VOLATILITY MODELLING IN THE OPTION AND BOND PRICING FRAMEWORK

Finn Kroijer

LYNGBY 1995
IMM-EKS-1995-34

IMM

ISSN 0909 6264

Trykt af IMM - DTU

Foreword

The increasing interest in the financial derivative markets has, in part, been a result of institutions becoming more confident in their use of advanced financial instruments, developing financial strategies which allow the element of risk to be monitored.
This development has lead to a growing interest in the mathematical methods, that can be applied to create a pricing framework. This project is concerned with mathematical methods used to price options and bonds. The project is therefore divided into two seperate sections, one with analysis of option pricing and one with the analysis of bond pricing.
One of the frequently used mathematical frameworks to price options is the Black-Scholes model. This model suffers from biases due to certain assumptions in its derivation.
This part of the project analyses a method to remove one of these biases which the Black Scholes model contains. The analysis is carried out with the initiative of Barlays de Zoete Wedd Securities Limited (BZW) in London.

The mathematical methods used to price bonds are partly based on the assumed behaviour of the interest rate. This part of the project analyses the implied interest rate from observed bond prices, using a specific bond pricing formula. The analysis is carried out with the initiative of Unibørs in Copenhagen, who have provided information and data to make it possible

Acknowledgements

I wish to thank Dr. Sam Howison at Oxford University and for his support and advice during the construction of the first part of this paper.

For the help on the revise of the first part of this project and the construction of the second part of this project I wish to thank Assoc. Prof., Ph.D Henrik Madsen at IMM, DTU. I would also like to thank Michael Preisel at Unibørs for his help and for supplying information.

Lyngby, November 1995

Finn Krøijer

Contents

Appendix

Chapter 1

Introduction to option pricing

1.1 Description of two basic option contracts

The simplest option contracts traded on the financial markets are the European Call and the European Put options.

A European Call option is a contract that gives the owner the rigth to,

- at a prescribed date in the future, named the expiry date (also called maturity),

- purchase a set amount of a prescibed asset known as the underlying asset,

- for a prescribed amount, called the strike price.

The owner posses the right to purchase, it is not an obligation to do so. In mathematcal terms the value of a European Call option at the expiry date is given by,

$$V_{Call} = max(S - E, 0)$$

where S=asset price and E=strike price.

A Put option gives the owner a rigth to sell a prescribed amount of the asset at the expiry date. The value of such an contract at the expiry date is given by,

$$V_{Put} = max(E - S, 0)$$

1.2 Biases of the Black Scholes equation

When pricing options, the Black Scholes (BS) equation is a well known framework, that gives a very flexible way of evaluating many different derivative products such as options. However in its derivation certain assumptions are made, some purely mathematical approximations, others about the market in which the underlying asset is traded. One of the assumptions about the market for the underlying asset is to be investigated in this part of the project. The derivation of the BS equation is based on the random walk that takes the form

$$\frac{dS}{S} = \sigma dX + \mu dt \tag{1.1}$$

where S is the price of the underlying asset, σ the volatility (vol) of the underlying asset and μ the growth of the asset. In the further derivation a portfolio Π consisting of one option and a number $-\Delta$ of the underlying asset is constructed. The jump in the value of this portfolio in one time-step is

$$d\Pi = dV - \Delta dS \tag{1.2}$$

Now applying Itô's lemma to V so that

$$dV = \sigma S\frac{\partial V}{\partial S}dX + (\mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t})dt \tag{1.3}$$

and formula (1.1) and (7.4) it is found that Π follows the random walk

$$d\Pi = \sigma S(\frac{\partial V}{\partial S} - \Delta)dX + (\mu S\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2 S^2\frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} - \mu\Delta S)dt.$$

To eliminate the random term dX, so that the risk is eliminated in $d\Pi$ we set $\Delta = \frac{\partial V}{\partial S}$. This means that the growth μ cancels out. Hence the BS equation takes no account of the growth of the asset itself. The σ term is the vol of the asset, and it is assumed to be constant. The consequence of these assumptions is, that the BS model cannot incorporate personal opinions of growth.

When pricing with BS these assumptions lead to mispricing. Observations in the

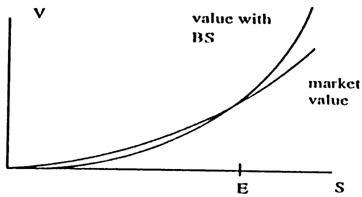


Figure 1.1: Biases with BS for Call options

market show that these assumptions lead to a tendency for Call options to be over priced in-the-money (when $S > E$) and underpriced out-the-money (when $E > S$), as it can be seen on figure 1.1. This tendency is vice versa for Put options.
This mispricing has a certain character that can be described by the relation between strikes and implied volatilities. By 'implied volatility' is meant the BS volatility implied by an option price observed in the market. The relation between strikes and implied volatilities forms a smile as is can be seen on figure 1.2, which for a set asset price exibits the underpricing out-the-money and over pricing in-the- money for a Call.

