Abstract

In this thesis the credit spread model developed by Schönbucher (1998) is implemented. It is a discrete time intensity model based on the two-factor Hull-White model for default-free interest rates. Advantages of the model include possibility to calibrate to any term structures, allowing for any degree of correlation between the risk-free rate and the default intensity as well different recovery models.

Furthermore, the Credit Default Swap and the Default Digital Swap is priced in this framework and the robustness of the model is examined by finding a stability region for the generated prices.
Acknowledgements

First and foremost I would like to thank Alexander Herbertsson for introducing me to the subject of credit risk. Without all the help and input that you have given, this thesis would never been completed. I would also like to thank Peywand Sarban for devoting your time and for the interesting discussions on interest-rate models and interest derivatives which strengthened my basic understanding of the area. Finally I would like to thank my girlfriend, family and friends for the support given during the period spent on writing this thesis.
Contents

1 Introduction 1

2 Theory 2

2.1 Definitions ................................................. 2

2.1.1 Credit risk ........................................... 2

2.1.2 Recovery rate ......................................... 2

2.1.3 Credit derivates ....................................... 3

2.1.4 Credit Default Swap ................................. 3

2.1.5 Default Digital Swap ............................... 4

2.2 Intensity models ......................................... 4

2.3 Bond prices ............................................... 5

2.4 Dynamics .................................................. 6

3 Implementation 7

3.1 Term structure ............................................ 7

3.2 Calibration of dynamics parameters ....................... 8

3.3 Structure of interest trees .............................. 9

3.3.1 Building the risk-free tree .......................... 10

3.3.2 Calibration of the risk-free interest tree ............ 13

3.3.3 Building the intensity tree .......................... 14

3.3.4 Combining the trees .................................. 14

3.3.5 Calibration of the combined tree .................... 15

3.4 Pricing instruments within the framework ............... 16

3.4.1 Backwards induction algorithm ...................... 17

3.4.2 Implementing the CDS instrument ................... 18
4 Analysis

4.1 Verification of implementation ........................................... 20

4.2 Default Digital Swap .................................................... 21
  4.2.1 Implied default probability .................................... 22
  4.2.2 The fair DDS premium ........................................... 22
  4.2.3 Sensitivity analysis of the DDS price ...................... 22

4.3 Credit Default Swap ..................................................... 23
  4.3.1 Implied default probability .................................... 25
  4.3.2 The fair CDS premium ........................................... 25
  4.3.3 Sensitivity analysis of CDS price ...................... 25

5 Conclusion and Discussion ............................................. 28

References ............................................................... 30

A Source code .......................................................... 32
  A.1 plot_dds_price_vs_correlation.m ................................. 32
  A.2 build_tree.m ......................................................... 33
  A.3 get_interest_vector.m ........................................... 34
  A.4 get_branching_probabilities.m ................................. 35
  A.5 get_interest_tree.m .............................................. 36
  A.6 get_interest_tree_structure.m ................................. 37
  A.7 get.transition_probabilities.m ................................ 38
  A.8 get_calibrated_combined_tree.m ................................ 40
  A.9 get_calibrated_combined_tree.m ................................ 43
1 Introduction

In this master thesis we examine the credit spread model for pricing credit derivatives developed in [16]. The model is an extension of the now standard two-factor Hull-White interest model, where one factor represents the risk-free interest rate and the other the default intensity. (See [6]) The main usage of the model is to price credit derivatives, given that the underlying risky bond is traded in the market. This is useful if e.g. a new CDS contract is introduced to the market. Other possible areas of use, given some extensions not covered within this thesis, is pricing of hybrid instruments that depends on both the interest and the credit spread and convertible bonds.

The thesis is organized as follows. First, in Section 2, some of the theoretical prerequisites are presented, including several assumptions in the implementation. In Section 3 we explain on how the model is implemented and related topics. This is followed by a benchmark of the implementation and an analysis of the robustness of the model in Section 4, where we also price a Credit Default Swap and a Default Digital Swap. The input parameters are varied to find a region of stability.

Furthermore, here we will not focus on making comparisons of prices calculated within the model and prices observed in the market for different types of derivatives. For this to be done the dynamics parameters should be carefully calibrated to the market, something which is not done within this implementation.

Finally it is not within the scope of this thesis to explain in-depth the inner workings of different types of credit derivatives, even though a short explanation will be given where needed. For a deeper introduction the reader is instead referred to The J.P. Morgan Guide to Credit Derivatives [18], The Lehman Brothers Guide to Exotic Credit Derivatives [11] or Schönbucher’s Credit Derivatives Pricing Models [16].
2 Theory

In this part of the thesis a concise introduction to some of the most important definitions and concepts used within the credit derivative pricing model are presented. That is, after this section we will have a theoretical foundation for the actual implementation.

2.1 Definitions

Before delving into the subject of actual pricing of credit derivatives, it is important to first define some key terms used throughout the modeling. In this section established definitions will be presented, and in which aspect they are adapted in the model. A brief explanation of the two types of credit derivatives implemented within the model framework, the credit default swap and the default digital swap, will follow.

2.1.1 Credit risk

Financial institutions active in financial markets are subjected to a wide range of risk types, such as market risk, operational risk, liquidity risk and credit risk. Within this thesis, the focus is on the latter. A broad definition is that credit risk is the risk of default or of reductions in market value caused by changes in the credit quality of issuers or counterparties.¹

However, in this paper only the part of credit risk that arises due to defaults will be considered and once a default occur the bond is considered obsolete. Similar to most other credit risk literature, no attention will be given to restructured debt where a fraction still continues to trade.

2.1.2 Recovery rate

The recovery rate is simple to calculate ex-default and its definition is normally the bonds value immediately after default, given as a percentage of its face-value.² The problem arises due to the fact that the ex-default value is stochastic by nature and thus difficult to estimate prior to the event. Hence, an assumption must be made within the model.

Duffie and Singleton propose the use of a fractional recovery model. The fraction can either be a relative to the face-value of debt or to the market value of some reference bond. The rationale to use a fraction of the face value is due to legal reasons, where the issuer will have to liquidate their assets according to the covenants in the contract. Thus, they will be able to return a fraction of the face value of the bond. The drawback is that much empirical evidence highlights that this is not always the case. The other method, using a fraction of

¹Duffie & Singleton [5], Credit risk, p.4
²Hull [7], Options, Futures and Other Derivatives 6th edition, p.483
the market value of some reference bond leads to a more tractable recovery modeling. The latter is discussed in e.g. [5].

2.1.3 Credit derivates

The market for credit derivatives is rapidly evolving and there exist several types of instruments covering different aims. However, up to date, the most common class of credit derivatives are the so-called default products. These credit derivatives are instruments that have a payoff which is contingent on a predefined credit event. Such instruments create the opportunity to transfer, hedge and actively manage the exposure to credit risk. Prior to the development of credit derivatives institutions providing loans were stuck with their basket of credit risk until maturity of debt.

2.1.4 Credit Default Swap

The most common credit derivative today is the single-name Credit Default Swap (CDS), see [13]. The buyer of the contract makes periodic premium payments to the seller until the contract has expired at time T or until a credit event occurs, payments which are usually done quarterly. In the case of a default of the reference entity, the protection buyer receives the credit loss inferred by the reference entity. The cash flows are illustrated in Figure 1, see e.g. [7].

\[\text{Figure 1: Structure of a CDS contract.}\]

\[^{3}\text{Schönbucher [16], Credit Derivatives Pricing Models, p.8}\]
2.1.5 Default Digital Swap

The Default Digital Swap (DDS), which is a special case of the CDS, has perhaps one of the most simple payoff structures in the credit risk market. In the case of a credit event, represented by a default, the buyer of the protection will receive the payoff. If no default occurs, nothing will happen. Hence, the DDS is simple the CDS with the recovery rate set to zero in advance. In accordance with the CDS, the protection buyer will have to pay its counterparty a periodic fee determined within the contract, see [16]

2.2 Intensity models

Today, there exist two main approaches to model the dynamics driving a default event. The first one, the so called structural approach, originates from Merton, [12], and relay upon the financial status of the issuer of the bond. The basic principle is that a company will default when its debt exceeds its assets. The drawback of the model is that the necessary information is not fully publicly available, which makes it difficult to estimate the involved parameters.

Hence, in accordance with the principle of Occam’s razor a more simplistic approach with fewer input variables might be preferable in several instances. In 1995 Jarrow and Turnbull, [8], developed a model that neglected the actual capital structure, and instead assumed that defaults occur according to a Possion distribution with an intensity supplied exogenously. The intensity parameters are found by calibrating the model to relevant market instruments. In the intensity based model, the default time $\tau$ is modeled as

$$P(\tau \leq t + dt | \tau > t) = \lambda(t) dt$$  \hspace{1cm} (1)

where $\lambda(t)$ is the so called default intensity (for $\tau$). Note that $\lambda(\tau)$ is deterministic, which yields that

$$P(\tau > t) = e^{- \int_0^t \lambda(s) \, ds},$$  \hspace{1cm} (2)

A quick sketch how Equation (1) relates to Equation (2) is straight-forward, but for a more thorough proof see e.g. [16].

