x_{i,j}$ are chosen such that the moments match. This method is shown in [7], but it boils down to computing the Gauss quadrature points, which are the zeros of the orthogonal polynomial which it belongs to. This leads to

$$\mathbb{E}\left[\hat{S}^k(T_i)\right] = \mathbb{E}\left[g^k(T_i, X(T_i))\right] + \varepsilon_{i,N}$$

for $i \in \{1, \ldots, M\}$ where $\varepsilon_N \to 0$ quadratically for $N \to \infty$.
Now that we have the collocation points, we have to ensure continuity of the function g. We want to be able to simulate the stock prices between the expiries at which we know the distribution. Suppose we want to know the value $g(t, X(t))$, where $t \in (T_i, T_{i+1})$. We then determine the collocation points x_j. For a random normal variable this is easily done by taking $x_j(t) = \mathbb{E}[X(t)] + \sqrt{\mathbb{V}\mathrm{ar}[X(t)]}\; x_j^{\mathcal{N}(0,1)}$, where $x_j^{\mathcal{N}(0,1)}$ is the j$^{\text{th}}$ collocation point for a standard normal variable.
We also need to look at the distribution $F_{\hat{S}(T_i)}$. Let us consider a set of expiry dates $T_1, \ldots, T_M$. We calibrate the function g to these expiries. Assume a set of implied volatilities is given at these expiry dates and they are free of arbitrage. Under the risk-neutral measure we see that for a European option

$$\begin{aligned} V_C(S_0, T_i, K) &= e^{-rT_i}\, \mathbb{E}^{\mathbb{Q}}\left[\max(S(T_i) - K, 0) | \mathcal{F}(t_0)\right] \\ &= e^{-rT_i} \int_{-\infty}^{\infty} \max(x - K, 0) f_{\hat{S}(T_i)}(x)\ \mathrm{d}x \\ &= e^{-rT_i} \int_{K}^{\infty} (x - K) f_{\hat{S}(T_i)}(x)\ \mathrm{d}x. \end{aligned}$$

Differentiating this with respect to K yields

$$\left.\frac{\partial V_C(S_0, T_i, y)}{\partial K}\right|_{y=K} = e^{-rT_i}\left[-F_{\hat{S}(T_i)}(\infty) + F_{\hat{S}(T_i)}(K)\right]$$

So we see that

$$F_{\hat{S}(T_i)}(y) = e^{rT_i}\left.\frac{\partial V_C(S_0, T_i, y)}{\partial K}\right|_{y=K} + 1.$$

We numerically approximate the inverse to obtain the collocation values $s_{i,j} = F^{-1}_{\hat{S}(T_i)}\left(F_{X(T_i)}(x_{i,j})\right)$. When we have these we have triplets $\{T_i, x_{i,j}, s_{i,j}\}$ we use these to approximate the function g by interpolation. Note that g should be monotonic thus we need to use a monotonic form of interpolation. Note that the only thing that influences the collocation points $x_{i,j}$ is the choice of the process X, thus this is not influenced by any views on the asset price S.

4.3 Construction of the PDE

Using this model we will now derive a stochastic PDE such that we can use the FDM.
The value of an option on an underlying stock is a function of the stock price and time. We have a way of expressing the option value at maturity in terms of the stock price and time (the payoff function). Thus we can express for an option $V(t, S(t)) = V(t, g(t, X(t))) = f(t, X(t))$ in a function f which depends on X and time and with the final condition known: $V(T, S(T)) = \Phi(T, g(T, X(T)))$. We assume a differentiable function Π_V exists such that:

$$\Pi_V(t, S(t)) = \mathbb{E}^{\mathbb{Q}}\left[\frac{V(T, S(T))}{M(T)}\ \middle|\ \mathcal{F}(t)\right] = \mathbb{E}^{\mathbb{Q}}\left[\frac{V(T, g(T, X(T)))}{M(T)}\ \middle|\ \mathcal{F}(t)\right] = \frac{V(t, g(t, X(t)))}{M(t)}.$$

We assume that the discounted option value is a martingale, this can be seen as a logical no arbitrage condition as pointed out earlier. We find for an infinitesimal change of Π_V:

$$\mathrm{d}\Pi_V = \mathrm{d}\frac{V}{M} = \frac{1}{M}\,\mathrm{d}V - \frac{V}{M^2}\,\mathrm{d}M = \frac{1}{M}\,\mathrm{d}V - r\frac{V}{M}\,\mathrm{d}t.$$

Applying Itô's Lemma to the option value yields:

$$\mathrm{d}V(t) = \left(\frac{\partial f}{\partial t} + \mu(X(t))\frac{\partial f}{\partial X} + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 f}{\partial X^2}\right)\,\mathrm{d}t + \sigma(X(t))\frac{\partial f}{\partial X}\,\mathrm{d}W^{\mathbb{Q}}(t).$$

So

$$\begin{aligned}
\mathrm{d}\Pi_V &= \frac{1}{M}\,\mathrm{d}V - r\frac{V}{M}\,\mathrm{d}t \\
&= \frac{1}{M}\left[\left(\frac{\partial f}{\partial t} + \mu(X(t))\frac{\partial f}{\partial X} + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 f}{\partial X^2}\right)\,\mathrm{d}t + \sigma(X(t))\frac{\partial f}{\partial X}\,\mathrm{d}W^{\mathbb{Q}}(t)\right] - r\frac{V}{M}\,\mathrm{d}t \\
&= \frac{1}{M}\left[\left(\frac{\partial f}{\partial t} + \mu(X(t))\frac{\partial f}{\partial X} + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 f}{\partial X^2} - rV\right)\,\mathrm{d}t + \sigma(X(t))\frac{\partial f}{\partial X}\,\mathrm{d}W^{\mathbb{Q}}(t)\right].
\end{aligned}$$

Since we assumed that the discounted option value is a martingale, the part before $\mathrm{d}t$ must be zero. Therefore we see that:

$$\frac{\partial f}{\partial t} + \mu(X(t))\frac{\partial f}{\partial X} + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 f}{\partial X^2} - rV = 0.$$

Substituting $V(t, S(t)) = f$ we now get the desired PDE:

$$\left\{\begin{array}{rl}
\frac{\partial V}{\partial t} + \mu(X(t))\frac{\partial V}{\partial X} + \frac{1}{2}\sigma^2(X(t))\frac{\partial^2 V}{\partial X^2} - rV &= 0 \\
V(T,S) &= \Phi(g(T,X(T)))
\end{array}\right. .$$

Which in turn becomes

$$\frac{\partial V}{\partial t} + \mu(X)\frac{\partial V}{\partial X} + \frac{1}{2}\sigma^2(X)\frac{\partial^2 V}{\partial X^2} - rV = 0,$$

with final condition

$$V(T,S) = \Phi(g(T,X(T))).$$

We will later use this PDE to price options using the FDM.

5 Numerical methods

5.1 Finite difference method

The idea behind the Finite Difference Method(FDM) is to approximate derivatives by using the Taylor expansion of the function around a certain point. Suppose we want to approximate $\frac{\partial f}{\partial x}$.

$$\begin{aligned}
f(x-h) =& f(x) - h\frac{\partial f}{\partial x}(x) + \frac{h^2}{2}\frac{\partial^2 f}{\partial x^2}(x) + -\frac{h^3}{6}\frac{\partial^3 f}{\partial x^3}(x) + \mathcal{O}(h^4) \\
f(x) =& f(x) \\
f(x+h) =& f(x) + h\frac{\partial f}{\partial x}(x) + \frac{h^2}{2}\frac{\partial^2 f}{\partial x^2}(x) + \frac{h^3}{6}\frac{\partial^3 f}{\partial x^3}(x) + \mathcal{O}(h^4) \\
&.
\end{aligned}$$

So we see that

$$\frac{f(x+h)-f(x-h)}{2h} = \frac{2h\frac{\partial f}{\partial x}(x) + \frac{h^3}{3}\frac{\partial^3 f}{\partial x^3}(x) + \mathcal{O}(h^4)}{2h} = \frac{\partial f}{\partial x}(x) + \mathcal{O}(h^2)$$

If we let $h \to 0$ we see that this converges quadratically to the desired derivative.
Since we have a PDE with a time component we need to do some form of time integration when using the FDM. The Euler Backwards time-integration is used here, because it is unconditionally stable. There are many textbook examples of how this method works, for example in [9]

5.2 Finite difference method for Black-Scholes model

To use the FDM on the Black-Scholes model we need to discretise the partial differential equation (PDE):

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0.$$

Since we have a final condition at $t = T$, we transform this by using $\tau = T - t$ to get an initial condition and find the PDE:

$$\frac{\partial V}{\partial \tau} - \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} - rS\frac{\partial V}{\partial S} + rV = 0$$