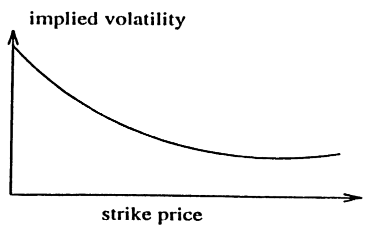


Figure 1.2: The volatility smile

1.3 Objective

The smile effect exhibits certain trends that give good background for deeper analysis. Two trends are particular noticeable. One trend can be described by the smile effect getting less conspicuous as time to expiry increases. The other trend is, as earlier mentioned, that the smile depends on whether the option is in- or out-the-money. These characteristics make a good foundation for investigating the possibilities of obtaining a deterministic function for vol dependent on asset price and time, to take some account of the mispricing by BS with constant vol.

A first step is to analyze the information than can be retrieved from the market. The actual observed prices in the market can be used to calculate the vol that should have been used, to price the options correctly with the BS equation. This gives a set of vol for different strikes and expiry dates. Through the analysis of the smile effect it shows that these vol are correlated.

Using this data as background, the task will be to deduce a deterministic vol function that leads to a BS valuation framework that takes the smile effect into account. The measure of success on this task is how close the values obtained with this framework, get to the prices observed in the market.

Such a framework can improve the evaluation of different derivative products, to construct a more consistent pricing framework. The last section of Part I will therefore contain the pricing of different options and comparing them with values obtained with constant vol.

When assuming vol to be deterministic instead of constant the BS equation changes. To see the effect of this, a short derivation is given in the next section.

1.4 Derivation of BS with deterministic volatility

Let V be the value of an option as a function of asset price S and time t : $V(S,t)$, so that $V(S+dS,t+dt)=dV$ expanded as a Taylor series is

$$dV = \frac{\partial V}{\partial S}dS + \frac{\partial V}{\partial t}dt + \frac{1}{2}\frac{\partial^2 V}{\partial S^2}dS^2 + ... \tag{1.4}$$

Now the random walk with deterministic vol is

$$\frac{dS}{S} = \mu dt + \sigma(S,t)dX \tag{1.5}$$

$$\begin{aligned} dS^2 &= (S\mu dt + S\sigma(S,t)dX)^2 \\ &= S^2\mu^2 dt^2 + S^2\sigma^2(S,t)dX^2 + 2S^2\mu\sigma(S,t)dtdX \ . \end{aligned}$$

Now $dX = O(\sqrt{dt})$ so to $O(dt)$

$$dS^2 = S^2\sigma^2(S,t)dX^2.$$

In fact $dX^2 \to dt$ for small dt, hence

$$dS^2 = S^2\sigma^2(S,t)dt. \tag{1.6}$$

Substituting (1.5) and (1.6) into (1.4)

$$dV = S\sigma(S,t)\frac{\partial V}{\partial S}dX + (S\mu\frac{\partial V}{\partial S} + \frac{\partial V}{\partial t} + \frac{1}{2}S^2\sigma^2(S,t)\frac{\partial^2 V}{\partial S^2})dt. \tag{1.7}$$

Setting up a portfolio by

$$\Pi = V - \Delta S \tag{1.8}$$

where $-\Delta$ = number of underlying assets, and letting $-\Delta$ be constant in one timestep dt we obtain

$$d\Pi = dV - \Delta dS.$$

Hence using (1.7) for dV and (1.5) for dS, a random walk for $d\Pi$ is obtained

$$d\Pi = \sigma(S,t)S(\frac{\partial V}{\partial S} - \Delta)dX + (rS\frac{\partial V}{\partial S} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + \frac{\partial V}{\partial t} - r\Delta S)dt. \tag{1.9}$$

To eliminate the random term dX and thereby the risk in the portfolio Π, we let $\Delta = \frac{\partial V}{\partial S}$, $r\Pi dt = d\Pi$ (r = interest rate) and using (1.8) we obtain BS with $\sigma(S,t)$ from (1.9)

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{1.10}$$

When comparing this equation with BS with constant vol, it can be seen that only the coefficient of $\frac{\partial^2 V}{\partial S^2}$ changes.

Chapter 2

Volatility analysis

2.1 Implied volatility

The only parameter in the BS pricing formulae that can not be observed in the market is the volatility of the asset price. To obtain this information with BS, the known option prices observed in the market are used to calculate what the vol should have been to equal the actual price observed (this procedure naturally assumes constant vol). To obtain the implied vol the following procedure is applied. The solution of formula (1.10) where V is the value of European Call option is

$$V(S,t) = SN(d_1) - Ee^{-r(T-t)}N(d_2) \tag{2.1}$$

where

$$N(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2}y^2} dy$$

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

$$d_2 = \frac{\log(S/E) + (r - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