---

5First, $P(\tau \leq t + \Delta t | \tau > t) = \frac{F(t \leq s \leq t + \Delta t)}{F(\tau > t)} = \{ F(t) = P(\tau \leq t), f(t) = F'(t) \} = \frac{F(t + \Delta t) - F(t)}{1 - F(t)} = \lambda(t) \Delta t$

where $F(t)$ is the cumulative distribution function and $f(t)$ is the density function. Further,

$$\lim_{\Delta t \to 0} \frac{F(t + \Delta t) - F(t)}{\Delta t} \cdot \frac{1}{1 - F(t)} = \frac{f(t)}{1 - F(t)} = - \frac{d}{dt} \ln(1 - F(t)) = \lambda(t) \Rightarrow$$

$$\ln(1 - F(t)) = - \int_0^t \lambda(s) \, ds \Rightarrow 1 - F(t) = \exp(- \int_0^t \lambda(s) \, ds)$$

Hence,

$$P(\tau > t) = \exp(- \int_0^t \lambda(s) \, ds) \qed$$
One of the main disadvantages using this specification is that credit spreads are not allowed to fluctuate stochastically, something that is important in order to produce a realistic credit risk model. Using the doubly stochastic default suggested by Lando, see [9], the survival probability conditional on the information \( \{ \lambda(s) : 0 \leq s \leq t \} \), is given by

\[
P(\tau > t | \{ \lambda(s) : 0 \leq s \leq t \}) = e^{-\int_0^t \lambda(s) \, ds}
\]

so that

\[
P(\tau > t) = \mathbb{E}[e^{-\int_0^t \lambda(s) \, ds}]
\]

(3)

where the default intensity \( \int_0^t \lambda(s) \, ds \geq 0 \) now is a stochastic process.

To find \( P(\tau > t) \) in Equation (3) we must know the distribution of \( \lambda(t) \) for all \( s \leq t \). Hence some assumption regarding the dynamics of the default intensity must be proposed, see e.g. [5], [9]

### 2.3 Bond prices

The two main inputs, in the model which we treat in this thesis, are the prices of risk-free bonds and risky bonds that might default. Within this section zero-coupon bonds with a notional amount of $1 will be considered for simplicity. A risk-free bond is typically issued by a government, which is assumed not to be able to default, see [2]. Hence, the price is found by simply discounting at the risk-free rate so

\[
B(0, t) = \mathbb{E}[e^{-\int_0^t r(s) \, ds}],
\]

(4)

The reason for the expectation in Equation (4) is that interest rates are stochastic. Now, considering an investor buying a bond that might default, the investor would demand some premium for undertaking this risk. That is, the rational investor would only buy a similar defaultable bond if the price were lower which follows from applying a higher yield. The level of this higher discounting is found by using no-arbitrage arguments, where the expected returns by investing in either the risk-free bond or the defaultable bond should be equal. By using Equation (3), one can show that (see e.g. [9]) the price \( \bar{B}(0, t) \) of the defaultable bond at time \( t \) is given by

\[
\bar{B}(0, t) = \mathbb{E}[e^{-\int_0^t r(s) + \lambda(s) \, ds}] \]

(5)

where both \( r(t) \) and \( \lambda(t) \) are stochastic and where we have assumed zero recovery at default. Here \( \lambda(t) \) represents the additional term modeling the possible default of the obligor and is
therefore sometimes referred to as the credit spread. If \( r(t) \) and \( \lambda(t) \) are independent, the expectation can be separated into two factors. However, one of the strengths of this model is that correlation can be handled why the assumption is valid in general, see [17].

### 2.4 Dynamics

When deciding which type of dynamics the short-rate and the intensity should exhibit, several factors must be considered. Obviously, to model bond prices according to Equation (5), the specification must support at least two stochastic parameters. There exists a wide range of multifactor interest-rate models that qualifies under this condition, such as the Cox-Ingersoll-Ross (CIR) model and the extended Vasicek model. Further, it should be possible to include correlation between the risk-free interest rate and the default intensity. In this aspect, the CIR model is subordinate the extended Vasicek model since it only supports positive correlation whereas the Vasicek model allows any degree of correlation, see [3],[16]. Hence, we choose the dynamics of the default-free short rate to be given by the extended Vasicek model according to

\[
\text{dr}(t) = (k(t) - ar(t)) \, dt + \sigma(t) \, dW(t)
\]  

(6)

where \( r(t) \) is the default-free short rate, \( k(t) \) the level of mean reversion, \( a \) the speed of mean reversion, \( \sigma(t) \) the local volatility and \( W(t) \) a standard Wiener process. In the same way, the default intensity is modeled as

\[
\text{d\lambda}(t) = (\bar{k}(t) - \bar{a}\lambda(t)) \, dt + \bar{\sigma}(t) \, d\bar{W}(t)
\]  

(7)

where \( \lambda(t) \) is the default-intensity, \( \bar{k}(t) \) the level of mean reversion, \( \bar{a} \) the speed of mean reversion, \( \bar{\sigma}(t) \) the local volatility and \( \bar{W}(t) \) a standard Wiener process. Finally it is assumed that there is a constant correlation \( \rho \) between the two Brownian motions, that is

\[
dW \cdot d\bar{W} = \rho \, dt.
\]  

(8)

For positive values of \( \rho \), the credit spread will tend to increase as the interest-rate increases and for negative value of \( \rho \), the credit spread will decrease for a similar movement. Several studies indicate that the correlation is non-zero, see e.g. [17]. Hence, in order to reproduce more realistic prices, it may be important to include Equation (8) in the model.
3 Implementation

In this section we describe the practicalities of implementing the model introduced in Section 2. First the method used to retrieve the term structure observed in the market is described, followed by a description how to build the tree model. For a more thorough explanation, the reader is referred to Schönbucher [16],[17] and Hull et. al [6],[7] which covers the steps in more detail.

3.1 Term structure

For the dynamics described in Equation (6) and Equation (7) to be realistic, the parameter \( k(t) \) and \( \bar{k}(t) \) describing the level of mean reversion must first be calibrated to the market. Optimally, the parameters \( a \) and \( \bar{a} \) specifying the speed of mean reversion and the parameters \( \sigma \) and \( \bar{\sigma} \) specifying the short-term volatility should be calibrated to the market, but that additional complexity is neglected within this thesis.

In the tree model studied here, we need market bond prices for each discrete time point in which the model is calibrated. However, since the number of such market prices are much less than the number of time steps in the tree and that the model should be calibrated in each step, we must rely on a proper interpolation of the available market bond prices. Since splines are known to give a good continuous representation of the points, and no assumption regarding the mathematical form is necessary, a cubic spline interpolation is performed. The polynomial spline \( S : [a, b] \rightarrow \mathbb{R} \) consists of polynomial pieces \( P_i : [t_i, t_{i+1}) \rightarrow \mathbb{R} \) where

\[
a = t_0 < t_1 < \cdots < t_{k-2} < t_{k-1} = b.
\]

That is, the polynomial pieces apply to the different regions as

\[
S(t) = P_0(t) \quad , \quad t_0 \leq t < t_1
\]

\[
... 
\]

\[
S(t) = P_{k-2} \quad , \quad t_{k-2} \leq t < t_{k-1}.
\]

Finally, cubic refers to the fact that the spline is smooth up to its second derivate according to

\[
P_k^{(n)}(t_{k+1}) = P_{k+1}^{(n)}(t_{k+1}) \quad , \quad 0 \leq t < k - 2 \quad , \quad n = 0, 1, 2. \tag{9}
\]
3.2 Calibration of dynamics parameters

In order for the model to price the default risk properly, the dynamics parameters $a$ and $\sigma$ in Equation (6) and $\bar{a}$ and $\bar{\sigma}$ in Equation (7) must be calibrated to the market. In the case of the risk-free parameters, a good explanation is given by [7]. However, in this particular implementation, the calibration is not performed. Instead values from other credit risk literature such as [17] will be selected.

Within the Hull-White model there exist analytical expression for the zero-coupon bond price at time $t$ according to

$$P(t, T) = A(t, T; a, \sigma)e^{-B(t, T; a)\tau(t)}$$

where

$$B(t, T; a) = \frac{1 - e^{-\alpha(T-t)}}{\alpha}$$

and $A(t, T)$ is given by

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{yield_curve.png}
\caption{A cubic spline interpolation of the DM yield curve on July 8, 1994. The markers represent the actual values observed in the market.}
\end{figure}
\[
\ln A(t, T; a, \sigma) = \ln \frac{P(0, T)}{P(0, t)} - B(t, T) \frac{\partial \ln P(0, t)}{\partial t} - \frac{1}{4a^3} \sigma^2 (e^{-aT} - e^{-at})^2 (e^{2at} - 1). \tag{10}
\]

Now, if time \( t = 0 \), Equation (10) simplifies to

\[
\ln A(0, T) = \ln \frac{P(0, T)}{P(0, t)} - B(t, T) f(0) \tag{11}
\]

where

\[
f(t) = \frac{\partial \ln P(0, t)}{\partial t}.
\]

Assume that the bond prices \( \hat{P}(0, T) \) are observed in the market for time \( T = 1, 2, 3, ..., N \). From this set of prices it is possible to interpolate values of \( f(t) \) necessary to solve Equation (11). Preferably the spline interpolation method described in Section 3.1 could be used. The parameters \( a \) and \( \sigma \) are now found by performing a LS optimisation. See e.g. [14].