We make a grid with points at $t = 0$, $t = T$ and $N_t + 1$ points thus with mesh width $k = \frac{T}{N_t}$. For the stock dimension we make a grid with points at $S = 0$, upper limit $S = L$ and $N_x + 1$ points thus with mesh width $h = \frac{L}{N_x}$.
We use a Taylor expansion:

$$\begin{aligned}
V_{j+1}^{i+1} =& \quad V_j^{i+1} + h\frac{\partial V_j^{i+1}}{\partial S} + \frac{h^2}{2}\frac{\partial^2 V_j^{i+1}}{\partial S^2} + \frac{h^3}{6}\frac{\partial^3 V_j^{i+1}}{\partial S^3} + \mathcal{O}(h^4) \\
V_{j+1}^{i+1} =& \qquad\qquad V_{j+1}^{i+1} \\
V_{j-1}^{i+1} =& \quad V_j^{i+1} - h\frac{\partial V_j^{i+1}}{\partial S} + \frac{h^2}{2}\frac{\partial^2 V_j^{i+1}}{\partial S^2} - \frac{h^3}{6}\frac{\partial^3 V_j^{i+1}}{\partial S^3} + \mathcal{O}(h^4).
\end{aligned}$$

If we now look at

$$\frac{\partial^2 V_{j+1}^{i+1}}{\partial S^2} = \frac{V_{j-1}^{i+1} - 2V_j^{i+1} + V_{j+1}^{i+1}}{h^2} + \mathcal{O}(h^2).$$

Furthermore we see that

$$\frac{V_j^{i+1} - V_j^i}{k} = \frac{\partial V_j^{i+1}}{\partial \tau} + \mathcal{O}(k)$$

$$\frac{V_{j+1}^{i+1} - V_{j-1}^{i+1}}{2h} = \frac{\partial V_j^{i+1}}{\partial S} + \mathcal{O}(h^2).$$

We approximate V at (jh, ik) by solving the difference equation:

$$\frac{V_j^{i+1} - V_j^i}{k} - \frac{\sigma^2}{2}(jh)^2 \frac{V_{j+1}^{i+1} - 2V_j^{i+1} + V_{j-1}^{i+1}}{h^2} - r(jh)\frac{V_{j+1}^{i+1} - V_{j-1}^{i+1}}{2h} + rV_{j+1}^{i+1} = 0.$$

So, for an internal point we have

$$\left(1 + rk + k\frac{\sigma^2}{h^2}(jh)^2\right) V_j^{i+1} - k\left(\frac{\sigma^2(jh)^2}{2h^2} + \frac{r(jh)}{2h}\right) V_{j+1}^{i+1} - k\left(\frac{\sigma^2(jh)^2}{2h^2} - \frac{r(jh)}{2h}\right) V_{j-1}^{i+1} = V_j^i$$

$$\left(1 + rk + k\sigma^2 \cdot j^2\right) V_j^{i+1} - k\left(\frac{\sigma^2 j^2}{2} + \frac{rj}{2}\right) V_{j+1}^{i+1} - k\left(\frac{\sigma^2 j^2}{2} - \frac{rj}{2}\right) V_{j-1}^{i+1} = V_j^i.$$

For a call option we have boundary conditions:

$$V_C(0, \tau) = 0 \quad \text{and} \quad V_C(L, \tau) = L - K \cdot e^{-r\tau}.$$

where $L \gg K$ and initial condition

$$V_C(S, 0) = \max(S(0) - K, 0).$$

We can write this in matrix-vector form with

$$\mathbf{V}^i = \begin{bmatrix} V_1^i \\ V_2^i \\ \vdots \\ \vdots \\ V_{N_x-1}^i \end{bmatrix}.$$

We then get

$$B\mathbf{V}^{i+1} = \mathbf{V}^i + \mathbf{q}^i.$$

We first write:

$$D = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 2 & 0 & \ddots & \vdots \\ \vdots & 0 & 3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & N_x - 1 \end{bmatrix}, \; E = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 4 & 0 & \ddots & \vdots \\ \vdots & 0 & 9 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & (N_x - 1)^2 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ -1 & 0 & 1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & -1 & 0 \end{bmatrix}, \; H = \begin{bmatrix} -2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix},$$

to write

$$B = (1 + rk)I - \frac{1}{2}k\left(rDG + \sigma^2 EH\right).$$

The boundary conditions become:

$$V_{N_x}^i = \left(1 + rk + k\sigma^2 \cdot (N_x)^2\right) V_{N_x}^{i+1}$$
$$- k\left(\frac{\sigma^2(N_x)^2}{2} + \frac{r(N_x)}{2}\right) V_{N_x+1}^{i+1} - k\left(\frac{\sigma^2(N_x)^2}{2} - \frac{r(N_x)}{2}\right) V_{N_x-1}^{i+1}.$$

Because we know $V_{N_x+1}^{i+1}$ we can shift that part such that

$$V_{N_x}^i + \frac{kN_x}{2}\left(\sigma^2 N_x + r\right) V_{N_x+1}^{i+1} = \left(1 + rk + k\sigma^2 N_x^2\right) V_{N_x}^{i+1}$$
$$- k\left(\frac{\sigma^2 N_x^2}{2} - \frac{rN_x}{2}\right) V_{N_x-1}^{i+1}.$$

This gives us

$$\mathbf{q}^i = \begin{bmatrix} 0 \\ \vdots \\ \vdots \\ 0 \\ \frac{1}{2}kN_x(\sigma^2 N_x + r)V_{N_x+1}^{i+1} \end{bmatrix}.$$

5.3 FDM for pricing a barrier option

We now consider a down-and-out call option $V_{C,B}(S,t)$. The Black-Scholes PDE is relevant as long as the barrier is not corssed, therefore the barrier option must satisfy the Black-Scholes PDE on the grid $[0,T],[B,S]$. So when the barrier B is crossed, the option becomes worthless, giving us the boundary condition

$$V_{C,B}(B,t) = 0, \quad \forall t \in [0,T].$$

The upper boundary condition used in the previous section still holds, therefore we now use the same FDM with a different boundary condition.

5.4 FDM for pricing under the local-volatility model

We have a volatility parameter that depends on the asset price, thus $\sigma(S,\tau)$. The local volatility now depends on this.
We use the Hagan formula from [5] for the market-implied volatility to get results that are easily reproducable. We take parameters that are supposed to be moderate and do not allow arbitrage(see [1]). Those parameters are: $\alpha = 0.2,\ \beta = 0.5,\ \gamma = 0.2,\ \rho = -0.9,\ f(t_0,t) = S_0 e^{rt}$ with $S_0 = 1$ and $r = 0.03$. This formula is given below and looks long and complicated, but it is explicit, which makes it easy and fast to calculate using a computer. It comes from the SABR model and allows for control of the volatility smile by altering the parameters α, β, γ and ρ.

$$\sigma_{\text{imp}} = \frac{\alpha}{(fK)^{(1-\beta)/2}\left\{1 + \frac{(1-\beta)^2}{24}\log^2(\frac{f}{K}) + \frac{(1-\beta)^4}{1920}\log^4(\frac{f}{K})\right\}} \cdot \left(\frac{z}{\chi(z)}\right) \cdot$$
$$\left\{1 + \left[\frac{(1-\beta)^2}{24}\frac{\alpha^2}{(fK)^{(1-\beta)}} + \frac{\rho\beta\gamma\alpha}{4(fK)^{(1-\beta)/2}} + \frac{(2-3\rho^2)\gamma^2}{24}\right]T\right\}$$

where

$$z = \frac{\gamma}{\alpha}(fK)^{(1-\beta)/2}\log\left(\frac{f}{K}\right)$$

and

$$\chi(z) = \log\left\{\frac{\sqrt{1-2\rho z+z^2}+z-\rho}{1-\rho}\right\}.$$

We take a grid of K and τ and determine the implied volatility at these points, to get a so-called volatility surface.
With this implied volatility surface, we can determine the local volatility. We approximate

$$\bar{\sigma}^2(\tau,K) = \frac{\bar{\sigma}_{\text{imp}}^2 + 2\tau\bar{\sigma}_{\text{imp}}\left(-\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial\tau} + r\cdot K\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)}{\left(1-\frac{K\cdot\log\left(\frac{K}{S_0e^{r\tau}}\right)}{\sigma_{\text{imp}}}\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K} - \frac{1}{4}K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\frac{\partial^2\bar{\sigma}_{\text{imp}}}{\partial K^2}\right)}$$

by approximating the derivatives of the implied volatility with respect to τ and K using the FDM. This gives us

$$\bar{\sigma}^2(\tau,K) = \frac{\bar{\sigma}_{\text{imp}}^2 + 2\tau\bar{\sigma}_{\text{imp}}\left(-\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial\tau} + r\cdot K\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)}{\left(1-\frac{K\cdot\log\left(\frac{K}{S_0e^{r\tau}}\right)}{\sigma_{\text{imp}}}\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K} - \frac{1}{4}K\tau\bar{\sigma}_{\text{imp}}\left(\frac{\partial\bar{\sigma}_{\text{imp}}}{\partial K}\right)^2 + K\frac{\partial^2\bar{\sigma}_{\text{imp}}}{\partial K^2}\right)}.$$

With the local volatility we can now approximate the option prices with the FDM with help of the PDE. These should match the option prices found by putting the implied volatility into the Black-Scholes model. We get the discretisation:

$$\frac{V_j^{i+1}-V_j^i}{k} - \frac{\bar{\sigma}^2(jh,k(i+1))}{2}(jh)^2\frac{V_{j+1}^{i+1}-2V_j^{i+1}+V_{j-1}^{i+1}}{h^2} - r(jh)\frac{V_{j+1}^{i+1}-V_{j-1}^{i+1}}{2h} + rV_j^{i+1} = 0.$$