For this purpose $S, Ee^{-r(T-t)}, V(S,t)$ are known constants. Now σ cannot be obtained explicitly but is approximated by solving the equation :

$$f(\sigma) = SN(d_1) - Ee^{-r(T-t)}N(d_2) - V(S,t) = 0$$

using Newton's method to give the iteration

$$\sigma_{n+1} = \sigma_n - \frac{f(\sigma_n)}{f'(\sigma_n)} \quad ,$$

where

$$f'(\sigma_n) = S\frac{\partial d_1}{\partial \sigma}\frac{\partial N}{\partial d_1} - Ee^{-r(T-t)}\frac{\partial d_2}{\partial \sigma}\frac{\partial N}{\partial d_2} = \frac{\partial \Pi}{\partial \sigma}$$

In this case the portfolio Π contains an option. This derivative gives the options sensitivity to volatillity, and it is normally referred to as the vega. This can for a European Call or Put option, be simplified to,

$$f'(\sigma_n) = S\sqrt{T-t}\frac{\partial d_1}{\partial \sigma}\frac{\partial N}{\partial d_1} \tag{2.2}$$

This method for retrieving the implied vol clarifies the relationship between the value of the option and the vol of the asset. In the following section the implied vol will take an essential part in analysis of volatility.

2.2 The volatility matrix

On the London Stock exchange, the index ftse100 is recognized to be a good measure of the general stock market level. The options traded on the ftse100 are some of the most liquid options on the English equity market and they are traded with the most broad variety of strikes and maturities. A careful analysis of the market prices of these options is therefore contributing to important information to estimate prices of new issued ftse100 options.

At BZW this information is analyzed through the implied vol of the ftse option prices, observed in the market. This procedure is carried out daily for call and put options for all different strikes and maturities. The procedure is based on closing prices on the business day before. This procedure is carried out for options on the ftse, DAX (Deutche Aktie Index) and other indices, but only options on the ftse100 index are relevant for this project. The implied vol are collected into a "Volatility Matrix" (VM) as seen on the table below. Due to Put - Call parity the same implied vol will be obtained for Put and Call. This example shows a sample VM from, say 24 May 1994. As mentioned earlier a VM dated 24 May 94 is actually based on closing prices from 23 May 94. Each element in the VM represents the implied vol of a contract.

	$\frac{S_0-E}{S_0} * 100$								
Month to maturity	-30	-20	-10	-5	0	5	10	20	30
1	31.0	25.8	22.0	17.5	17.0	15.6	15.0	13.9	12.9
3	24.7	21.8	18.6	17.3	15.9	15.7	14.6	14.2	13.2
12	22.3	19.9	18.2	17.4	16.8	16,4	16.0	15.4	14.3
24	21.7	19.6	18.4	17.6	17.0	16.6	16.3	15.9	15.0
36	21.2	19.6	18.4	17.7	17.3	16.9	16.9	16.6	15.9
48	20.6	19.3	18.4	17.6	17.3	17.3	17.2	16.8	16.4
60	20.4	19.3	18.6	18.1	17.8	17.8	17.4	17.4	16.8

Table 2.1: The volatility matrix

The x-label gives the strike relative to the asset price, so that for a Call option 30 can be translated as the option being 30% in-the-money, meaning that if the option expired that day it would pay 30% of the asset price. For a Put option this relation is vice versa so that -30 can be translated to 30% in-the-money, only 0 at-the-money is equal for Call and Put.

On the y-label is time to maturity in months. The options are issued with several different maturities. This means, that, eg the row of options with 36 month to maturity can be options just issued or options that have been recorded two times before in the VM, at 48- and 60 month maturity.

The index changes from day to day. This has the impact on the VM that the different strikes do not necessarily fit into the x-labels. Correspondingly there is a problem

with fitting into the y-labels as time goes by. This occurs because the options only are issued once a month, and so it is rare for there to be an option with exactly an integral number of months to run to fit the y-label.
Both of these biases are corrected for, so that the x- and y-labels are kept consistently the same regarding the time of recording. The correction methods are not relevant to this project, as only one VM is taken into consideration. The reason for this simplification is due to the fact that the VM does not change considerably over time, meaning that the impacts on the results retrieved from the VM in this project, only would give minor changes.

The VM informs about the implied volatilities for options with strikes in the range of ±30% relative to the asset price at the time of recording. This means, that over time, where strikes remain the same, the asset price changes. That effects the information the VM is based on. The implied volatilities noted under x-label 10 in the VM today is retrieved from options with a strike (and different maturities) relative to the current asset price, in a month time where the asset price have changed, the x-label 10 in the VM will based on implied volatilities for options with a strike that is different. The influence of this, is that the information that the VM is retrieved from changes as the asset price changes with time. To fit the VM this must satisfy the formula

$$100 * \frac{S_0(t) - E}{S_0(t)} = p \quad p = -30, -20, .., 30$$

so that the options that gives the information to fill the VM as time goes by must have strikes E that satisfy

$$E = S_0(t)(1 - \frac{p}{100})$$

The consequence of this, is that an option issued today with strike E_0, and where the asset price is S_0 will be "moving" around in the future VM's x-labels satisfying the relation

$$100 * \frac{(S_0 + \Delta S) - E_0}{S_0 + \Delta S} = p$$

The conclusion is that because this relation does not correspond to the linear relation there is between asset price and the value of an option, the VM, can over time only be used to price options with determined strikes, and not option that are described by their relation to asset price.