The dynamics of the intensity process in Equation (7) can be calibrated in a similar fashion. The survival probability for the risky bond from time \( t \) until \( T \) is given by

\[
P(t, T) = A(t, T; \bar{a}, \bar{\sigma}) e^{-B(t, T) \lambda(t)}.
\]

Hence, similar to Equation (5) the price of the risky bond, neglecting the correlation in Equation (8), is given by

\[
\hat{B}(t, T) = B(t, T) A(t, T; \bar{a}, \bar{\sigma}) e^{-B(t, T) \lambda(t)}. \tag{12}
\]

For a more extensive explanation of the derivation of theoretical bond prices, see e.g. [7] or [16], where the latter also includes correlation. For Equation (12) to result in a good fit, it is important that the underlying risky bond is traded without any liquidity premium. If that is the case, similar, but more heavily traded instruments could be used or historical data could be used to estimate the parameters.

### 3.3 Structure of interest trees

At this point the actual approach to the implementation of the credit spread model is described. The stochastic processes defined in Equation (6) and Equation (7) are both diffusion driven mean reverting processes. There are a wide range of trees in the literature that could be used to model such a behavior, but in this thesis the Hull-White model will be used. When
applied according to [16],[17], the implementation has five basic steps. These steps are, in order of execution, building the risk-free tree, calibration of the risk-free tree, building the intensity tree, combining the two trees and finally calibration of the combined tree.

### 3.3.1 Building the risk-free tree

In this subsection we describe how to build the risk-free interest tree. The same methodology will later on be applied when modeling the default intensity. It is important to realize that the dynamic part of Equation (6) can be modeled independently of the level of mean reversion $k(t)$. Hence, the interest tree can be built in two steps. First the tree is build according to the auxiliary process

$$dr^*(t) = -ar^*(t)dt + \sigma(t)dW(t)$$  \hspace{1cm} (13)

with the initial condition $r^*(0) = 0$. This process is later on calibrated to match the term structure observed in the market. The Hull-White model mimics the process in Equation (13) by a trinomial tree. The advantage of using a trinomial tree is that both the first and the second probability moment is matched, instead of just the first as with a binomial tree, see [10]. The according values for $dr^*$ are given by

$$E[dr^*] = Mr^* = (e^{-a\Delta t} - 1)r^*$$  \hspace{1cm} (14)

and

$$Var(dr^*) = V = \sigma^2(1 - e^{-2a\Delta t})$$  \hspace{1cm} (15)

for some short time interval $\Delta t$, see [6]. For short time intervals it is a good approximation to perform a Taylor expansion of Equation (14) and Equation (15).\(^6\) This results, if higher order terms are neglected, in the equations

$$E[dr^*] = -a\Delta tr^*$$  \hspace{1cm} (16)

and

$$Var(dr^*) = \sigma^2\Delta t.$$  

\(^6\)In the original paper by Hull-White, [6], the unexpanded version of the equations are used whereas Schönbucher, [16], uses the expanded versions. The result is slightly different expressions for the branching probabilities.
The spacing between the interest rate steps in the tree is set to

\[ \Delta r^* = \sqrt{3V} = \sigma \sqrt{3 \Delta t} \]

where \( V \) is given by Equation 15, i.e. the variance of \( dr^* \) in the time-interval \( \Delta t \).

Now, if the expected value and the variance for the standard branching pattern is matched along with the condition that probabilities sum to unity, the following set of equations is obtained

\[
\begin{align*}
p_u \Delta r + p_d \Delta r & = -a_j \Delta r \Delta t \\
p_u \Delta r^2 + p_d \Delta r^2 & = \sigma^2 \Delta t + a^2 j^2 \Delta r^2 \Delta t^2 \\
p_u + p_m + p_d & = 1
\end{align*}
\]

where \( p_u = p_u(n, j) \), \( p_m = p_m(n, j) \) and \( p_d = p_d(n, j) \) are the probabilities for branching up, middle or down from node \((n, j)\) where \( n \) is the time and \( j \) the node level. Note that at \((n, j)\), the time \( t = n \Delta t \). Since the equations constitute a fully determined linear equation system, the unique solution is found as

\[
\begin{align*}
p_u & = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 - a j \Delta t}{2} \\
p_m & = \frac{2}{3} - a^2 j^2 \Delta t^2 \\
p_d & = \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 + a j \Delta t}{2}
\end{align*}
\]

Until now, no restriction forces the branching probabilities to remain positive. For \( a > 0 \), the probabilities will turn negative for sufficiently large values of \( j \). The maximum and minimum node to retain standard branching are given by the inequality

\[ j_{\text{min}} < j_{\text{standard}} < j_{\text{max}} \quad , \quad j_{\text{standard}} \in \mathbb{N} \]

where

\[ j_{\text{max}} \geq \frac{0.184}{a \Delta t} \quad , \quad j_{\text{max}} \in \mathbb{N} \] (17)

and
\( j_{\text{min}} = -j_{\text{max}} \), \( j_{\text{min}} \in \mathbb{N} \). (18)

Hence, for the nodes \( j_{\text{max}} \) and \( j_{\text{min}} \) some other branching geometry must be applied. Hull and White, [6], suggest that the branching seen in Figure 3 b and Figure 3 c should be used as the maximum and minimum nodes are reached. In this way, there geometry is altered in such a way that the probabilities remain positive. The probabilities for the “up” branching are given by the equations

\[
\begin{align*}
    p_u &= \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 + a j \Delta t}{2} \\
    p_m &= -\frac{1}{3} - a^2 j^2 \Delta t^2 - 2a j \Delta t \\
    p_d &= \frac{7}{6} + \frac{a^2 j^2 \Delta t^2 + 3a j \Delta t}{2}
\end{align*}
\]

and for the “down” branching as

\[
\begin{align*}
    p_u &= \frac{7}{6} + \frac{a^2 j^2 \Delta t^2 - 3a j \Delta t}{2} \\
    p_m &= -\frac{1}{3} - a^2 j^2 \Delta t^2 + 2a j \Delta t \\
    p_d &= \frac{1}{6} + \frac{a^2 j^2 \Delta t^2 - a j \Delta t}{2}
\end{align*}
\]

The same formulas apply both for the risk-free tree and the intensity tree using the parameters given for each process. All three types of branchings can be seen in Figure 3 and an example of a tree can be seen in Figure 4.

Figure 3: The different types of branching possible within the Hull-White model. From left to right the standard, “up” and “down” branching are illustrated.
3.3.2 Calibration of the risk-free interest tree

Since the vertical steps are symmetrical around the initial condition, this will form the line of symmetry in the uncalibrated model. It is important to realize that each time-step is independent with respect to $k(t)$, and that $k(t)$ by definition is constant within that time interval. Hence, it is possible to retrieve the calibrated interest tree by adding a shift in each time-step according to

$$r_{jn} = r_{jn}^* + \alpha_n$$  \hspace{1cm} (25)

where $\alpha_n$ is a time-dependent offset and $r_{jn}^*$ is the interest rate in node $(n, j)$. The offset is found through forward induction, starting at $n = 0$ and iterating to the end of the tree. The calibration is done in discrete time, and the offset is determined such that the term structure retrieved earlier is matched exactly. The calibration is executed by using state prices $\pi_{jn}$, with payoff $\$1$ if node $(n, j)$ is reached, and zero otherwise. This can mathematically be formulated as

$$\pi_{jn} = \mathbb{E} \left[ 1 \{ r(n\Delta t) = r_j^* \} \prod_{m=0}^{n-1} e^{-r_m \Delta t} \right]$$

The first step, $n = 0$, when calibrating the interest tree is to determine $\alpha_0$. This is done by initializing $\pi_{0j}^* = 1$, and then determining the shift that prices $B_1$ correctly according to
\[ B_1 = e^{-r_0 \Delta t} \] (26)

The preceding steps are calibrated by first calculating the state price at time step \( n \) as

\[ \pi^n_j = \sum_{k \in \text{Pre}(n,j)} p^{n-1}_{kj} \pi^{n-1}_k e^{-r^{n-1}_k \Delta t} \]

where \( \text{Pre}(n,j) \) is the set of all immediate predecessors of node \((n, j)\). The offset at time \( n\Delta t \) is now found by matching the bond prices at time \((n + 1)\Delta t\) according to

\[ B_{n+1} = \sum_j \pi^n_j e^{-r^{n+1}_k \Delta t} = \sum_j \pi^n_{kj} e^{-r^m_k \Delta t} e^{-r^m_\alpha \Delta t} \] (27)

This procedure is iterated throughout the tree until all nodes have been matched to the term structure. Furthermore, it should be stated that the bond prices \( B_n \) in Equation (26) and Equation (27) are the interpolated prices found in Section 3.1.

### 3.3.3 Building the intensity tree

The tree for the default intensity is build using the same algorithm as is described initially when building the tree for the risk-free interest in Section 3.3.1. However, the intensity tree should not be calibrated at this stage. Furthermore, the branch to default could be incorporated at this stage but for simplicity it is incorporated directly into the combined tree.

### 3.3.4 Combining the trees

The actual combination of the risk-free interest tree and the intensity tree is a relatively straight-forward step in the implementation. The combined tree will extend into the interest rate dimension, the intensity dimension and the time dimension. The main calculation within this step is to calculate the transition probabilities. The transition matrix is calculated in two separate ways depending on whether the correlation is positive or negative. The probabilities in case of a positive correlation coefficient can be seen in Table 1 and the case of a negative correlation coefficient in Table 2. The rationale behind these transition matrices is that some of the probability mass has been move to the diagonal. By defining the matrices in this way, one can be assured the marginal probabilities will remain correct and sum up to unity. The parameter \( \epsilon \) is defined according to

\[ \epsilon = \begin{cases} \frac{\rho}{\rho^2} & \text{if } \rho > 0 \\ -\frac{\rho}{\rho^2} & \text{if } \rho < 0 \end{cases} \] (28)
3.3.5 Calibration of the combined tree

It may at this stage be valuable to summarize the current status of the implementation. In the first two steps a calibrated tree for the dynamics of the interest rate was built. Hence, it is necessary to not upset the calibration done in with respect to the risk-free interest dimension. In the third step an intensity tree was build to match the dynamics, but was never calibrated to market data at that stage. Finally, in the prior step the transition matrices for both positive and negative correlation between the risk-free interest and the default intensity were defined.