This means that:

$$(1+rk)V_j^{i+1} - \frac{\bar{\sigma}^2(jh,k(i+1))}{2}kj^2\left(V_{j+1}^{i+1}-2V_j^{i+1}+V_{j-1}^{i+1}\right) - \frac{rkj}{2}\left(V_{j+1}^{i+1}-V_{j-1}^{i+1}\right) = V_j^i.$$

We get a similar matrix representation as for the standard European call option, but with another matrix for the volatility:

$$D = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 2 & 0 & \ddots & \vdots \\ \vdots & 0 & 3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & N_x-1 \end{bmatrix}, \quad E = \begin{bmatrix} 1 & 0 & \cdots & \cdots & 0 \\ 0 & 4 & 0 & \ddots & \vdots \\ \vdots & 0 & 9 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & (N_x-1)^2 \end{bmatrix}$$

and

$$G = \begin{bmatrix} 0 & 1 & 0 & \cdots & \cdots & 0 \\ -1 & 0 & 1 & \ddots & \ddots & \vdots \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & -1 & 0 & 1 \\ 0 & \cdots & \cdots & 0 & -1 & 0 \end{bmatrix}, \quad H = \begin{bmatrix} -2 & 1 & 0 & \cdots & \cdots & 0 \\ 1 & -2 & 1 & \ddots & \ddots & \vdots \\ 0 & 1 & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & 1 & -2 & 1 \\ 0 & \cdots & \cdots & 0 & 1 & -2 \end{bmatrix},$$

and volatility matrix:

$$R^{i+1} = \begin{bmatrix} \bar{\sigma}^2(h, k(i+1)) & 0 & \cdots & \cdots & 0 \\ 0 & \bar{\sigma}^2(2h, k(i+1)) & 0 & \ddots & \vdots \\ \vdots & 0 & \bar{\sigma}^2(3h, k(i+1)) & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \bar{\sigma}^2((N_x-1)h, k(i+1)) \end{bmatrix}$$

Giving us

$$B^{i+1} = (1 + rk)I - \frac{1}{2}k\left(rDG + EHR^{i+1}\right)$$

such that

$$B^{i+1}\mathbf{V}^{i+1} = \mathbf{V}^i + \mathbf{q}^i.$$

5.5 FDM for pricing under the CLV framework

A process must be chosen for X. In this example the Ornstein-Uhlenbeck process is chosen with the parameters λ, θ and η to be specified freely. Remember the OU process is determined as:

$$\mathrm{d}X(t) = \lambda(\theta - X(t))\ \mathrm{d}t + \eta\ \mathrm{d}W^{\mathbb{Q}}(t)$$

This leads to the PDE (backwards in time):

$$\begin{cases} \frac{\partial V}{\partial \tau} - \lambda(\theta - X(t))\frac{\partial V}{\partial X} - \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial X^2} + rV &= 0 \\ V(T,S) &= \Phi(g(T, X(T))) \end{cases}$$

A grid for X and τ is constructed on which the finite difference method, with Euler backwards time integration, is used to determine the option value at starting time.
We need to find the inverse of the market-implied density:

$$F_{\hat{S}(T_i)}(x) = 1 + e^{rT}\ \frac{\partial V(T_i, K)}{\partial K}\bigg|_{x=K}$$

Since the CDF is monotonic, and a bijective function, so is its inverse. Therefore we will use interpolation on the inverse to determine

$$s_{i,j} = F^{-1}_{\hat{S}(T_i)}\left(F_{X(T_i)}(x_{i,j})\right)$$

With the collocation points $x_{i,j}$ and the collocation values $s_{i,j}$ the asset price can be approximated at any time in $[0, T]$. Because the grid for X is relatively large, some values of X are smaller and greater than the collocation points $x_{i,j}$. For a normal process, like the Ornstein-Uhlenbeck process, the collocation points can be determined analytically. The collocation values in between $[T_i, T_{i+1}]$ are determined by linear interpolation with time. So for $t \in [T_i, T_{i+1}]$:

$$s_j(t) = s_{i,j} + (s_{i+1,j} - s_{i,j})\frac{t - T_i}{T_{i+1} - T_i}.$$

After this linear extrapolation of the collocation values is used to determine the asset prices for low and high values of X at each timestep. With these values the boundary conditions can be determined using the knowledge that the discounted option price is a martingale. With the collocation values and the collocation points a monotone interpolation is performed to determine the boundary condition at expiry. In this case piecewise cubic Hermite interpolation is used to ensure montonicity. For more details on this see [9].

We now write the PDE

$$\begin{cases} \frac{\partial V}{\partial \tau} - \lambda(\theta - X(t))\frac{\partial V}{\partial X} - \frac{1}{2}\eta^2\frac{\partial^2 V}{\partial X^2} + rV &= 0 \\ V(T,S) &= \Phi(g(T, X(T))) \end{cases}$$

in terms of finite difference approximations, where we take $X \in [BB, L]$, where BB is the lower bound and L the upper bound. From a heuristical approach we take $BB = \mathbb{E}[X] - 1$ and $L = \mathbb{E}[X] + 1$ for the expiry of one year. The variance of the process X this is smaller than one and the mapping function g uses the quantiles of the distribution X and the interval $[\mathbb{E}[X] - 1, \mathbb{E}[X] + 1]$ captures the majority of this distribution. Thus

$$\frac{V_j^{i+1} - V_j^i}{k} - \lambda(\theta - jh)\left(\frac{V_{j+1}^{i+1} - V_{j-1}^{i+1}}{2h}\right) - \frac{1}{2}\eta^2\left(\frac{V_{j+1}^{i+1} - 2V_j^{i+1} + V_{j-1}^{i+1}}{h^2}\right) + rV_j^{i+1} = 0.$$

This can be rewritten as:

$$(1 + rk)V_j^{i+1} - k\lambda(\theta - jh)\left(\frac{V_{j+1}^{i+1} - V_{j-1}^{i+1}}{2h}\right) - \frac{k}{2}\eta^2\left(\frac{V_{j+1}^{i+1} - 2V_j^{i+1} + V_{j-1}^{i+1}}{h^2}\right) = V_j^i.$$

This is put into matrix-vector form to speed up the computation in matlab.
For a barrier option the FDM for the CLV model works similarly, but the value $X = g^{-1}(t, S(t))$ has to be determined for $S = B$ at each timestep. The option value at time t is set to zero if the value of X is smaller than $g^-1(t, B)$. This inverse is determined by using linear inverse interpolation.

5.6 Monte Carlo pricing under the CLV framework

The Monte Carlo(MC) approach for option pricing is based on the discounted martingale property. We want to evaluate the discounted payoff. By taking very many samples of the process S we can approximate the expected value of the payoff and discount this to receive the option price. So we generate many paths for, use the payoff function and determine the mean of these values. In this case we must sample from the "cheap" stochastic process X and map this onto S. For European call and put options this requires only one step to simulate X. The construction of the mapping function g goes analogous as in the previous section. The paths of X are mapped and we compute the payoff for each mapped path. Taking the mean of the discounted value of these payoffs results in an estimate for the option value. In the implementation antithetic variables are used to ensure a smaller variance in our estimate, since we can construct a stochastic process with the same distribution but negative correlation. It is known that for a normally distributed variable $X = U \sim \mathcal{N}(0, 1)$ that also $Y = -U \sim \mathcal{N}(0, 1)$. Then

$$\mathbb{V}\text{ar}\left(\frac{X+Y}{2}\right) = \frac{\mathbb{V}\text{ar}(X) + \mathbb{V}\text{ar}(Y) + 2\text{Cov}(X, Y)}{4} < \frac{\mathbb{V}\text{ar}(X) + \mathbb{V}\text{ar}(Y)}{4}.$$

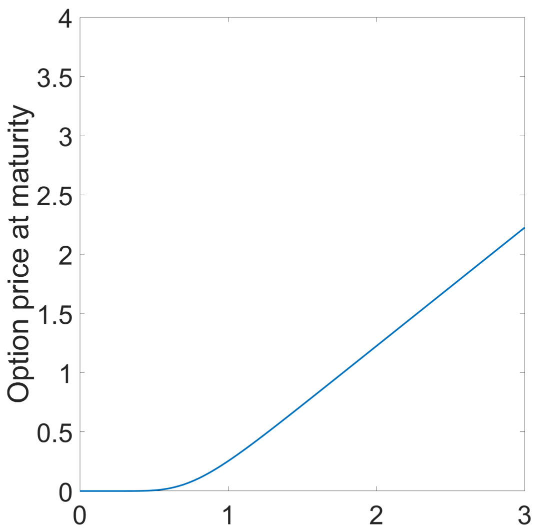
This will thus result in a smaller variance, and thus improve our estimate.
When dealing with a barrier option, it is necessary to check if the asset price hits the barrier between the expiry and starting time, which requires us to take small timesteps and make the simulation more computationally expensive. When considering a barrier option under the CLV model at every timestep determine $g^{-1}(B) = X_B$ must be determined, similarly as in the FDM approach. At every timestep it is checked if the barrier in the X domain is crossed. This is used to set the option value to zero if the boundary is crossed at any time between $[0, T]$.