Chapter 3

Theoretical background

3.1 Surface fit for volatility matrix

For evaluating options with BS with a variable vol, the discrete 63 observations in the VM must be approximated by a smooth interpolation method that can minimize the differences between the observations and the derived continuous function. To make the best possible background for choosing a surface fit method, a 3D-graph of the VM is made, se graph 3.1 on next page .

The graph shows that fitting the data points to a model that is a linear combination (namely $a+bS+ct$) not would be advisable, rather a linear combination of M specified basis functions $X_k(S,t)$. The function can therefore be written as.

$$\sigma(x) = a_1 + a_2 X_2(x) + ... + a_k X_k(x), \quad x = (S,t)$$

The general form being

$$\sigma(x) = \sum_{k=1}^{M} a_k X_k(x))$$

It is important to notify that the basis functions can be nonlinear in x. The most flexible method to obtain the coefficients is the use of the least squares method ref [20]. By defining a cost function

$$\chi^2 = \sum_{i=1}^{N} \left[\frac{y_i - \sum_{k=1}^{M} a_k X_k(x_i)}{w_i} \right]^2$$

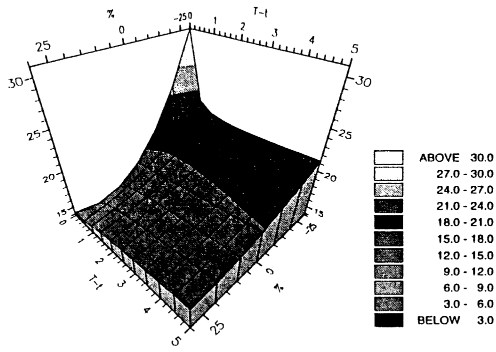


Figure 3.1: VM from 24 May 94

where N is the number of data points, in this case $N = 63$. w_i is the standard deviation on the measurement of the ith data point. In this case $\frac{1}{w_i}$ is used as a weight so that the data points with the least importance in the approximations will be given the largest values of w_i. This weight is relevant because certain areas in the VM often only is based on information from either very small or no trading volume. In the further analysis this means that the pricing of these options with a deterministic volatility have less importance. The w_i for this purpose can be observed in the table below.

	$\frac{S_0 - E}{S_0} * 100$								
Month to maturity	-30	-20	-10	-5	0	5	10	20	30
1	2	1	1	1	1	1	1	1	2
3	2	1	1	1	1	1	1	1	2
12	2	1	1	1	1	1	1	1	2
24	2	2	2	2	2	2	2	2	2
36	2	2	2	2	2	2	2	2	2
48	4	4	4	4	4	4	4	4	4
60	4	4	4	4	4	4	4	4	4

For the use of minimizing χ^2 define a 3 dimensional matrix of $N * M * L$ components.

$$A_{ijl} = \frac{X_j(x_{il})}{w_i}, \quad where\ x_{i1} = S, \quad x_{i2} = t$$

Also define a vector **b** of the length N by

$$b_i = \frac{y_i}{w_i}$$

and denote the M vector whose components are the parameters to be fitted, $a_1, ..., a_M$, by **a**. There are several different methods to find the minimum of χ^2, the method used in this case is a solution by the use of normal equations. The minimum of χ^2 is found where the derivatives with respect to all M parameters a_k vanish. This gives

$$0 = \sum_{i=1}^{N} \frac{1}{w_i^2} \left[y_i - \sum_{j=1}^{M} a_j X_j(x_i) \right] X_k(x_i) \quad k = 1, ..., M \tag{3.1}$$

Interchanging the order of summations, (3.1) can be written as

$$\sum_{j=1}^{M} \alpha_{kj} a_j = \beta_k \tag{3.2}$$

where the numbers

$$\alpha_{kj} = \sum_{i=1}^{N} \frac{X_j(x_i) X_k(x_i)}{w_i^2} \tag{3.3}$$

form an $M * M$ matrix, and

$$\beta_k = \sum_{i=1}^{N} \frac{y_i X_k(x_i)}{w_i^2} \tag{3.4}$$

a vector of size M. Equations (3.1) and (3.2) are referred to as *normal equations* of the least squares method.

They can be solved for the vector **a** of M parameters, and the system that has to be solved can be written as $\alpha \cdot a = \beta$ or more conveniently $(A^T \cdot A) \cdot a = A^T \cdot b$. This system can be solved by several different standard methods. In this case Gauss-Jordan elimination has been used.