As just mentioned the combined tree should only be shifted in the intensity dimension, according to

$$\lambda_j^n = \lambda_j^{sn} + \bar{\alpha}_n.$$  \hspace{1cm} (29)

The procedure to find the shifts $\bar{\alpha}_n$ is similar to the one just described when calculating the risk-free shifts. It becomes slightly more complicated since the defaultable bond prices have to be discounted both with the risk-free interest as well as the intensity. Once again we initialize the first survival state price $\bar{\pi}^{00}_0 = 1$ and try to find the shift $\bar{\alpha}_0$ such that the equation

$$\bar{B}_1 = e^{-r_0^n \Delta t} \cdot e^{-\lambda_0 \Delta t}$$

The initial default state price will now be given as

$$\bar{\pi}^{00}_0 = 1 - e^{-\lambda_0 \Delta t}$$
where $\lambda_0^n$ is the initial default intensity valid for the first time interval of length $\Delta t$. The preceding steps are calibrated by first calculating the state price at time step $n$ as

$$\bar{\pi}^n_{ij} = \sum_{(n-1,k,l) \in \text{Pre}(n,i,j)} p_{kl,ij}^n n^{-1} e^{-r_l^n \Delta t} e^{-\lambda_k^n \Delta t}$$

where $\text{Pre}(n,i,j)$ is the set of all immediate predecessors of node $(n, k, l)$. Once again the $\bar{\alpha}_n$ are retrieved by solving the equation

$$\bar{\alpha}_n = \sum_{i,j} \bar{\pi}^n_{ij} e^{-r_j^n \Delta t} e^{-\lambda_k^n \Delta t} e^{-\bar{\alpha}_n \Delta t}.$$ 

At each iteration the default state price in that time-step can be retrieved according to

$$\bar{\pi}^n_{ij} = \bar{\pi}^n_{ij} (1 - e^{-\lambda_k^n \Delta t}).$$

The forward induction, or building of the tree, is now completed and the tree can now be used to price different types of derivatives. A graphical representation of a node from the combined tree branching both into the interest dimension and the intensity dimension is depicted in Figure 5. Each node $(n - 1, i, j)$ in the tree holds a value for the interest rate, the default intensity and a branch to default. Since the risk-free interest is calibrated within the two dimensional tree seen in Figure 4, the interest will remain constant for a given value of $j$. In general, this is not the case for the intensity since the intensity is correlated to the interest rate. However, if independency is assumed, also the intensity will remain constant for a given value of $j$. In this case, it is not necessary to combined the trees. Instead two independent two dimensional trees could be used, which would significantly reduce the number of values that has to be stored. It will still be necessary to store the default probabilities in a three dimensional tree though.

### 3.4 Pricing instruments within the framework

Now that the model is implemented and calibrated to some term structure, it is possible to price several types of credit derivatives such as the CDS, the Callable Default Swap and Credit Spread Options. Within this section we will first explain the backwards induction algorithm used from a general perspective. Furthermore, to exemplify the procedure it is also shown how the single-name CDS contract could be implemented and how it relates to its theoretical value.
3.4.1 Backwards induction algorithm

For simplicity, the same naming convention as is adopted by Schönbucher, [16], is used. Furthermore, we perform the calculation w.r.t. the protection buyer. Today’s value $V^0$ is found by a standard backwards induction scheme throughout the tree. Since fees and payoffs often are given in terms of the underlying instrument, it is important to first have it priced within the framework. Here we assume that it is the risky bond $\overline{B}$, where the notional amount $N_{\overline{B}} = 1$. The value $V^N_{ij}$ at the final time-step $N$ is then given by

$$V^N_{ij} = F^N_{ij}$$

where $F^N_{ij}$ is the value of the payoff given that no default of the risky bond occurred before maturity $T$ of the credit derivative. It is possible to price instruments depending on the risk-free rate, default intensity or, as for Credit Spread Options, the credit spread. For the remaining steps $n + 1 \rightarrow n$, an iterative procedure is used to calculate the value of the credit derivative throughout the tree. The value of node $(n,i,j)$, before default and premium payments are considered, is given by simply discounting using the branching probabilities according to

$$V''_{ij} = \sum_{k,l \in \text{Succ}(n,i,j)} p^{n}_{kl} e^{-r^d_{ij} \Delta t} V^{n+1}_{kl}$$

where $\text{Succ}(n,i,j)$ are the succeeding nodes. The value, neglecting early exercise, at node $(n,i,j)$ is now given by calculating the sum of the expected survival value $V''_{ij}$, expected
default value \( f^n_{ij} \) and the payoff \( F^n_{ij} \) payable within the time interval according to

\[
V'_{ij}^n = e^{-\lambda^n \Delta t} V''_{ij}^n + (1 - e^{-\lambda^n \Delta t}) f^n_{ij} + F^n_{ij}.
\]

The final term \( F^n_{ij} \), represents the periodic payment of the buyer that must be done in exchange for the protection. Hence, its value will mostly be negative. Finally, if early exercise is allowed, the value is given by

\[
V^n_{ij} = \max(V'_{ij}^n, G^n_{ij})
\]  

(30)

where \( G^n_{ij} \) is the value if exercised early. This procedure is now iterated all the way from \( n = N \) to \( n = 0 \). The value \( V^0 \) retrieved is the arbitrage free price today given the term structures observed when calibrating the model.

### 3.4.2 Implementing the CDS instrument

Now that the general guidelines for implementing an arbitrary credit derivative has been explained, we will focus on the single-name CDS contract. However, initially we will argue for the theoretical CDS price and how it relates to the tree model. First we assume that the nominal value \( N_B \) of the underlying bond \( B^* \) is \( N_B = 1 \) and that the maturity of the CDS contract is at time \( T \). The protection buyer pays a periodic fee \( s(T) N \Delta_n \) at times \( t_1, ..., t_{nT} \), where \( \Delta_n = t_n - t_{n-1} \) is measured as fractions of a year and \( s(T) \) is the constant credit spread for maturity in \( T \) years. If a default occurs at time \( \tau \in [t_n, t_{n+1}] \), the protection buyer will receive the amount \( (1 - \phi)N_B \) at time \( \tau \) where \( \phi \) is the recovery rate of the obligor. Finally, \( D(t, T) \) is a discounting factor given by

\[
D(t, T) = e^{-\int_t^T r_s \, ds}
\]

where \( r_t \) is the short-term risk-free rate at time \( t \). The net value of the contract is then given by

\[
V_t = V^{DL}_t - V^{PL}_t
\]  

(31)

where the default leg \( V^{DL}_t \) is given by

\[
V^{DL}_t = 1_{\{\tau \leq t\}} D(0, \tau)(1 - \phi)
\]  

(32)

and the premium leg \( V^{PL}_t \) follows as
\[ V_t^{PL} = \sum_{n=1}^{n_T} D(0, t_n) \Delta_n s(T) 1_{\{\tau \leq t\}}. \] 

(33)

For a more thorough explanation of the derivation of the CDS value, see [Herbertsson, working paper]. The arbitrage-free premium is now given by the credit spread \( s(T) \) which make the expected net value \( \mathbb{E}[V_t] = 0 \). Now, the only remaining step until the CDS can be priced within the framework is to define all the payoff function described in Section 3.4.1. It should also be mentioned that we assume that fractional recovery model is used and that the defaultable reference bond \( \bar{B}^* \) is priced within the tree. It then follows for node \((n, i, j)\) that the payoff given default is

\[ f_{ij}^n = 1 - (1 - q) \bar{B}_{ij}^{*n} \]

where \( q \) is the fraction of the bond value that is lost in the default. Hence, the recovery rate at node \((n, i, j)\) is \( \phi_{ij}^n = (1 - q) \bar{B}_{ij}^{*n} \). Since we assumed that default occur at the beginning of each time-step, the discounting in Equation (32) is neglected in the implementation. The payoff if survival until node \((n, i, j)\) is reached, is given by

\[
\begin{cases} 
F_{ij}^n = -s \Delta t & \text{if } n \Delta t \text{ is a payment time} \\
F_{ij}^n = 0 & \text{otherwise}
\end{cases}
\]

where \( s \) is the constant credit spread and it is assumed that all the payment times coincides with the time steps. If the step size in the time dimension do not match the payment times, the payments must be discounted according to what is done in Equation (33). Finally, since the CDS contract we price do not have an early exercise feature \( G_{ij}^n = -\infty \). By assigning negative infinity to \( G_{ij}^n \), we ensure that Equation (30) always return \( V_{ij}^n \). The indicator function in both the premium and the default leg is evaluated throughout the tree by using applying Equation (3). The DDS can be prices using the same rules, but with the recovery rate \( \bar{\phi} = 0 \).
4 Analysis

Within this part of the thesis we will try to verify that the model is correctly implemented by benchmarking the results to similar models. Moreover the results from implementing the CDS and DDS in the framework will be examined. The characteristics of the DDS will be examined by using dummy data, simply to get an initial view how the dynamics of the implementation of the model works. For the CDS instrument, the model will be calibrated to market conditions observed.