6 Numerical results

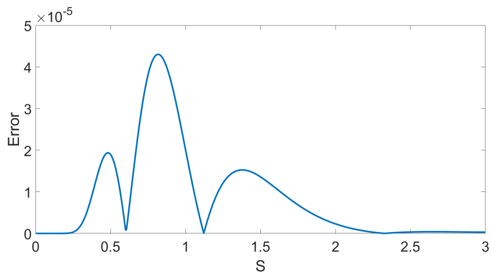
In this section we will use the techniques described in the previous chapter to illustrate the results of the different models and highlight the differences between them. For all examples in this section we will first consider a European call option with strike price $K = 0.8$, risk-free interest rate $r = 0.03$ and expiry of one year. When we consider a barrier option, we will use a down-and-out call option with a barrier at $B = 0.6$. All programs were created with matlab and the code used to obtain the results can be found in the appendix.

6.1 Results for pricing under the Black-Scholes model

For a given volatility and stock price we can now determine the value analytically using the Black-Scholes model. We can also approximate this by using the FDM. We take the volatility to be $\sigma = 0.3$. As you can see in the picture below, these are almost exactly the same. At expiry there is a larger error around the strike price, this is only natural since there is more curvature there, which the FDM does not fully capture. When we are further from expiry this error becomes smaller.



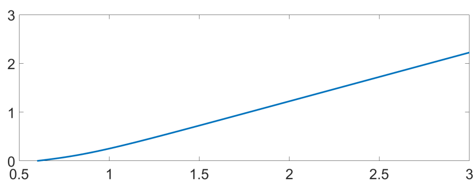
(a) Value of a call option expiring in one year as a function of asset price $S(T)$.



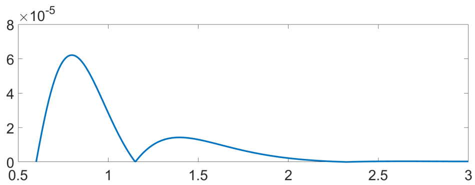
(b) Absolute difference between the call option values found with the FDM and the one found by using the Black-Scholes analytic solution.

6.2 Results for pricing a barrier option under the Black-Scholes model

Just like for the regular call option, we compare the prices found by the Black-Scholes analytical solution with the price found with the FDM.



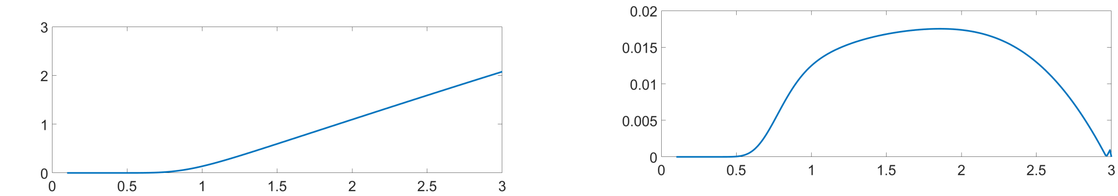
(a) Value of a down-and-out call option expiring in one year.



(b) Absolute difference between the down-and-out call option values found with the FDM and the one found by using the Black-Scholes analytic solution.

6.3 Results for pricing under the local volatility model

The formula by Hagan is used, as described in section 3.4. This gives us an implied volatility surface is used to compute the local volatility which is then used in the FDM. The parameters used are: $\alpha = 0.2,\ \beta = 0.5,\ \gamma = 0.2,\ \rho = -0.9,\ f(t_0, t) = S_0 e^{rt}$ with $S_0 = 1$ and $r = 0.03$. The option price found is compared with the option price found by putting the implied volatility into the Black-Scholes model. The local volatility method produces an even smaller error when used to price a European call option.

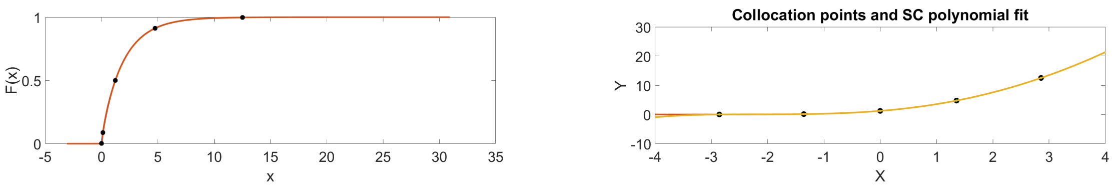


(a) Value of a down-and-out call option expiring in one year found using the local volatility model.

(b) Absolute difference between the down-and-out call option values found with the FDM and the one found by using the Black-Scholes analytic solution.

6.4 Results for using stochastic collocation

Before stochastic collocation is applied to the CLV framework an example of stochastic collocation is shown. Consider the non-central χ^2 distribution. Using matlab a χ^2 distribution is taken with shape parameter $d = 1.732$ and $\lambda = 0.15$ degrees of freedom and $N = 5$ collocation points. This is then interpolated with the Lagrange polynomial and compared with the actual distribution, as can be seen in the figure below.



(a) Cumulative distribution function of our approximation

(b) Values of the mapping function $Y = g(X)$ and approximation of g using Lagrange interpolation.

6.5 Results for pricing under the CLV model with the FDM

The Ornstein-Uhlenbeck process is chosen for X with parameters $\theta = 0.1$, $\lambda = 1.3$ and $\eta = 0.25$. Note that the choice of these parameters should not matter for how well the mapping function g behaves. Once again the formula by Hagan is used to give an implied volatility surface, which is used to calibrate the CLV model. Consider a set of expiries $\{0.05, 0.25, 0.5, 1, 2, 3, 4\}$. $N = 4$ collocation points are chosen, so 28 collocation points and values are determined. The collocation points are found below. These collocation points yield the following collocation values:

A grid is take from $[\mathbb{E}[X(T)] - 1, \mathbb{E}[X(T)] + 1] = [-0.6547, 1.3453]$. This results in an option price of 0.1594 while the Black-Scholes solution yields a price of 0.1587, thus there is an error of approximately 0.5%.

6.6 Results for pricing under the CLV model with MC simulation

The same set-up as for the FDM is used to price with MC simulation. $M = 10^6$ samples are generated. For the maturity time of one years the CLV method finds the price 0.1589. The price

T	$x_{i,1}$	$x_{i,2}$	$x_{i,3}$	$x_{i,4}$
0.05	0.8170	0.9032	0.9835	1.0697
0.25	0.5001	0.6707	0.8298	1.0005
0.5	0.2611	0.4717	0.6680	0.8785
1	-0.0030	0.2346	0.4560	0.6935
2	-0.1941	0.0521	0.2816	0.5278
3	-0.2436	0.0032	0.2332	0.4801
4	-0.2570	-0.0101	0.2200	0.4669

T	$s_{i,1}$	$s_{i,2}$	$s_{i,3}$	$s_{i,4}$
0.05	0.8952	0.9684	1.0350	1.1047
0.25	0.7638	0.9340	1.0828	1.2341
0.5	0.6668	0.9109	1.1224	1.3289
1	0.5296	0.8835	1.1842	1.4641
2	0.3384	0.8534	1.2832	1.6574
3	0.1980	0.8370	1.3695	1.8093
4	0.0863	0.8279	1.4501	1.9410

Table 1: Collocation points and values for the CLV model.

given by the Black-Scholes analytical solution is 0.1589 so the error is approximately 0.1% off the actual price.
A series of tests were done to compare option prices for different levels of the strike price, barrier level and expiry time using the different methods:

T	K	B	Black-Scholes	LV	CLV FDM	CLV MC
1	0.9	0	0.1587	0.1598	0.1594	0.1589
1	0.9	0.7	0.1586	0.1728	0.1301	0.1286
1	0.8	0.75	0.2322	0.2433	0.1792	0.173
1	0.85	0.7	0.1963	0.2112	0.1638	0.1628
2	0.8	0.75	0.2559	0.2715	0.1437	0.1432
2	0.85	0.7	0.2344	0.2579	0.1403	0.1377

Table 2: Option prices for different parameters and different models.

In the table above it can be seen that the Black-Scholes model and Local Volatility model give higher option prices than the CLV model. The prices of the CLV model using the FDM and MC approach are quite similar. There are differences in between them but these are relatively small.

7 Conclusion

Even though the Black-Scholes model is often used and relatively straightforward with an analytical solution, the model does not reflect the reality of the stock markets. Especially the so-called volatility skew or volatility smile is not incorporated into the model. This means that for options on the same stock with the same time to expiry, the implied volatility is different for different strike prices. This is not possible under the Black-Scholes model but can be seen in the market.

The local volatility model overcomes this weakness by allowing a volatility skew. The local volatility model perfectly fits the market data and differs from the Black-Scholes model by assuming that the volatility is also dependant on the asset price. However when using the local volatility model it is not always possible to compute prices for exotic options, because there is no control over the forward-start volatility. The CLV model overcomes this weakness. It takes a simple to compute process X which is then mapped onto the asset price S. We can choose the process X freely which gives us control over the forward-start volatility while the model ensures fit to market data with a mapping function g. This enables the model to compute the correct option prices and implied volatilities while enabling one to choose the shape of the forward-start implied volatility.