The solution of the system have been written in a C++ computer program (see appendix A). The program has been constructed so that the number and shape of the basis functions are user defined. This gives the atmost flexibility in obtaining a close approximation. The variables S and t used in this approximation is from the VM.

$$S_{rel} = \frac{S_0 - S}{S_0} * 100 \tag{3.5}$$

the subscript rel is used for relative ($-30 < S_{rel} < 30$), and t is the time to maturity in month. The data points used for approximation, have been taken from a VM 24 May 94, the one in graph 3.1. Intuitively the graph 3.1, looks like some sort of paraboloid of the form

$$\sigma(S_{rel}, t) = a_1 + a_2 S_{rel} + a_3 t + a_4 S_{rel}^2 + a_5 t^2 + a_6 S_{rel} t$$

By changing the basis functions, one at a time, to satisfy a local behaviour such as the rapid increase as t tends to 0, and S_{rel} tends to -25%, an approximation of the following form was obtained :

$$\sigma(S_{rel}, t) = a_1 + a_2 S_{rel} + a_3 S_{rel}^2 + a_4 t^2 + a_5 \left(\frac{S_{rel}}{t+2}\right) + a_6 \left(\frac{S_{rel}}{t+2}\right)^2 \tag{3.6}$$

With the values for a given in table 3.1.

a_1	16.592607
a_2	-0.068240
a_3	0.001144
a_4	0.021546
a_5	-0.673210
a_6	0.056017

Table 3.1: optimal coefficients

The error of this approximation was

$$\sum_{i=1}^{N} |\sigma(S_{rel}, t) - y_i| = 18.1176 \qquad (3.7)$$

$$\frac{\sum_{i=1}^{N} |\sigma(S_{rel}, t) - y_i|}{N} = 0.2875 \qquad (3.8)$$

The information given in the VM only covers values of $-30 < S_{rel} < 30$ and $1 <$

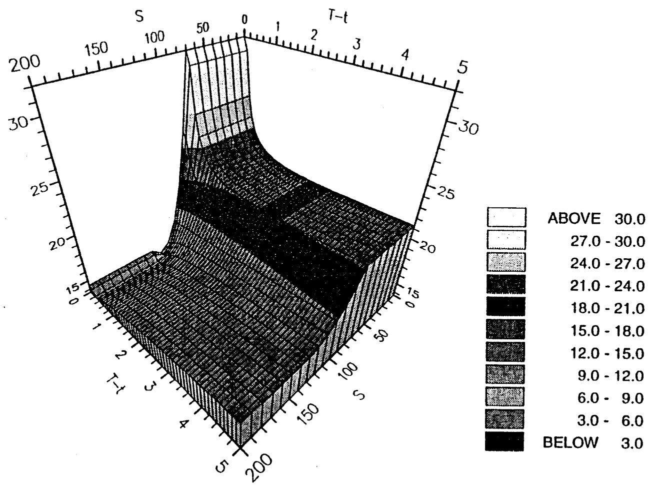


Figure 3.2: approximated vol

$t < 60$. For the valuation of options, $\sigma(S_{rel}, t)$ needs to be defined for $0 < S_{rel} < \infty$ and $0 < t < 60$. The most reasonable assumption is to let the surface continue with constant values at the boundaries given by the VM, so that

$$for \quad S_{rel} < -30 \quad : \sigma(S_{rel}, t) = \sigma(-30, t)$$

$$for \quad S_{rel} > 30 \quad : \sigma(S_{rel}, t) = \sigma(30, t)$$

$$for \quad t < 1 \quad : \sigma(S_{rel}, t) = \sigma(S_{rel}, 1)$$

$$for \quad t > 60 \quad : \sigma(S_{rel}, t) = \sigma(S_{rel}, 60)$$

These assumptions gives the surface on graph 3.2

3.2 Solving BS with deterministic volatility

The standard BS with constant vol can be transformed into the heat equation. In this case where vol is a variable a numerical solution after two basic transformations is preferred. From the derivation of BS with deterministic volatility, the equation (1.10) is given

$$\frac{\partial V}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV = 0 \tag{3.9}$$

For a European call the boundary and initial conditions are given by figure 3.3.

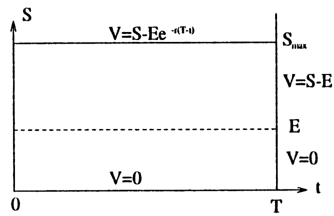


Figure 3.3: bc and ic for European Call

To remove the source term $-rV$ we transform with

$$V(S,t) = re^{-r(T-t)}u(s,t)$$

and obtain

$$\frac{\partial u}{\partial t} + \frac{1}{2}\sigma^2(S,t)S^2\frac{\partial^2 u}{\partial S^2} + rS\frac{\partial u}{\partial S} = 0$$

This is the Kolmogorov equation, and it is a backward parabolic partial differential equation. To make the equation more convenient for a numerical treatment, it is transformed into a forward pde by

$$t = T - \tau \qquad dt = -d\tau$$

which leads to

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2(S, T-\tau)S^2\frac{\partial^2 u}{\partial S^2} + rS\frac{\partial u}{\partial S} \tag{3.10}$$

The boundary and initial conditions changes after the transformation to figure 3.4 on next page. Formula (3.10) and the transformed conditions makes the background for the numerical approach.