4.1 Verification of implementation

In order to verify that the model is properly implemented, it is valuable to have some benchmark of the model components. In the explanatory paper by Hull-White, [6], on how to use Hull-White interest rate trees to price derivatives on default-free bonds an example of a calibrated tree is given. Hence, it is possible to compare the values produced by that implementation with the results from the thesis implementation. Notice that the only comparison done at this stage is the calibration of the risk-free tree. That is, the $\alpha_n$'s of Equation (25) are compared between the two implementations. The main differences are that in the Hull-White paper a linear interpolation is performed to retrieve a continuous representation of the bond prices, opposite to a cubic spline interpolation performed within our implementation, see [6]. Another difference is that the comparing data is calculated using the expected interest rate move given as

$$\mathbb{E}[dr^*] = (e^{-a\Delta t} - 1)r^*$$  \hspace{1cm} (34)

whereas Schönbucher [16] applies a power-series expansion according to

$$\mathbb{E}[dr^*] = -a\Delta t r^*.$$  

Hence, to make a better comparison the non-expanded expectation in Equation (34) has been temporarily implemented when calculating the branching probabilities to achieve a more fair comparison. The term structure feed to the model is the DM yield curve of July 8, 1994. As can be seen from Table 3, the calculated shifts are almost equal. Some minor differences are expected since different interpolation schemes are used when retrieving a continuous approximation of the term structure. Hence, the model is assumed to be correctly implemented with respect to the risk-free yield curve. Since the intensity is calibrated similarly, the results supports that the intensity part of the algorithm is calibrated correctly as well.

In Figure 6 the $\Delta t$ has been decreased such that a new interest shift is calculated each day.
<table>
<thead>
<tr>
<th>Year</th>
<th>Hull-White</th>
<th>Implementation</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.09275</td>
<td>5.1509</td>
<td>-0.058173</td>
</tr>
<tr>
<td>1</td>
<td>6.50257</td>
<td>6.4623</td>
<td>0.040232</td>
</tr>
<tr>
<td>2</td>
<td>7.33932</td>
<td>7.5283</td>
<td>-0.18903</td>
</tr>
<tr>
<td>3</td>
<td>8.05381</td>
<td>8.3917</td>
<td>-0.33793</td>
</tr>
</tbody>
</table>

*Table 3: There are small variations between the implementation and the Hull-White paper. The interest rate shifts $\alpha_t$ are given in percentages using $\Delta t = \frac{1}{365} \text{[year]}$. 

*Figure 6: The continuous extension of the interest rate shifts has been plotted together with the $\alpha_n$'s found by Hull-White.*

### 4.2 Default Digital Swap

The first credit derivative that will be examined within the model framework is the DDS instrument which, as was mentioned in the introduction, is a special case of the CDS where the recovery rate is set to zero. Hence, it is amongst simplest credit derivatives imaginable and constitutes a good starting point for an initial examination of the model dynamics.

When performing the analysis of the DDS, the model will be calibrated to bond prices $B_t$ and $\bar{B}_t$ generated from a flat yield curve. That is, the risk-free interest is constant at 6% and the risky interest rate is constant at 9%. By using such a setup, the values found are easily benchmarked by other implementations. Furthermore, characteristics such as correlation of the model become more obvious when such simple data is used. The speed of mean reversion and short term volatility for the two mean reverting processes are given values.
that could typically be observed in the market. Hence, the parameters are not calibrated
to the market, but rather choosen similar to [17] where $a = 0.15$, $\bar{a} = 0.10$, $\sigma = 0.02$ and
$\bar{\sigma} = 0.01$. The maturity of the DDS is set to $T = 5$ years, with a step size of $\Delta t = \frac{1}{12}$ years
unless otherwise stated.

4.2.1 Implied default probability

As mentioned earlier, the reason for the higher yield on credit sensitive instruments is that
they might default. However, after making the assumptions on the dynamics of the default-
free interest and the default intensity and calibrating the processes to the observed term
structures, the distribution of the default intensity $\lambda(t)$ is known. Thus it is possible to
back-out an implied default probability term structure, also called the risk-neutral default
probability.

Since the credit spread model is discrete, a reasonable assumption is that the defaults occurs
in the beginning of each small time-step and that the intensity remains constant within this
time interval. The probability of default in the interval $[t, t + \Delta t]$ used in the tree is then
given by $1 - e^{-\lambda(t)\Delta t}$, where we have used Equation (3). To find the intensity at time $t = n\Delta t$
the intensities $\lambda^j_n$ must be averaged using the branching probabilities at that particular
time step, which is exactly what $\bar{\alpha}_n$ in Equation (29) represents. In Figure 7 the cumulative
default probability and survival probability have been plotted. As can be seen the default
probability increase at a constant rate due to the constant credit spread and zero correlation
used when calibrating the model.

4.2.2 The fair DDS premium

To find the “no-arbitrage” premium of DDS instrument just described, the value at time zero
is plotted against different premiums. Since the swap should be free to enter, the correct
premium is that which corresponds to the intersection of zero value. According to [7], the
periodic premium payments are usually done quarterly. Hence, assume that applies to the
DDS as well. In Figure 8 the value of the contract is plotted for various premiums. In this
case the fair price is about 281 b.p. If the payment is smaller than this, the contract has a
positive value for the protection buyer and the contract should be entered. Vice verse if the
premium is higher, the contract takes a negative value to the buyer and should be avoided.

4.2.3 Sensitivity analysis of the DDS price

Now that the fair premium has been found for the given conditions, it is important to make
sure how the value of the contract changes as its underlying parameters changes. Since
the model has the ability to adapt to different degrees of correlation between the risk-free
interest and the default intensity, it is essential that it is examined whether the correlation
is an important parameter. In Figure 9 the value of the contract has been plotted for values
Figure 7: The cumulative default and survival probabilities calibrated to bond prices generated using risk-free and risky yield curves leveled at 6% and 9%. The speed of mean reversion are $a = 0.15$ and $\bar{a} = 0.10$ and the short term volatility $\sigma = 0.02$ and $\bar{\sigma} = 0.01$. The time to maturity $T = 5$ and time interval $\Delta t = \frac{1}{12}$. No correlation between the interest and the intensity is used at this stage.

of the correlation in the interval $\rho \in [-0.9, 0.9]$. As can be seen there is a positive relation between the value of the contract to the protection buyer and the correlation.

4.3 Credit Default Swap

Now that the simpler DDS instrument has been analyzed, the attention is turned to the most commonly traded credit derivative of today. As of 2002 the CDS had a market share of 67%, which by far marks it the most traded credit derivative, see [13]. A recent survey suggests that the outstanding notional is approximately $26,006$ billions.\footnote{Survey by the British Bankers’ Association, http://www.isda.org/statistics/pdf/ISDA-Market-Survey-historical-data.pdf, 21/12/2006} The major difference from the analysis just performed, is that in this case the model will be calibrated to a realistic term structure. Here we use the term structures found by [1], which is the US Treasury yields and BBB-rated industrial spreads averaged between 05/1992 - 04/2004. The data is presented in Table 4. However, the dynamics parameters and the correlation are still given as inputs to the model. Similar to the DDS case, we set the parameters according to [17] where $a = 0.15$, $\bar{a} = 0.10$, $\sigma = 0.02$ and $\bar{\sigma} = 0.01$. The maturity of the CDS is set to $T = 5$ years, with a step size of $\Delta t = \frac{1}{12}$ years unless otherwise stated.
Figure 8: The value of the DDS contract for the protection buyer given different periodic payments. The premium is paid quarterly and the time to maturity $T = 5$. The no-arbitrage premium is when the value of the contract is zero, which occurs for premium $s = 0.0281$.

Figure 9: The value of the DDS depends on the correlation between the risk-free interest and the default intensity. The premium is set to the no-arbitrage $s = 0.0281$ found for $T = 5$ years.
Further it should be emphasized that the main use of the tree is not to price CDS contracts, since the CDS spread is easily observable in the market. However, other credit derivatives have the same source of uncertainty as the CDS why it can be used as a benchmark of the calibration. As mentioned in the introduction, the pricing of a CDS could be relevant if a newly issued contract is to be priced where the underlying bonds is traded in the market.

<table>
<thead>
<tr>
<th>Time [Months]</th>
<th>1</th>
<th>3</th>
<th>12</th>
<th>36</th>
<th>60</th>
</tr>
</thead>
<tbody>
<tr>
<td>US Treasury yield</td>
<td>3.71</td>
<td>3.98</td>
<td>4.30</td>
<td>4.93</td>
<td>5.30</td>
</tr>
<tr>
<td>BBB-rated industrial spread</td>
<td>-</td>
<td>-</td>
<td>1.05</td>
<td>1.06</td>
<td>1.11</td>
</tr>
</tbody>
</table>

*Table 4: US Treasury yields and BBB-rated industrial spreads averaged over the period 05/1992 - 04/2004, see [1]*

### 4.3.1 Implied default probability

Once again, the most important factor when pricing credit sensitive instruments is the default probability. However, when real term structures are feed to the calibration algorithm the implied default probability is given a more interesting dynamics. In Figure 10 the term structures used for calibration as well as the implied default probability has been plotted. As can be seen the cumulative default probability increases faster at the point in time where the credit spread increases which occurs between year 1 and 2 and close to the maturity of the contract at year 5.