The CLV model can be used by both Monte Carlo simulation and the FDM. Both have their benefits, the MC simulation is more intuitive and easy to implement, while retaining accuracy. When using the Ornstein-Uhlenbeck process the MC simulation requires only one time step for each simulation, which is very efficient. It can also be used in higher dimensions very well, while the efficiency of the FDM suffers greatly if the number of dimensions is increased. Furthermore, the FDM allows for easy computation of the derivatives, which are required for hedging, a common market practice. However when considering a barrier option the FDM only needs to change the boundary conditions, while the MC simulation needs to take many small timesteps, which decrease the computational efficiency. The results in the previous section show that for pricing the results do not differ much.

In short, the CLV method looks very promising because it allows for great control over the forwards-start volatility while retaining very good perfect market fit. This ensures that the CLV model is capable of accurately pricing more exotic options than the other models discussed. When ipmlementing the CLV model the FDM could be used for hedging strategies while for more accurate pricing or higher dimension MC simulation is preferred.

8 Discussion

This thesis does not use actual market data, so it should be validated with actual market data to see if the CLV model behaves well. While the MC approach and the FDM approach to the CLV model generate different prices, as they both have small errors, the prices generated should be similar. When considering a more complex exotic option it might be worthwhile to use both methods and compare the results to get a better understanding of the option price. A possible extension of this research is to use the CLV model for higher dimensions, such as different stocks that are correlated and compare the results. It might also be interesting to see how the CLV model and the different implementations behave when pricing an option which allows early excercise.

References

[1] L. A. Grzelak. The CLV Framework: A fresh look at efficient pricing with Smile. *Forthcoming, Available at SSRN: http://ssrn.com/abstract=2747541,* 2016.

[2] M. Baxter and A. Rennie Financial Calculus: An Introduction to Derivative Pricing. *Cambridge University Press,* 1996. ISBN: 978-0521552899

[3] F. Black and M. Scholes. The Pricing of Options and Corporate Liabilities. *Journal of Political Economy,* 1973.

[4] D. J. Higham. An introduction to financial option valuation. *Cambridge University Press,* 2009. ISBN: 978-0-521-54757-4

[5] P. S. Hagan, D. Kumar, A. S. Leśniewski and D. E. Woodward. Managing smile risk. *Wilmott Magazine*, page 84-108, 2002.

[6] L. A. Grzelak and C. W. Oosterlee. From arbitrage to arbitrage-free volatilities. *Journal of Computational Finance, Forthcoming, Available at SSRN: http://ssrn.com/abstract=2529684,* 2016.

[7] L. A. Grzelak, J. A. S. Witteveen, M. Suarez and C. W. Oosterlee. The stochastic collocation Monte Carlo sampler: Highly efficient sampling from "expensive" distributions. *Submitted, Available at SSRN: http://ssrn.com/abstract=2529691,* 2014.

[8] B. Dupire. Pricing with a smile. *Risk,* 7:18-20, 1994.

[9] C. Vuik, F. J. Vermolen, M. B. van Gijzen and M. J. Vuik. Numerical Methods for Ordinary Differential Equations. *Delft Academic Press/VSSD*, 2015.

A Appendix

A.1 Matlab code for results

A.1.1 Black_Scholes_call.m

```
%By Jan Tijink 4203240
%Euler Backwards time integration for call option
%using Finite central differences for approximations of derivatives
clf
%parameters
E = 0.9; sigma = 0.3; r = 0.03; T = 1;
L = 3; Nx = 500; Nt = 300; k = T/Nt; h = L/Nx;
%
T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1)  + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [h:h:L-h]/h;
D1 = diag(mvec);
D2 = diag(mvec.^2);

B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*sigma.^2*D2*T2 - 0.5*k*r*D1*T1;
U = zeros(Nx-1,Nt+1);
%boundary conditions and initial condition
U(:,1) = max([h:h:L-h]'-E,0);
bca = zeros(1,Nt+1);
bcb = L-E*exp(-r*[0:k:T]);
U = [bca;U;bcb];
q=zeros(Nx-1,1);
%Euler backward time integration
for i = 1:Nt
    q(Nx-1,1)= k*(0.5*sigma^2*((Nx-1)*h)^2/(h^2) + 0.5*r*((Nx-1)*h)/h)*U(Nx+1,i+1);
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);
end
%computing the exact value
U_exact=zeros(Nx+1,Nt+1);
for j=1:Nt+1
    for i=1:Nx+1
        [c,cdelta,p,pdelta]=ch08((i-1)*h,E,r,sigma,(j-1)*k);
        U_exact(i,j)=c;
    end
end
```

A.1.2 Black_Scholes_DownOutCall.m

```
%By Jan Tijink 4203240
%Euler Backwards time integration for down-and-out call
%using Finite central differences for approximations of derivatives
clf
%parameters
E = 0.9; sigma = 0.3; r = 0.03; T = 1; Barrier=0.6;
L = 3; Nx = 500; Nt = 300; k = T/Nt; h = (L-Barrier)/Nx;

T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1)  + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [Barrier+h:h:L-h]/h;
D1 = diag(mvec);
D2 = diag(mvec.^2);
B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*sigma^2*D2*T2 - 0.5*k*r*D1*T1;
U = zeros(Nx-1,Nt+1);
%boundary conditions and initial condition
U(:,1) = max([Barrier+h:h:L-h]'-E,0);
bca = zeros(1,Nt+1);
bcb = L-E*exp(-r*[0:k:T]);
U = [bca;U;bcb];
q=zeros(Nx-1,1);
%Euler backward time integration
for i = 1:Nt
    q(Nx-1,1)= k*(0.5*sigma^2*(Barrier+(Nx-1)*h)^2/(h^2) + 0.5*r*(Barrier+(Nx-1)*h)/h)*U(Nx+1,i+1);
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);
end
%Black-Scholes solution
Cbarrier_exact=zeros(Nx+1,Nt+1);
for j=1:Nt+1
    for i=1:Nx+1
        [c,cdelta,p,pdelta]=BlackScholes(Barrier+(i-1)*h,E,r,sigma,(j-1)*k);
        [c2,cdelta2,p2,pdelta2]=BlackScholes(Barrier^2/(Barrier+(i-1)*h),E,r,sigma,(j-1)*k);
        Cbarrier_exact(i,j)=c-((Barrier+(i-1)*h)/Barrier)^(1-2*r/(sigma^2))*c2;
    end
end
```

A.1.3 LocalVolatility.m

```
%By Jan Tijink 4203240
%Euler Backwards time integration for a European call option using Hagan
%formula for volatility
%using Finite central differences for approximations of derivatives
clf
%parameters Hagan formula (SABR)
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
%other parameters
T=1;S0=1; r=0.03; E = 0.9; L = 3; Nx = 300; Nt = 300; k = T/Nt; h = L/Nx; S_0=1;
%matrices for Euler Backwards
T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
```

```
T2 = -2*eye(Nx-1,Nx-1) + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [h:h:L-h]/h;
D1 = diag(mvec);D2 = diag(mvec.^2);
U = zeros(Nx-1,Nt+1);
%boundary conditions and initial condition
U(:,1) = max([h:h:L-h]'-E,0);
bca = zeros(1,Nt+1);
bcb = L-E*exp(-r*[0:k:T]);
U = [bca;U;bcb];
q=zeros(Nx-1,1);

sigma=zeros(Nx+1,Nt+1,3);
%implementing the Hagan formula for the implied volatility surface
for i=1:Nt
    for j=2:Nx+1
        for b=1:3
        sigma(j,i,b)=Haganformula(alpha,beta,gamma,rho,E+(b-2)*h,S0*exp(r*(T-(i-1)*k)),T);
        end
    end
end
%computing the Local volatility
Localvol2=zeros(Nx+1,Nt+1);
for i=2:Nt
    for j=2:Nx
        tau=(i-1)*k;
        SS=E;
        dsdt=(sigma(j,i+1,2)-sigma(j,i-1,2))/(2*k);
        dsdk=(sigma(j,i,3)-sigma(j,i,1))/(2*h);
        dsdk2=(sigma(j,i,3)-2*sigma(j,i,2)+sigma(j,i,1))/(h^2);
        Localvol2(j,i)=(sigma(j,i,2)^2+2*sigma(j,i,2)*tau*(-dsdt+r*SS*dsdk))/((1-SS*log(SS/(exp(r*tau)))/sigma(j,i,2)*dsdk)^2+SS*tau*sigma(j,i,2)*(dsdk-0.25*SS*tau*sigma(j,i,2)*(dsdk)^2+SS*dsdk2));
    end
end
%numerical integration
for i = 1:Nt
    B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*D2*T2*diag(Localvol2(2:Nx,i+1)) - 0.5*k*r*D1*T1;
    q(Nx-1,1)= k*(0.5*(Localvol2(Nx,i))*((Nx-1)*h)^2/(h^2) + 0.5*r*((Nx-1)*h)/h)*U(Nx+1,i+1);
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);
end
%computing the exact value
C_exact=zeros(Nx+1,Nt+1);
for i=2:Nt+1
    for j=1:Nx+1
        [c,cdelta,p,pdelta]=BlackScholes((j-1)*h,E,r,sigma(j,end,2),(i-1)*k);
        C_exact(j,i)=c;
    end
end
price=interp1([0:h:L],U(:,end),1)
```