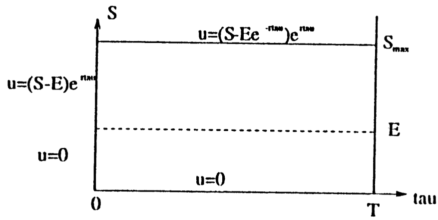


Figure 3.4: transformed bc and ic for European Call

3.2.1 Crank-Nicolson

The Crank-Nicolson scheme ref [15], [16] approximates the second space derivative by a half explicit and a half implicit difference. The derivative with respect to time is approximated by a forward difference. Now denoting the approximation to u by

$$U_j^n = u(j\Delta S, n\Delta\tau)$$

where $J\Delta S = S_{max}$ and $N\Delta\tau = T$

The Crank-Nicolson scheme for (3.10) can therefore be written as

$$\frac{\Delta_{+\tau}U_j^n}{\Delta\tau} = \frac{1}{2}S_j^2\frac{\frac{1}{2}((\sigma_j^{n+1})^2\delta_S^2U_j^{n+1} + (\sigma_j^n)^2\delta_S^2U_j^n)}{(\Delta S)^2} + rS_j\frac{1}{2}\frac{(\Delta_{0S}U_j^{n+1} + \Delta_{0S}U_j^n)}{\Delta S} \tag{3.11}$$

where

$$\Delta_{+\tau}U^n = \frac{U_j^{n+1} - U_j^n}{\Delta\tau}$$

$$\delta_S^2 U_j^n = \frac{U_{j+1}^n - 2U_j^n + U_{j-1}^n}{(\Delta S)^2}$$

$$\Delta_{0S}U_j^n = \frac{U_{j+1}^n - U_{j-1}^n}{2\Delta S}$$

Separating the scheme into U^{n+1} and U^n leads to

$$-\frac{\Delta\tau}{\Delta S}\frac{S_j}{4}\left((\sigma_j^{n+1})^2\frac{S}{\Delta S} - r\right)U_{j-1}^{n+1} + \left(1 + \frac{\Delta\tau}{(\Delta S)^2}\frac{1}{2}(\sigma_j^{n+1})^2S_j^2\right)U_j^{n+1} - \frac{\Delta\tau}{\Delta S}\frac{S}{4}\left((\sigma_j^{n+1})^2\frac{S}{\Delta S} + r\right)U_{j+1}^{n+1}$$
$$= \frac{\Delta\tau}{\Delta S}\frac{S}{4}\left((\sigma_j^n)^2\frac{S}{\Delta S} - r\right)U_{j-1}^n + \left(1 - \frac{\Delta\tau}{(\Delta S)^2}\frac{1}{2}(\sigma_j^n)^2S_j^2\right)U_j^n - \frac{\Delta\tau}{\Delta S}\frac{S}{4}\left((\sigma_j^n)^2\frac{S}{\Delta S} + r\right)U_{j+1}^n \quad (3.12)$$

3.2.2 Stability and convergence

Stability

The local stability of this scheme can be examined by dropping in a Fourier mode

$$U_j^n = \lambda^n e^{ij\zeta}, \quad \lambda = e^{\omega\Delta\tau}, \quad \zeta = m\pi\Delta S$$

Substituting this mode into (3.11), dividing by U_j^n and for simplicity assuming that $\sigma_j^{n+1} = \sigma_j^n$ gives

$$\lambda = 1 - (1+\lambda)2(\sigma_j^n)^2S_j^2\frac{\Delta\tau}{(\Delta S)^2}sin^2\frac{\zeta}{2} + i(1+\lambda)rS_j\frac{\Delta\tau}{\Delta S}sin\zeta \quad (3.13)$$

Now denoting

$$a = 2(\sigma_j^n)^2S_j^2\frac{\Delta\tau}{(\Delta S)^2}sin^2\frac{\zeta}{2}, \quad b = rS_j\frac{\Delta\tau}{\Delta S}sin\zeta$$

and after some simplification the stability condition $|\lambda| \le 1$ can be seen satisfied by

$$|\lambda| = \frac{1 - 4a + 4a^2 + b^2}{1 + 4a + 4a^2 + b^2} \le 1 \quad \forall a, b$$

hence the scheme is unconditional stable. The assumption that $\sigma_j^{n+1} = \sigma_j^n$ can be justified by the fact that for any realistic volatility function $\sigma(S,t)$ the difference will be bounded by a constant k satisfying $|\sigma_j^{n+1} - \sigma_j^n| \le k << 1, \ \ \forall j, n$ so that formula (3.13) still holds.