### 4.3.2 The fair CDS premium

Similar to the case where the fair premium of the DDS where found, the value of the CDS contract is plotted for different premiums \( s \) in Figure 11. It turns out that the no-arbitrage premium, given a fractional loss \( q = 60\% \) and quarterly premium payments, is 65 b.p. As expected, the value of the contract increases for the protection buyer as the premium decrease, and vice versa.

### 4.3.3 Sensitivity analysis of CDS price

Now that the term structures are more realistic, it is interesting to examine the stability of the model. In Figure 12 the parameters \( a, \bar{a}, \sigma \) and \( \bar{\sigma} \) have been varied slightly around it equilibrium, or the fair contract price. By examining these plots, it is possible to conclude that the parameters \( a \) and \( \bar{a} \) will affect the models stability whereas the values of \( \sigma \) and \( \bar{\sigma} \) does not. A explanation for this behavior follows as a consequence of Equation (17) and Equation (18), which indicates \( j_{\text{max}} \) and \( j_{\text{min}} \) will be very low is \( a \) increase. In other words, the trees build will be extremely thin if \( a \) and \( \bar{a} \) increase resulting in a low resolution in the solution scheme.
Figure 10: Left: A spline interpolation of the data presented in Table 4. Right: The cumulative implied default probability given the observed term structures.

Figure 11: The value of the CDS contract for the protection buyer given different periodic payments. The premium is paid quarterly and the time to maturity $T = 5$. The no-arbitrage premium is when the value of the contract is zero, which occurs for premium $s = 0.0065$. 
When working with the CDS contract, there is an additional parameter compared to the DDS that must be examined. As mention in the theoretical section, there is a large portion of uncertainty regarding the recovery rate. In Figure 13 the value of the contract is plotted as a function of the fractional recovery rate. Once again the result seen in the graph is qualitatively what is expected. The value of the CDS decrease as the recovery rate decreases.

![Graphs showing value of CDS contract as a function of speed of mean reversion and short-term volatility.](image)

**Figure 12:** A sensitivity analysis performed around the point of stability found when calculating the fair premium of the CDS. Unless the parameter is the one varied, the values of the speed of mean reversion are \( a = 0.15 \) and \( \bar{a} = 0.10 \), the short term volatility \( \sigma = 0.02 \) and \( \bar{\sigma} = 0.01 \). The time to maturity \( T = 5 \), the time interval \( \Delta t = \frac{1}{12} \), correlation \( \rho = 0 \) and a fractional loss of 60%. 

---

27


Figure 13: The value of the CDS contract as a function of the recovery rate. The speed of mean reversion $a = 0.15$ and $\bar{a} = 0.10$, the short term volatility $\sigma = 0.02$ and $\bar{\sigma} = 0.01$. The time to maturity $T = 5$, the time interval $\Delta t = \frac{1}{12}$ and the correlation $\rho = 0$.

5 Conclusion and Discussion

The main task in this thesis was to implement the credit spread model developed by Schönbucher, [17], and to examine the robustness of the model. At this point, a working implementation has been achieved, and during the work a deeper understanding regarding the models strengths and limitations has been gained. It has also been showed how a CDS contract could be priced when the corresponding bond is traded in the market. Today most CDS contracts tend to be more liquid than the underlying instrument, but the model is of use when the contract is first issued.

It has been showed, especially for the CDS where slightly more complex term structures were used, that it is possible to calibrate the model by adding time-dependent shift in each step. Furthermore, it has been proved that the short-term volatility of the interest rate and intensity does indeed change the value of the credit derivative contract but not the stability to any noticeable extent. However, a more careful approach must be taken when selecting the values of the speed of mean reversion, $a$ and $\bar{a}$, since the prices found show evidence of instabilities. The source for the problem with the robustness was deduced to the fact that the interest trees become very thin if $a$ and $\bar{a}$ is increased. The remedy is simple in theory, where the solution should be to decrease the interval $\Delta t$. In practice one might run into computational difficulties if the steps become too small.

As shown in Section 3.4, it is relatively straight forward to implement credit derivatives of a large variety. Similar to the case of option pricing, it is necessary that the price of the instrument does not depend on the actual path of the interest and intensity. In such cases a Monte Carlo approach would be necessary.
If the model is intended for use in the actual credit market, where the speed of pricing is important, several optimizations should be done to improve usability. Since this has not been a crucial factor within this thesis, computer power has greatly been neglected. The apparent changes that should be done in that case, is to only build the trees for the time interval necessary to price the derivative. The bond prices needed to calibrate the model should at these new end-point be calibrated using analytical pricing formulas. Furthermore, the programming language should probably not be Matlab. Since many moments, such as building of the risk-free and intensity tree are similar, it would in most cases be beneficial to use an object-oriented programming approach. Finally, some improvement regarding the timing of default could be achieved by using the expected time of default instead of just assuming that the default occurs in the beginning of the interval.

Further possibilities of the model include pricing of hybrid instruments or convertible bonds. For hybrids, where the price depends on both interest rate and the credit spread, all the information necessary already exist at each node. For convertible bonds however, the model must be extended. This could probably be done by attaching a binomial tree representing the share price, [4], to nodes where conversion is possible.
References


[17] Schönbucher P., *A tree implementation of a credit spread model for credit derivatives* Bonn University, 1998
   J.P. Morgan and the RiskMetrics Group, Risk


A Source code

A.1 plot.dds_price_vs_correlation.m

%% Initialisation

clear all;
cf;

n_figures = 0;

%% delta_t is the discrete step length of the tree building algorithm, 
% denominated in years. Hence 1/252 correspond to one trading day, 1/12 
% one month, etc. The smaller delta_t is, the "thinner" the actual tree 
% will become.
delta_t = 1./5;
time_period = 5; % [years]

a_risk_free = 0.15;
sigma_risk_free = 0.02; % std(yield);

a_risky = 0.10;
sigma_risky = 0.01;

%% Term structure

interest_risk_free = 0.06;
interest_risky = 0.09;

[yield_risk_free, time] = get_yield_curve_flat(interest_risk_free, [0 time_period], inv(delta_t));
price_risk_free = get_risk_free_bond_price(yield_risk_free, time);

yield_risky = get_yield_curve_flat(interest_risky, [0 time_period], inv(delta_t));
price_risky = get_risk_free_bond_price(yield_risky, time);

%% Calculate DDS prices for different correlations

correlation_vector = -0.9:0.1:0.9;
dds_price = size(correlation_vector);
i = 0;
s = 0.0735;
for correlation = correlation_vector
    i = i + 1;
    disp(['Now at iteration: ' num2str(i) ' of ' num2str(length(correlation_vector))])

    % When correlation is changed, the tree must be rebuild since the
    % transition matrix is changed
    [ctree, state_prices, alpha_risk_free, alpha_risky, p_transition] = build_tree(delta_t,
        price_risk_free, a_risk_free, sigma_risk_free, ...
        price_risky, a_risky, sigma_risky);

    dds_price(i) = default_digital_swap(ctree, p_transition, time, delta_t, s, price_risky);
end

n_figures = n_figures + 1;
figure(n_figures);
plot(correlation_vector, dds_price);
xlabel('Correlation \rho'); ylabel('Value \hat{V}^0');
title('Default Digital Swap');

A.2 build_tree.m

% tree_build, build the 3-dimensional (combined) interest-intensity-time
% tree that is used to price different types of credit derivatives.
function [ctree, state_prices, alpha_risk_free, alpha_risky, p_transition] = build_tree(delta_t,
    price_risk_free, a_risk_free, sigma_risk_free, ...
    price_risky, a_risky, sigma_risky)

    % Build trees
    % The value of a should be estimated from market data. Don’t even know in which
    % range it should be at the moment. According to Schönbucher it should be
    % estimated from some liquid interest rate derivative such as floors, caps
    % or swaptions.
    % Information from Herbertsson: Use some very liquid derivative where there
    % exists an analytical expression, containing the parameters, for the
    % price.
    delta_r = sigma_risk_free * sqrt(3) * sqrt(delta_t); % [delta of interest]
    delta_l = sigma_risky * sqrt(3) * sqrt(delta_t); % [delta of intensity]

    % The maximum and minimum nodes of the interest tree and the intensity tree
    j_max = ceil(0.184 / (a_risk_free * delta_t));
    i_max = ceil(0.184 / (a_risky * delta_t));

    % The uncalibrated tree_risk_free can be represented by a single vector, where the
% entries are simply set to zero if unreachable early in the tree_risk_free.
T_risk_free = get_interest_vector(delta_r, j_max);
T_risky = get_interest_vector(delta_l, i_max);

% The probabilities can be calculated once
p_risk_free = get_branching_probabilities(a_risk_free, delta_t, j_max);
p_risky = get_branching_probabilities(a_risky, delta_t, i_max);

% The shift in interest needed to calibrate the tree_risk_free to market prices.
n_time_steps = length(1:length(time));

% The tree_risk_free could be saved to the disk at this stage to save some
% running time in the future
[alpha_risk_free, tree_risk_free] = get_interest_tree(n_time_steps, T_risk_free, price_risk_free,
%save '../Data/riskfree_tree.mat' alpha tree_risk_free ;
%load '../Data/riskfree_tree.mat';

% The transition matrix for movements in the 3d tree
p_transition = get_transition_probabilities(p_risk_free, p_risky, correlation);

% Combine risk-free and defaultable tree
[ctree, alpha_risky, state_prices] = get_calibrated_combined_tree(tree_risk_free, tree_risky,

A.3 get_interest_vector.m

function T = get_interest_vector(delta_r, jmax)