A.1.4 HaganBarrierCallLocalVolatility.m

```
%By Jan Tijink 4203240
%Euler Backwards time integration for down and out call option
%using Finite central differences for approximations of derivatives
clf
%parameters Hagan formula (SABR)
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
%other parameters
T=1; r=0.03; E = 0.9; L = 3; Nx = 300; Nt = 300; k = T/Nt;Barrier=0.6; h = (L-Barrier)/Nx;

%matrices for Euler Backwards
T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1) + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [Barrier+h:h:L-h]/h;
D1 = diag(mvec);
D2 = diag(mvec.^2);

U = zeros(Nx-1,Nt+1);
%boundary conditions and initial condition
U(:,1) = max([Barrier+h:h:L-h]'-E,0);
bca = zeros(1,Nt+1);
bcb = L-E*exp(-r*[0:k:T]);
U = [bca;U;bcb];
q=zeros(Nx-1,1);

sigma=zeros(Nx+1,Nt+1,3);
%implementing the Hagan formula for the implied volatility surface by
%taking t and then reverting the order in time
for i=1:Nt+1
    for j=2:Nx+1
        for b=1:3
        sigma(j,i,b)=Haganformula(alpha,beta,gamma,rho,E+(b-2)*h,(Barrier+(j-1)*h)*exp(r*(T-(i-1)*k)),T);
        end
    end
end
%computing the local volatility
Localvol2=zeros(Nx+1,Nt+1);
for i=2:Nt
    for j=2:Nx
        tau=(i-1)*k;
        SS=E;
        dsdt=(sigma(j,i+1,2)-sigma(j,i-1,2))/(2*k);
        dsdk=(sigma(j,i,3)-sigma(j,i,1))/(2*h);
        dsdk2=(sigma(j,i,3)-2*sigma(j,i,2)+sigma(j,i,1))/(h^2);
```

```
            Localvol2(j,i)=(sigma(j,i,2)^2+2*sigma(j,i,2)*tau*(-dsdt+r*SS*dsdk))/((1-SS*log(SS/(exp(r*tau)))/sigma(j,i,2)*dsdk)^2+SS*tau*sigma(j,i,2)*(dsdk-0.25*SS*tau*sigma(j,i,2)*(dsdk)^2+SS*dsdk2));
        end
end

%numerical integration
for i = 1:Nt
    B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*D2*T2*diag(Localvol2(2:Nx,i+1)) - 0.5*k*r*D1*T1;
    q(Nx-1,1)= k*(0.5*(Localvol2(Nx,i))*(Barrier+(Nx-1)*h)^2/(h^2) + 0.5*r*(Barrier+(Nx-1)*h)/h)*U(Nx+1,i+1);
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);
end
%computing the exact value
C_exact=zeros(Nx+1,Nt+1);
C_exact(:,1)=max([Barrier:h:L]'-E,0);
for i=2:Nt+1
    for j=1:Nx+1
        [c,cdelta,p,pdelta]=BlackScholes(Barrier+(j-1)*h,E,r,sigma(j,end,2),(i-1)*k);
        C_exact(j,i)=c;
    end
end
```

A.1.5 CLV_FDM.m

```
%by Jan Tijink 4203240
N=4;
theta=0.1;
lambda=1.3;
S0=1;
X_0=S0;
eta=0.25;
T=[0.05,0.25,0.5,1,2,3,4];
xcoll=zeros(length(T),N);
x=[-2.3344;-0.7420;0.7420;2.3344]';
%getting the collocation values and points
F=zeros(length(T),N);
for i=1:7
    for j=1:4
        expectation=X_0*exp(-lambda*T(i))+theta*(1-exp(-lambda*T(i)));
        variance=eta^2/(2*lambda)*exp(-lambda*(2*T(i)))*(exp(2*lambda*T(i))-1);
        xcoll(i,j)=expectation+sqrt(variance)*x(j);
        F(i,j)=normcdf(xcoll(i,j),expectation,sqrt(variance));
    end
end
r=0.03; h = S0/2000;
E=[0.001:(10-0.03)/400:10]*S0;
sigma=zeros(length(T),length(E),3);
V=zeros(length(T),length(E),3);
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
for i=1:length(T)
    for j=1:length(E)
        for b=1:3
        sigma(i,j,b)=Haganformula(alpha,beta,gamma,rho,E(j)+(b-2)*h,S0*exp(r*T(i)),T(i));
        V(i,j,b)=BlackScholes(S0,E(j)+(b-2)*h,r,sigma(i,j,b),T(i));
        end
    end
end
dVdK=(V(:,:,3)-V(:,:,1))./(2*h);
SIJ=zeros(size(F));
FS=zeros(length(T),length(E));
FS2=zeros(length(T),length(E));
FSS=zeros(size(F));
extradvdk=zeros(1,N);
V2=size(FSS);
h=0.001;
%(inefficient) linear inverse interpolation to approximate distribution of S
for i=1:length(T)
    FS(i,:)=1+exp(r*T(i))*dVdK(i,:);
        lower=((1+exp(r*T(i))*dVdK(i,:))>F(1,1));
        upper=((1+exp(r*T(i))*dVdK(i,:))<F(end,end));
        teller1=0;
        teller2=0;
        for j=1:length(E)
            if lower(j)==1
                if teller1==0
                    teller1=j-1;
                end
            end
            if upper(j)==0
                if teller2==0
                    teller2=j;
                end
            end
        end
        dVdK2=dVdK(i,teller1:teller2);
        E2=E(teller1:teller2);
        FS2(i,1:length(dVdK2))=1+exp(r*T(i))*dVdK2;
        SIJ(i,:)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,:));
        for j=1:N
            sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
            sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
            FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
            while abs(FSS(i,j)-F(i,j))>10^(-4)
                E2=[E2,SIJ(i,j)];
```

```
                V2=BlackScholes(S0,SIJ(i,j)+h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i)),T(i));
                V3=BlackScholes(S0,SIJ(i,j)-h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i)),T(i));
                dVdK2=[dVdK2, (V2-V3)/(2*h)];
                SIJ(i,j)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,j));
                sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
                sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
                FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
            end
        end
end
%parameters and gridsize
expec=X_0*exp(-lambda*T(4))+theta*(1-exp(-lambda*T(4))); L=expec+2;
TT=T(4); K = 0.9; Nx = 300; Nt = 300; k = TT/Nt; BB=expec-2; h = abs(L-BB)/Nx;
%FDM implementation
q=zeros(Nx-1,1);
U = zeros(Nx+1,Nt+1);
tt=[k:k:TT];
U(:,1) = max(UBC0-K,0);
T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1)  + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [BB+h:h:L-h];
D1 = diag(mvec);
%setting the Boundary Conditions
UBC0=zeros(1,Nx+1);
BCS=interp1(xcoll(4,:),SIJ(4,:),[BB,L],'linear','extrap');
svals=[max(BCS(1),0),SIJ(4,:),BCS(2)];
xvals=[BB,xcoll(4,:),L];
UBC0=pchip([BB, xcoll(4,:),L],[BCS(1),SIJ(4,:),BCS(2)],[BB:h:L]);
collvalues=zeros(Nt+1,N);
BC=zeros(2,Nt);
for i=1:Nt
    xcols=S0*(exp(-lambda*tt(i))+theta*(1-exp(-lambda*tt(i)))) + sqrt(eta^2/(2*lambda)*exp(-lambda*(2*tt(i)))*(exp(2*lambda*tt(i))-1))*x;
    for j=1:N
        collvalues(i,j)=polyinterp(T,SIJ(:,j)',tt(i));
    end
    BC(:,i)=interp1(xcols,collvalues(i,:),[BB,L],'linear','extrap');
end
BC(2,:)=BC(2,:)-K*exp(-r*tt);
BC(1,:)=BC(1,:);%max(BC(2,:),0);
U(end,2:end) = BC(2,:);
U(1,2:end) = BC(1,:);
%numerical integration
for i = 1:Nt
    B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*T2*eta^2/(h^2) - 0.5*k*lambda*(eye(Nx-1,Nx-1)*theta-D1)*T1/h;
    q(Nx-1)=(k/2*eta^2/(h^2)+k*lambda/(2*h)*(theta-(BB+Nx*h)))*U(Nx+1,i+1);
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);
end
%computing the exact value
sigma=Haganformula(alpha,beta,gamma,rho,K,X_0*exp(r*T(4)),T(4));
[c,cdelta,p,pdelta]=BlackScholes(X_0,K,r,sigma,T(4));
Checkvalue=c
Optionvalue=interp1([BB:h:L],U(:,end),X_0)
```