Convergence

To calculate the truncation error, u is expanded in a Taylor series. When using a six point Crank-Nicolson scheme, the series is expanded around the center of the six mesh points $(s_j, \tau_{n+1/2})$. Using the same superscript/subscript for u as well as U gives

$$u_j^{n+1} = \left[u + \frac{1}{2}\Delta\tau u_\tau + \frac{1}{2}(\frac{1}{2}\Delta\tau)^2 u_{\tau\tau} + \frac{1}{6}(\frac{1}{2}\Delta\tau)^3 u_{\tau\tau\tau} + ...\right]_j^{n+1/2} \quad (3.14)$$

$$u_j^{n} = \left[u - \frac{1}{2}\Delta\tau u_\tau + \frac{1}{2}(\frac{1}{2}\Delta\tau)^2 u_{\tau\tau} - \frac{1}{6}(\frac{1}{2}\Delta\tau)^3 u_{\tau\tau\tau} + ...\right]_j^{n+1/2} \quad (3.15)$$

Using these expansions for expanding (3.11), and assuming $\sigma_j^{n+1} = \sigma_j^n$ the following expressions are obtained

$$T_j^{n+1/2} := \frac{\Delta_{+\tau} u_j^n}{\Delta\tau} - a_j^n \frac{\frac{1}{2}(\delta_S^2 u_j^{n+1} + \delta_S^2 u_j^n)}{(\Delta S)^2} - b_j \frac{\frac{1}{2}(\Delta_{0S} u_j^{n+1} + \Delta_{0S} u_j^n)}{\Delta S} \quad (3.16)$$

where $a_j^n = \frac{1}{2}S_j^2(\sigma_j^n)^2$ *and* $b_j = rS_j$, further expansion gives

$$\Delta_{+\tau} u_j^n = \Delta\tau u_\tau + \frac{1}{24}(\Delta\tau)^3 u_{\tau\tau\tau} + ... \quad (3.17)$$

$$\frac{1}{2}(\delta_S^2 u_j^{n+1} + \delta_S^2 u_j^n) = (\Delta S)^2 u_{SS} + \frac{1}{12}(\Delta S)^4 u_{SSSS} + \frac{1}{8}(\Delta\tau)^2(\Delta S)^2 u_{SS\tau\tau}... \quad (3.18)$$

$$\Delta_{0S} u_j^{n+1} + \Delta_{0S} u_j^n = 2\Delta S u_S + \frac{2}{6}(\Delta S)^3 u_{SSS} + ... \quad (3.19)$$

Substituting formulae (3.17), (3.18) and (3.19) into (3.16), reduces the truncation error to

$$T_j^{n+1/2} = (\Delta\tau)^2(\frac{1}{24}u_{\tau\tau\tau} - a_j^n\frac{1}{8}u_{SS\tau\tau}) - (\Delta S)^2(a_j^n\frac{1}{12}u_{SSSS} + b_j\frac{1}{6}u_{SSS}) \quad (3.20)$$

This shows second order accuracy in space and time. Now proving convergence by introducing the maximum error

$$E^n : max\{|e_j^n|,\ j = 0, 1, ..., J\}, \quad e_j^n = U_j^n - u_j^n.$$

Substituting E^n into (3.20) and assuming there exist a bounds $M_{SSSS}, M_{SS\tau\tau}, M_{SSS}$ for $|u_{SSSS}|, |u_{SS\tau\tau}|, |u_{SSS}|$, a simple induction argument can be used to obtain

$$E^n \leq \left[(\Delta\tau)^2(\frac{1}{24}M_{\tau\tau\tau} - a_j^n\frac{1}{8}M_{SS\tau\tau}) - (\Delta S)^2(a_j^n\frac{1}{12}M_{SSSS} + b_j\frac{1}{6}M_{SSS})\right] t_F, \quad t \in [0, t_F]. \quad (3.21)$$

This shows that the scheme gives quadratic convergence in space and time.

3.2.3 The Thomas algorithm

Denoting the coefficients for U in (3.12) so that

$$-\alpha_j^{n+1}U_{j-1}^{n+1} + \beta_j^{n+1}U_j^{n+1} - \gamma_j^{n+1}U_{j+1}^{n+1} = d_j^n \quad (3.22)$$

where

$$d_j^n = \alpha_j^n U_{j-1}^n + \beta_j^n U_j^n + \gamma_j^n U_{j+1}^n.$$

This system of equations can now be solved with the Thomas algorithm. Now using (3.22) and the given initial and boundary conditions.