% To start with, the tree is simply a matrix
T = zeros(2*jmax+1, 1);

jmin = -jmax;
middle = jmax+1;

for j = jmin:jmax
    T(j+middle) = j*delta_r;
end
A.4 get_branching_probabilities.m

% get_branching_probabilities returns the probabilities with which the tree
% should move according to Schönbucher. Note that there is no algorithmic difference
% between calculating the risk-free probabilities and the risky
% probabilities.

function p = get_branching_probabilities(a, delta_t, j_max)

TOLERANCE = 0.0000001;
j_min = -j_max;

% Probability layout
% p(i) = [p_up; p_middle; p_down];
p = cell(2*j_max+1, 1);

% Standard branching
middle = j_max+1;
for j = j_min+1:j_max-1
    p{j+middle} = [1/6 + (aˆ2 * jˆ2 * delta_tˆ2 - a * j * delta_t) / 2;
    2/3 - aˆ2 * jˆ2 * delta_tˆ2;
    1/6 + (aˆ2 * jˆ2 * delta_tˆ2 + a * j * delta_t) / 2];
end

% "Up"-branching
p{1} = [1/6 + (aˆ2 * j_minˆ2 * delta_tˆ2 + a * j_min * delta_t) / 2;
    -1/3 - a^2 * j_min^2 * delta_t^2 - 2 * a * j_min * delta_t;
    7/6 + (a^2 * j_min^2 * delta_t^2 + 3 * a * j_min * delta_t) / 2];

% "Down"-branching
% end also corresponds to j_max+middle
p{end} = [7/6 + (a^2 * j_max^2 * delta_t^2 - 3 * a * j_max * delta_t) / 2;
    -1/3 - a^2 * j_max^2 * delta_t^2 + 2 * a * j_max * delta_t;
    1/6 + (a^2 * j_max^2 * delta_t^2 - a * j_max * delta_t) / 2];

for j = 1:length(p)
    if(sum(p{j}) - 1 > TOLERANCE)
        warning('get_branching_probabilities: Marignal probaility does not have unit sum!');
        sprintf('DEBUG: warning at j = %g', j)
    end
end

end
A.5 get_interest_tree.m

% get_interest_tree returns the basic structure of the individual tree. 
% There is calibration done in this function.
% This function has been verified to calculate the same results as the
% Excel sheet "Risk-free flat term-structure.xls".

function [alpha, interest_tree, state_price, info_cell] = get_interest_tree(periods, T)

alpha = zeros(1, size(T, 2));
interest_tree = cell(1, periods);
state_price = cell(1, periods);

% To see how many time each position in the matrix has been visited
info_cell = cell(1, periods);

% To know where in the transition matrix we are.
j_prob_middle = ceil(size(prob, 1) / 2);

for t = 1:periods-1 %time_period

% Keep track of the current process. Usefull when small delta_t is
% small.
if(mod(t, 100) == 0)
    sprintf('get_interest_tree: %g / %g', t, periods-1)
end

% The center of the interest rate tree is always used as the reference.
j_length = min(length(T), 2*(t-1)+1);
j_max = floor(j_length/2);
j_min = -j_max;
j_middle = j_max + 1;

% Calibrate the risk-free interest tree
if(t == 1)
    % Initialize the first state price to 1
    state_price(1, t) = 1;

    % Fit zero-coupon bonds
    alpha(t) = -log(price_risk_free(2))/delta_t;
    interest_tree(t) = T(j_prob_middle, 1) + alpha(t);
end

end
else

    state_price{1, t} = zeros(j_length, 1);
    info_cell{1, t} = zeros(j_length, 1);

    for j = j_min_pre:j_max_pre

        j_shift = get_shift(j, j_min, j_max, j_length, j_length_pre);

        for k = -1:1

            state_price{t}(j_middle+j_shift+j+k) = state_price{t}(j_middle+j_shift+j+k) + prob{j_prob_middle+j}(2+k) * state_price{t-1}(j_middle_pre+j) * exp(-interest_tree{t-1}(j_middle_pre+j) * if(prob{j_prob_middle+j}(2+k) * state_price{t-1}(j_middle_pre+j) * exp

            info_cell{t}(j_middle+j_shift+j+k) = info_cell{t}(j_middle+j_shift+j+k)

        end

    end

end

% Fit zero-coupon bonds
A = state_price{1, t}(:) .* exp(-T(j_prob_middle+j_min:j_prob_middle+j_max, 1))
alpha(t) = log(sum(A)./price_risk_free(t+1))/delta_t;
interest_tree{t} = T(j_prob_middle+j_min:j_prob_middle+j_max, 1) + alpha(t);

end

j_length_pre = j_length;
j_max_pre = j_max;
j_min_pre = j_min;
j_middle_pre = j_middle;

end

A.6 get_interest_tree_structure.m

% get_interest_tree_structure returns the basic structure of the individual tree.
% There is calibration done in this function.
%
function [interest_tree] = get_interest_tree_structure(periods, T)

interest_tree = cell(1, periods);

37
for t = 1:periods-1 %time_period

% Keep track of the current process. Useful when small delta_t is small.
if(mod(t, 100) == 0)
    sprintf('get_interest_tree_structure2: %g / %g', t, periods-1)
end

% The center of the interest rate tree is always used as the reference.
j_length = min(length(T), 2*(t-1)+1);
j_max = floor(j_length/2);
j_min = -j_max;
j_middle = ceil(length(T)/2);

interest_tree{t} = T(j_middle+j_min:j_middle+j_max, 1);
end

A.7 get_transition_probabilities.m

% get_transition_probabilities returns the transition matrix calculated according to Schönbucher (p.195). Neutral, positive and negative correlations have to be treated separately.

function p = get_transition_probabilities(p_risk_free, p_risky, correlation)

TOLERANCE = 0.0000001;

p = cell(length(p_risk_free), length(p_risky));
% p - risk-free probabilities, p’ - risky probabilities
% p{i} = [p’u*pd p’u*pm p’u*pu;
% p’m*pd p’m*pm p’m*pu;
% p’d*pd p’d*pm p’d*pu]

% Maybe this is a better layout of the matrix. Make sure that the correlation skews are set correctly though.
% p{i} = [p’u*pu p’u*pm p’u*pd;
% p’m*pu p’m*pm p’m*pd;
% p’d*pu p’d*pm p’d*pd]

for row = 1:size(p, 1)
    for col = 1:size(p, 2)
\[
p\{\text{row}, \text{col}\} = [p_{\text{risky}}(\text{col})(1) \cdot p_{\text{risk_free}}(\text{row})(1) \quad p_{\text{risky}}(\text{col})(1) \cdot p_{\text{risk_free}}(\text{row})(2) \\
p_{\text{risky}}(\text{col})(2) \cdot p_{\text{risk_free}}(\text{row})(1) \quad p_{\text{risky}}(\text{col})(2) \cdot p_{\text{risk_free}}(\text{row})(2) \\
p_{\text{risky}}(\text{col})(3) \cdot p_{\text{risk_free}}(\text{row})(1) \quad p_{\text{risky}}(\text{col})(3) \cdot p_{\text{risk_free}}(\text{row})(2)]
\]

end
end

if(correlation == 0)
    disp(['neutral correlation: ', num2str(correlation)]);
elseif(correlation > 0)
    disp(['positive correlation: ', num2str(correlation)]);
    epsilon = correlation / 36;
    for row = 1:size(p, 1)
        for col = 1:size(p, 2)
            p\{\text{row}, \text{col}\} = [p\{\text{row}, \text{col}\}(1, 1) + 5 \cdot \epsilon p\{\text{row}, \text{col}\}(1, 2) - 4 \cdot \epsilon \\
p\{\text{row}, \text{col}\}(2, 1) - 4 \cdot \epsilon p\{\text{row}, \text{col}\}(2, 2) + 8 \cdot \epsilon \\
p\{\text{row}, \text{col}\}(3, 1) - 1 \cdot \epsilon p\{\text{row}, \text{col}\}(3, 2) - 4 \cdot \epsilon]
        end
    end
elseif(correlation < 0)
    disp(['negative correlation: ', num2str(correlation)]);
    epsilon = -correlation / 36;
    for row = 1:size(p, 1)
        for col = 1:size(p, 2)
            p\{\text{row}, \text{col}\} = [p\{\text{row}, \text{col}\}(1, 1) - 1 \cdot \epsilon p\{\text{row}, \text{col}\}(1, 2) - 4 \cdot \epsilon \\
p\{\text{row}, \text{col}\}(2, 1) - 4 \cdot \epsilon p\{\text{row}, \text{col}\}(2, 2) + 8 \cdot \epsilon \\
p\{\text{row}, \text{col}\}(3, 1) + 5 \cdot \epsilon p\{\text{row}, \text{col}\}(3, 2) - 4 \cdot \epsilon]
        end
    end
end

% Make sure that all the probabilities where calculated correctly by testing that the marginal probability sum to unity.
for row = 1:size(p, 1)
    for col = 1:size(p, 2)
        if(sum(sum(p\{\text{row}, \text{col}\})) - 1 > TOLERANCE)
            warning('get_transition_probabilites: Marignal probability does not have un')
        end
    end
end
disp([row, col], sum(sum(p{row, col}))); end end

A.8 get_calibrated_combined_tree.m

function [tree_combined, alpha, state_prices] = get_calibrated_combined_tree(tree_risk_free,
INTEREST = 1;
INTENSITY = 2;
SURVIVAL = 1;
DEFAULT = 2;