A.1.6 CLV_FDM_Barrier.m

```
%by Jan Tijink
tic
N=4;
theta=0.1;
lambda=1.3;
X_0=1;
eta=0.25;
T=[0.05,0.25,0.5,1,2,3,4];
xcoll=zeros(length(T),N);
x=[-2.3344;-0.7420;0.7420;2.3344]';
Barrier=0.6;
%computing the collocation points
F=zeros(length(T),N);
for i=1:7
    for j=1:4
        expectation=X_0*exp(-lambda*T(i))+theta*(1-exp(-lambda*T(i)));
        variance=eta^2/(2*lambda)*exp(-lambda*(2*T(i)))*(exp(2*lambda*T(i))-1);
        xcoll(i,j)=expectation+sqrt(variance)*x(j);
        F(i,j)=normcdf(xcoll(i,j),expectation,sqrt(variance));
    end
end
S0=X_0; r=0.03; h = S0/2000; r=0.03;
E=[0.001:(10-0.03)/400:10]*S0;
sigma=zeros(length(T),length(E),3);
V=zeros(length(T),length(E),3);
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
for i=1:length(T)
    for j=1:length(E)
        for b=1:3
        sigma(i,j,b)=Haganformula(alpha,beta,gamma,rho,E(j)+(b-2)*h,S0*exp(r*T(i)),T(i));
        V(i,j,b)=BlackScholes(S0,E(j)+(b-2)*h,r,sigma(i,j,b),T(i));
        end
    end
end
dVdK=(V(:,:,3)-V(:,:,1))./(2*h);
SIJ=zeros(size(F));
FS=zeros(length(T),length(E));
```

```
FS2=zeros(length(T),length(E));
FSS=zeros(size(F));
extradvdk=zeros(1,N);
V2=size(FSS);
h=0.001;
%Determine collocation values
for i=1:length(T)
    FS(i,:)=1+exp(r*T(i))*dVdK(i,:);
    lower=((1+exp(r*T(i))*dVdK(i,:))>F(1,1));
    upper=((1+exp(r*T(i))*dVdK(i,:))<F(end,end));
    teller1=0;
    teller2=0;
    for j=1:length(E)
        if lower(j)==1
            if teller1==0
                teller1=j-1;
            end
        end
        if upper(j)==0
            if teller2==0
                teller2=j;
            end
        end
    end
    dVdK2=dVdK(i,teller1:teller2);
    E2=E(teller1:teller2);
    FS2(i,1:length(dVdK2))=1+exp(r*T(i))*dVdK2;
    SIJ(i,:)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,:));
    for j=1:N
        sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
        sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
        FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
        while abs(FSS(i,j)-F(i,j))>10^(-4)
            E2=[E2,SIJ(i,j)];
            V2=BlackScholes(S0,SIJ(i,j)+h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i)),T(i));
            V3=BlackScholes(S0,SIJ(i,j)-h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i)),T(i));
            dVdK2=[dVdK2, (V2-V3)/(2*h)];
            SIJ(i,j)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,j));
            sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
            sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
            FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
        end
    end
end

expec=X_0*exp(-lambda*T(end))+theta*(1-exp(-lambda*T(end)));
L=expec+2;
TT=max(T); r=0.03; K = 0.9; Nx = 300; Nt = 300; k = TT/Nt; BB=expec-2; h = abs(L-BB)/Nx;
%matrices for Euler Backwards
q=zeros(Nx-1,1);
%insert BC's
U = zeros(Nx+1,Nt+1);
%boundary conditions and initial condition
UBC0=zeros(1,Nx+1);
BCS=interp1(xcoll(end,:),SIJ(end,:),[BB,L],'linear','extrap');
svals=[max(BCS(1),0),SIJ(end,:),BCS(2)];
xvals=[BB,xcoll(end,:),L];
boundary=pchip(svals,xvals,Barrier);
UBC0=pchip(xvals,svals,[BB:h:L]);
UBC0([BB:h:L]<=boundary)=0;
tt=[k:k:TT];

U(:,1) = max(UBC0-K,0);
bca = zeros(1,Nt+1);
bcb = zeros(1,Nt+1);
T1 =  diag(ones(Nx-2,1),1) - diag(ones(Nx-2,1),-1);
T2 = -2*eye(Nx-1,Nx-1)  + diag(ones(Nx-2,1),1) + diag(ones(Nx-2,1),-1);
mvec = [BB+h:h:L-h];
D1 = diag(mvec);

collvalues=zeros(Nt+1,N);
BC=zeros(2,Nt);
barrierx=Barrier*ones(1,Nt+1);
for i=1:Nt
    xcols=S0*(exp(-lambda*tt(i))+theta*(1-exp(-lambda*tt(i)))) + sqrt(eta^2/(2*lambda)*exp(-lambda*(2*tt(i)))*(exp(2*lambda*tt(i))-1))*x;
    for j=1:N
        collvalues(i,j)=polyinterp(T,SIJ(:,j)',tt(i));
    end
    BC(:,i)=interp1(xcols,collvalues(i,:),[BB,L],'linear','extrap');

    barrierx(i+1)=pchip([max(BC(:,i)',0),collvalues(i,:)],[BB,L,xcols],Barrier);
    teller=0;
    while(abs(pchip([BB,L,xcols],[max(BC(:,i)',0),collvalues(i,:)],barrierx(i+1))-Barrier)>5e-3)&&(teller<10)
        barrierx(i+1)=interp1([pchip([BB,L,xcols],[max(BC(:,i)',0),collvalues(i,:)],barrierx(i+1)),max(BC(:,i)',0),collvalues(i,:)],[barrierx(i+1),BB,L,xcols],Barrier);
        teller=teller+1;
    end
end
BC(2,:)=BC(2,:)-K*exp(-r*tt);
BC(1,:)=max(BC(1,:),0);
```

```
U(end,2:end) = BC(2,:);
U(1,2:end) = BC(1,:);
%numerical integration
for i = 1:Nt
    B = (1+r*k)*eye(Nx-1,Nx-1) - 0.5*k*T2*eta^2/(h^2) - 0.5*k*lambda*(eye(Nx-1,Nx-1)*theta-D1)*T1/h;
    q(Nx-1)=(k/2*eta^2/(h^2)+k*lambda/(2*h)*(theta-(BB+Nx*h)))*U(Nx+1,i+1);
    U([BB:h:L]<=barrierx(i+1),i)=0;
    U(2:Nx,i+1) = B\(U(2:Nx,i)+q);

end
sigma=Haganformula(alpha,beta,gamma,rho,K,X_0*exp(r*T(end)),T(end));
[c,cdelta,p,pdelta]=BlackScholes(X_0,K,r,sigma,T(end));
[c2,cdelta2,p2,pdelta2]=BlackScholes(Barrier^2/X_0,K,r,sigma,T(end));
Checkvalue=c-(X_0/Barrier)^(1-2*r/(sigma^2))*c2

Optionvalue=pchip([BB:h:L],U(:,end),X_0)
```

A.1.7 CLV_MC.m

```
%by Jan Tijink 4203240
theta=0.1;
lambda=1.3;
X_0=1;
eta=0.25;
T=[0.05,0.25,0.5,1,2,3,4];

x=[-2.3344;-0.7420;0.7420;2.3344]';
N=length(x);
xcoll=zeros(length(T),N);
F=zeros(length(T),N);
for i=1:length(T)
    for j=1:N
        expectation=X_0*exp(-lambda*T(i))+theta*(1-exp(-lambda*T(i)));
        variance=eta^2/(2*lambda)*exp(-lambda*(2*T(i)))*(exp(2*lambda*T(i))-1);
        xcoll(i,j)=expectation+sqrt(variance)*x(j);
        F(i,j)=normcdf(xcoll(i,j),expectation,sqrt(variance));
    end
end
S0=1; r=0.03; h = S0/2000; r=0.03;
E=[0.001:(10-0.03)/400:5]*S0;
sigma=zeros(length(T),length(E),3);
V=zeros(length(T),length(E),3);
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
for i=1:length(T)
    for j=1:length(E)
        for b=1:3
        sigma(i,j,b)=Haganformula(alpha,beta,gamma,rho,E(j)+(b-2)*h,S0*exp(r*T(i)),T(i));
        V(i,j,b)=BlackScholes(S0,E(j)+(b-2)*h,r,sigma(i,j,b),T(i));
        end
    end
end
dVdK=(V(:,:,3)-V(:,:,1))./(2*h);
SIJ=zeros(size(F));
FS=zeros(length(T),length(E));
FS2=zeros(length(T),length(E));
FSS=zeros(size(F));
extradvdk=zeros(1,N);
V2=size(FSS);
h=0.001;
for i=1:length(T)
    FS(i,:)=1+exp(r*T(i))*dVdK(i,:);
        lower=((1+exp(r*T(i))*dVdK(i,:))>F(1,1));
        upper=((1+exp(r*T(i))*dVdK(i,:))<F(end,end));
        teller1=0;
        teller2=0;
        for j=1:length(E)
            if lower(j)==1
                if teller1==0
                    teller1=j-1;
                end
            end
            if upper(j)==0
                if teller2==0
                    teller2=j;
                end
            end
        end
        dVdK2=dVdK(i,teller1:teller2);
        E2=E(teller1:teller2);
        FS2(i,1:length(dVdK2))=1+exp(r*T(i))*dVdK2;
        SIJ(i,:)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,:));
        for j=1:N
            sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
            sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
            FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)
                -h,r,sigmafs2,T(i)))/(2*h);
            while abs(FSS(i,j)-F(i,j))>10^(-3)
                E2=[E2,SIJ(i,j)];
                V2=BlackScholes(S0,SIJ(i,j)+h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T
                    (i)),T(i)),T(i));
                V3=BlackScholes(S0,SIJ(i,j)-h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T
                    (i)),T(i)),T(i));
                dVdK2=[dVdK2, (V2-V3)/(2*h)];
                SIJ(i,j)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,j));
                sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
                sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
```

```
                FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
            end
        end
end
M=1000000;
TT=max(T); r=0.03; K = 0.9;
randoms=randn(1,M);
X=X_0*exp(-lambda*T(4))+theta*(1-exp(-lambda*T(4)))+sqrt(eta^2/(2*lambda)*(1-exp(-2*lambda*(T(4)))))*randn(1,M);
Xanti=X_0*exp(-lambda*T(4))+theta*(1-exp(-lambda*T(4)))-sqrt(eta^2/(2*lambda)*(1-exp(-2*lambda*(T(4)))))*randn(1,M);
V=exp(-r*(T(4))).*max(pchip(xcoll(4,:),SIJ(4,:),X) -K ,0);
V2=exp(-r*(T(4))).*max(pchip(xcoll(4,:),SIJ(4,:),Xanti) -K ,0);
value=(V+V2)*0.5;
Optionvalue=mean(value);
sigma=Haganformula(alpha,beta,gamma,rho,KK(j),S0*exp(r*T(4)),T(4));
[c,cdelta,p,pdelta]=BlackScholes(S0,KK(j),r,sigma,T(4));
Check=c
optionsd = std(value);
confidenceinterv = [Optionvalue -1.96*optionsd/sqrt(M), Optionvalue+1.96*optionsd/sqrt(M)]
```