% Create one cell for each time step. Each cell will then contain a new
% cell of the dimension {1, 2}, where the first dimension is used for the
% risk-free interest r and the second dimension is used for the default
% intensity.

tree_combined = cell(1, n_time_steps);
state_prices = cell(1, n_time_steps);
for t = 1:n_time_steps
    tree_combined{1, t} = cell(1, 2);
    state_prices{1, t} = cell(1, 2);
end

% To know where in the transition matrix we are.
j_prob_middle = ceil(size(p_transition, 1) / 2);
i_prob_middle = ceil(size(p_transition, 2) / 2);
alpha = zeros(1, size(tree_combined, 2));

for t = 1:n_time_steps-1 %time_period
    % Keep track of the current process. Useful when small delta_t is
    % small.
    if(mod(t, 100) == 0)
        sprintf('get_calibrated_combined_tree: %g / %g', t, n_time_steps-1)
    end

end
j_length = length(tree_risk_free(t));
j_max = floor(j_length/2);
j_min = -j_max;
j_middle = j_max + 1;

i_length = length(tree_risky(t));
i_max = floor(i_length/2);
i_min = -i_max;
i_middle = i_max + 1;

% Kolla om längden ökar. På så vis vet man hur trädet ska byggas. Om % storleken inte längre ökar kommer yttrarna att hoppa två steg in. % Annars är så inte fallet.

if(t == 1)
    tree_combined{1}{INTEREST} = tree_risk_free{1};

    % Initialize the first state-price to 1
    state_prices{1}{SURVIVAL} = 1;

    % Fit defaultable bond
    % BUG: Should this be interest in step 0 or 1? It should correspond % to risk-free bond price in step 1.
    alpha(1) = -(tree_risk_free{1} + inv(delta_t)*log(price_risky(2)));
    tree_combined{1}{INTENSITY} = alpha(t);

    state_prices{1}{DEFAULT} = 1 - exp(-tree_combined{1}{INTENSITY} * delta_t);
%tree_combined{1}{DEFAULT} = alpha(t);
else
    state_prices{t}{SURVIVAL} = zeros(j_length, i_length);
    state_prices{t}{DEFAULT} = zeros(j_length, i_length);

    for j = j_min_pre:j_max_pre
        for i = i_min_pre:i_max_pre
            j_shift = get_shift(j, j_min, j_max, j_length, j_length_pre);
            i_shift = get_shift(i, i_min, i_max, i_length, i_length_pre);

            % Sum all the state prices reachable from the preceeding time-step
            for k = -1:1
                for l = -1:1
                    state_prices{t}{SURVIVAL}(j_middle+j_shift+j+k, i_middle+i_shift+i+l) +
                    p_transition(j_prob_middle+j, i_prob_middle+i)(2+k, 2+l) *
                    exp(-(tree_risky(t-1)(i_middle_pre+i) + tree_risk_free(t)));
                end
            end
        end
    end
end

41
\[ A = 0; \]
\[ \text{for } j = j_{\min} : j_{\max} \]
\[ \text{for } i = i_{\min} : i_{\max} \]
\[ A = A + \text{state\_prices}(t)\{\text{SURVIVAL}\}(j_{\text{middle}}+j, i_{\text{middle}}+i) \times \exp(-\text{tree\_risky}(t)(i_{\text{middle}}+i)) \]
\[ \text{end} \]
\[ \text{end} \]
\[ \text{alpha}(t) = \text{inv}(\delta_{t}) \times \log(A / \text{price\_risky}(t+1)); \]
\[ \text{for } j = j_{\min} : j_{\max} \]
\[ \text{for } i = i_{\min} : i_{\max} \]
\[ \text{tree\_combined}(t)\{\text{INTEREST}\}(j_{\text{middle}}+j, i_{\text{middle}}+i) = \text{tree\_risk\_free}(t)(j_{\text{middle}}+j) \]
\[ \text{tree\_combined}(t)\{\text{INTENSITY}\}(j_{\text{middle}}+j, i_{\text{middle}}+i) = \text{tree\_risky}(t)(i_{\text{middle}}+i) \]
\[ \% \text{Include default state prices} \]
\[ \text{state\_prices}(t)\{\text{DEFAULT}\} = \text{state\_prices}(t)\{\text{SURVIVAL}\} \times (1 - \exp(-\text{tree\_combined}(t)\{\text{INTENSITY}\}(j_{\text{middle}}+j, i_{\text{middle}}+i)) \]
\[ \text{end} \]
\[ \text{end} \]
\[ \% \text{Have to update this tree as well since it’s used for the} \]
\[ \% \text{calibration. Must be outside the loop to avoid multiple addition.} \]
\[ \text{tree\_risky}(t) = \text{tree\_risky}(t) + \text{alpha}(t); \]
\[ \text{j\_length\_pre} = \text{j\_length}; \]
\[ \text{j\_max\_pre} = \text{j\_max}; \]
\[ \text{j\_min\_pre} = \text{j\_min}; \]
\[ \text{j\_middle\_pre} = \text{j\_middle}; \]
\[ \text{i\_length\_pre} = \text{i\_length}; \]
\[ \text{i\_max\_pre} = \text{i\_max}; \]
\[ \text{i\_min\_pre} = \text{i\_min}; \]
\[ \text{i\_middle\_pre} = \text{i\_middle}; \]
\[ \text{end} \]
\[ \text{tree\_combined}(n_{\text{time\_steps}})\{\text{SURVIVAL}\} = \text{zeros}(\text{j\_length}, \text{i\_length}); \]
A.9 get_calibrated_combined_tree.m

% credit_default_swap calculates the price of a CDS instrument under the % given term structure of interest rate and credit spread.

function [price] = default_digital_swap(ctree, p_transition, time, delta_t, s, price_risky)

%% Back-wards induction
INTEREST = 1;
INTENSITY = 2;
n_time_steps = length(1:length(time));
V = cell(1, n_time_steps);
V_prim = cell(1, n_time_steps);
V_biss = cell(1, n_time_steps);

% To know where in the transition matrix we are.
j_prob_middle = ceil(size(p_transition, 1) / 2);
i_prob_middle = ceil(size(p_transition, 2) / 2);

for t = n_time_steps:-1:1
    % Keep track of the current process. Useful when small delta_t is
    % small.
    if(mod(t, 100) == 0)
        sprintf('default_digital_swap: %g / %g', t, n_time_steps-1)
    end
    % Visa var dena på hur trädet ser ut från ctree.
j_length = size(ctree{t}{INTEREST}, 1);
j_max = floor(j_length/2);
j_min = -j_max;
j_middle = j_max + 1;

    % There is no difference of the matrix dimensions between the INTEREST
    % and INTENSITY.
i_length = size(ctree{t}{INTEREST}, 2);
i_max = floor(i_length/2);
i_min = -i_max;
i_middle = i_max + 1;
if(t == n_time_steps)
    V(t) = zeros(j_length, i_length);

for j = j_min:j_max
    for i = i_min:i_max
        % In this case the buyer must pay the seller an annual fee
        % of s. Hence, there is a negative sign in front of s.
        if(mod(time(t), inv(delta_t)) == 0)
            payoff = -s;
        else
            payoff = 0;
        end
        V(t){j_middle + j, i_middle + i} = payoff;
    end
end

else

    V(t) = zeros(j_length, i_length);
    V_prim(t) = zeros(j_length, i_length);
    V_biss(t) = zeros(j_length, i_length);

for j = j_min:j_max
    for i = i_min:i_max
        j_shift = get_shift(j, j_min_succ, j_max_succ, j_length_succ, j_length);
        i_shift = get_shift(i, i_min_succ, i_max_succ, i_length_succ, i_length);

        for k = -1:1
            for l = -1:1
                V_biss(t){j_middle+j, i_middle+i} = V_biss(t){j.middle+j, i.middle+i}
                    + p_transition{j_prob_middle+j, i_prob_middle+i}(2+k, 2+l) *
                    * V(t+1){j.middle+succ+j_shift+j+k, i.middle+succ+i_shift+l};
            end
        end
    end
end

for j = j_min:j_max
    for i = i_min:i_max
        f = 1;

        % In this case the buyer must pay the seller an annual fee
        % of s. Hence, there is a negative sign in front of s.
        if(mod(time(t), inv(delta_t)) == 0)

\[ F = -s; \]
\[ \text{else} \]
\[ F = 0; \]
\[ \text{end} \]

\[ V_{prim}(j_{middle}+j, i_{middle}+i) = \exp(-ctree(t)(\text{INTENSITY})(j_{middle}+j + (1 - \exp(-ctree(t)(\text{INTENSITY})(j_{middle}+j, i_{middle}+i) \cdot \delta_t))) \]

% Not necessary in this case, but will be needed in the case of early exercise.
\[ V(t)(j_{middle}+j, i_{middle}+i) = \max(V_{prim}(t)(j_{middle}+j, i_{middle}+i), \]
\[ \text{end} \]
\[ \text{end} \]

\[ j_{length\_succ} = j_{length}; \]
\[ j_{max\_succ} = j_{max}; \]
\[ j_{min\_succ} = j_{min}; \]
\[ j_{middle\_succ} = j_{middle}; \]

\[ i_{length\_succ} = i_{length}; \]
\[ i_{max\_succ} = i_{max}; \]
\[ i_{min\_succ} = i_{min}; \]
\[ i_{middle\_succ} = i_{middle}; \]
\[ \text{end} \]

% The initial price of the credit derivative
price = V(1);