A.1.8 CLV_MC_Barrier.m

```
%by Jan Tijink
theta=0.1;
lambda=1.3;
X_0=1;
eta=0.25;
T=[0.05,0.25,0.5,1,2,3,4];
x=[-2.3344;-0.7420;0.7420;2.3344]';
N=length(x);
NN=100;%number of simulation steps to see if we cross the barrier
xcoll=zeros(length(T),N);
F=zeros(length(T),N);
for i=1:length(T)
    for j=1:N
        expectation=X_0*exp(-lambda*T(i))+theta*(1-exp(-lambda*T(i)));
        variance=eta^2/(2*lambda)*exp(-lambda*(2*T(i)))*(exp(2*lambda*T(i))-1);
        xcoll(i,j)=expectation+sqrt(variance)*x(j);
        F(i,j)=normcdf(xcoll(i,j),expectation,sqrt(variance));
    end
end
S0=1; r=0.03; h = S0/2000; r=0.03;
E=[0.001:(10-0.03)/400:5]*S0;
sigma=zeros(length(T),length(E),3);
V=zeros(length(T),length(E),3);
beta=0.5;  alpha=0.2; rho=-0.9; gamma=0.2;
for i=1:length(T)
    for j=1:length(E)
        for b=1:3
        sigma(i,j,b)=Haganformula(alpha,beta,gamma,rho,E(j)+(b-2)*h,S0*exp(r*T(i)),T(i));
        V(i,j,b)=BlackScholes(S0,E(j)+(b-2)*h,r,sigma(i,j,b),T(i));
        end
    end
end
dVdK=(V(:,:,3)-V(:,:,1))./(2*h);
SIJ=zeros(size(F));
FS=zeros(length(T),length(E));
FS2=zeros(length(T),length(E));
FSS=zeros(size(F));
extradvdk=zeros(1,N);
V2=size(FSS);
h=0.001;
for i=1:length(T)
    FS(i,:)=1+exp(r*T(i))*dVdK(i,:);
        lower=((1+exp(r*T(i))*dVdK(i,:))>F(1,1));
        upper=((1+exp(r*T(i))*dVdK(i,:))<F(end,end));
        teller1=0;
        teller2=0;
        for j=1:length(E)
            if lower(j)==1
                if teller1==0
                    teller1=j-1;
                end
            end
            if upper(j)==0
                if teller2==0
                    teller2=j;
                end
            end
        end
        dVdK2=dVdK(i,teller1:teller2);
        E2=E(teller1:teller2);
        FS2(i,1:length(dVdK2))=1+exp(r*T(i))*dVdK2;
        SIJ(i,:)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,:));
        for j=1:N
            sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
            sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
            FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
            while abs(FSS(i,j)-F(i,j))>10^(-3)
                E2=[E2,SIJ(i,j)];
                V2=BlackScholes(S0,SIJ(i,j)+h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i)),T(i));
```

```
                    V3=BlackScholes(S0,SIJ(i,j)-h,r,Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i)),T(i));
                    dVdK2=[dVdK2, (V2-V3)/(2*h)];
                    SIJ(i,j)=interp1(1+exp(r*T(i))*dVdK2,E2,F(i,j));
                    sigmafs=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)+h,S0*exp(r*T(i)),T(i));
                    sigmafs2=Haganformula(alpha,beta,gamma,rho,SIJ(i,j)-h,S0*exp(r*T(i)),T(i));
                    FSS(i,j)=1+exp(r*T(i))*(BlackScholes(S0,SIJ(i,j)+h,r,sigmafs,T(i))-BlackScholes(S0,SIJ(i,j)-h,r,sigmafs2,T(i)))/(2*h);
                end
            end
end
B=0.6;
K=0.9;
M=1000;
TT=max(T); r=0.03; K = 0.9;
Dt=TT/(NN);
colvalue=zeros(1,N);
barrierx=B*ones(1,NN+1);
XX=zeros(NN,M);
XXanti=zeros(NN,M);
randoms=eta/(sqrt(2*lambda))*sqrt(-exp(-2*lambda*Dt)+1)*randn(NN,M);
XX(1,:)=X_0*exp(-lambda*Dt)+theta*(1-exp(-lambda*Dt))+randoms(1,M);
XXanti(1,:)=X_0*exp(-lambda*Dt)+theta*(1-exp(-lambda*Dt))-randoms(1,M);
xcolvalue=ones(NN,N);
colvalue=ones(NN,N);
U2=ones(NN,M);
U=ones(NN,M);
 for i=2:NN
     xcolvalue(i,:)=X_0*exp(-lambda*i*Dt)+theta*(1-exp(-lambda*i*Dt))+sqrt(eta^2/(2*lambda)*(1-exp(-2*lambda*(i*Dt))))*x;
     for k=1:N
         colvalue(i,k)=interp1(T,SIJ(:,k)',i*Dt);
     end
     XX(i,:)=XX(i-1,:).*exp(-lambda*Dt)+theta*(1-exp(-lambda*Dt))+ randoms(i,:);
     XXanti(i,:)=XXanti(i-1,:)*exp(-lambda*Dt)+theta*(1-exp(-lambda*Dt))- randoms(i,:);
     extremes_s=interp1(xcolvalue(i,:),colvalue(i,:),[min(XX(i,:)),max(XX(i,:))],'linear','extrap');
     barrierx(i)=pchip([max(extremes_s,0),colvalue(i,:)],[min(XX(i,:)),max(XX(i,:)),xcolvalue(i,:)],B);
     U(i,XX(i,:)<=barrierx(i))= 0;
     U2(i,XXanti(i,:)<=barrierx(i))= 0;
end
   Xval = [X_0*ones(1,M);XX];
  Xval2 = [X_0*ones(1,M);XXanti];
  Xend = Xval (end ,:);
  Xend2 = Xval2 (end ,:);
  A=cumprod(U);
  AA=cumprod(U2);
  V=A(end,:).*exp(-r*(TT)).*max(pchip([min(XX(end,:)),max(XX(end,:)),xcolvalue(end,:)],[max(extremes_s,0),colvalue(end,:)],Xend) -K ,0);
  V2=AA(end,:).*exp(-r*(TT)).*max(pchip([min(XX(end,:)),max(XX(end,:)),xcolvalue(end,:)],[max(extremes_s,0),colvalue(end,:)],Xend) -K ,0);
  Optionvalue=mean(V+V2)*0.5
  sigma=Haganformula(alpha,beta,gamma,rho,K,S0*exp(r*T(end)),T(end));
  [c,cdelta,p,pdelta]=BlackScholes(S0,K,r,sigma,T(end));
  [c2,cdelta2,p2,pdelta2]=BlackScholes(B^2/S0,K,r,sigma,T(end));
  Checkvalue=c-(S0/B)^(1-2*r/(sigma^2))*c2
  error=Optionvalue-Checkvalue
 optionsd = std(value);
 confidenceinterv = [Optionvalue-1.96*optionsd/sqrt(M), Optionvalue+1.96*optionsd/sqrt(M)]
```

A.1.9 Haganformula.m

```
function [sigma_implied] = Haganformula(alpha,beta,gamma,rho,E,forward,T)
%computes the implied volatility by the Hagan formula with parameters
f=forward;
z=gamma/alpha*(f*E)^((1-beta)/2)*log(f/E);
xz=log((sqrt(1-2*rho*z+z^2)+z-rho)/(1-rho));
sigma_implied1=alpha/((f*E)^((1-beta)/2)*(1+(1-beta)^2*(log(f/E))^2/24+(1-beta)^4/1920*(log(f/E))^4));
sigma_implied2=1+((1-beta)^2*alpha^2/(24*(f*E)^(1-beta))+rho*beta*gamma*alpha/(4*(f*E)^((1-beta)/2))+(2-3*rho^2)*gamma^2/24)*T;
sigma_implied=sigma_implied1*(z/xz)*sigma_implied2;
end
```