$$\text{bc}: \quad U_0 = 0 \qquad U_J = (S - Ee^{-r\tau})e^{r\tau}$$

$$\text{ic}: \quad S < E \quad U_j^0 = 0 \qquad S > E \quad U_j^0 = S - E.$$

With these conditions, a forward elimination given by

$$E_j^{n+1} = \frac{\gamma_j^{n+1}}{\beta_j^{n+1} - \alpha_j^{n+1}E_{j-1}^{n+1}}, \quad F_j^{n+1} = \frac{d_j^n + \alpha_j^{n+1}F_{j-1}^{n+1}}{\beta_j^{n+1} - \alpha_j^{n+1}E_{j-1}^{n+1}} \quad for\ j = 1, 2, ..J - 1 \quad (3.23)$$

subject to the condition $E_0^{n+1} = 0, \quad F_j^{n+1} = 0$ Now a backward substitution of the form

$$U_j = E_j U_{j+1} + F_j \quad for \quad j = J-1, ..., 2, 1 \ .$$

To avoid a build-up of errors in the back substitution, the following condition must be satisfied

$$\beta_j \geq |\alpha_j| + |\gamma_j| \quad \beta_j > 0$$

This conditions can be seen satisfied by substituting in the coefficients and obtaining the final condition

$$\left(1 + \frac{\Delta\tau}{(\Delta S)^2}\frac{1}{2}(\sigma_j^{n+1})^2 S^2\right) \geq \frac{\Delta\tau}{(\Delta S)^2}\frac{1}{2}(\sigma_j^{n+1})^2 S^2$$

The Crank-Nicolson scheme was implemented in a Fortran program (see appendix B).

3.2.4 Computing time and accuracy

To obtain a reasonable balance between computing time and accuracy of the nummerical approximation, a few experiments have been carried out.
The results have been benchmarked with results obtained with BS solved with formula (2.1) using NAG routines ref [17] for the cumulated normal distributions, so that the results are accurate to machine accuracy around 10^{-13}. The Crank-Nicolson (CN) scheme have been solved with constant vol.
A Call option with data

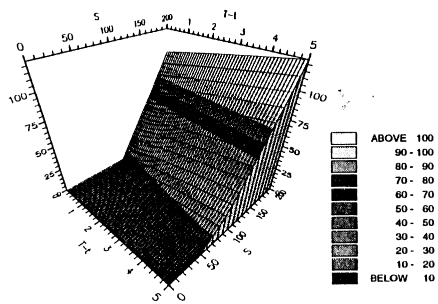
$$E = 100, \quad S_0 = 100, \quad T = 5 \ years \quad \sigma = 0.20$$

has been used for benchmarking. The reason for using an option with a 5 year maturity is, that when the vol function is to be derived, all values for options with the same strike but with the 7 different maturities in the VM can be solved in one numerical procedure. This means that using the longest dated option will be a good indicator for determining the best balance between number of steps and computing time.

The valuation 4.CN is a scaled version of the option, carried out to test the nature of the convergence. That naturally means that the result have been multiplied by 500 to match the magnitude of the other values.
The column 'comp time' is the time it took to run the program on a Sun SPARC station 1+, running with SunOS 4.1.3_U1B operating system.

Method	E	S_0	S_{max}	J (spacesteps)	N (timesteps)	$\frac{V_{BS}-V}{V_{BS}}$	comp time
1.BS	100	100	N/A	N/A	N/A	0	2 sec
2.CN	100	100	500	500	1000	-8.3305E-4	81 sec
3.CN	100	100	1000	500	1000	-7.5291E-4	81 sec
4.CN	0.2	0.2	1.0	500	1000	-8.3305E-4	81 sec
5.CN	100	100	500	1000	1000	-8.7421E-4	155 sec
6.CN	100	100	1000	1000	1000	-8.6121E-4	155 sec
7.CN	100	100	500	500	2000	-3.8464E-4	167 sec
8.CN	100	100	1000	500	2000	-3.0421E-4	167 sec

From the table it can be inferred that a good accuracy together with a reasonable low computing time is obtained with 2.CN. This set of data (S_0, S_{max}, J, N) will be used in the further analysis.
The graphs below show a Call and a Put with $E = 100$, $S_0 = 100$.



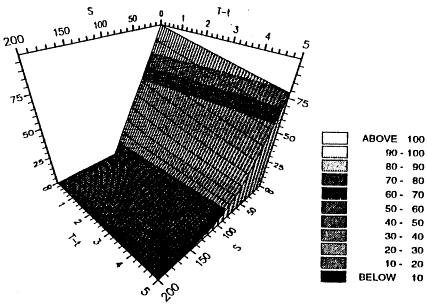


Figure 3.5: Value of European Call and Put

3.3 Put-Call parity

A Put and a Call option with the same strike are correlated through their underlying asset with the Put-Call parity, given by

$$S + P - C = Ee^{-r(T-t)} ,$$

where P and C are the value for Put and Call options.
To investigate whether this relation is satisfied when the BS is evaluated with a deterministic volatility, an example is carried out. This is accomplished with a Put and a Call option with $S_0 = 100$ and E=100 for the different maturities given in the VM. The vol func (3.6) with coefficients from table 3.1 has been used. Now for convenience define $L = S + P - C - Ee^{-r(T-t)}$, and noting maturities in months. For BS with constant volatility L=0. To avoid the influence of errors in the numerical solution between the different maturities, the relation T/N have been set constant.

T maturity	60	48	36	24	12	3	1
L	-1.314E-2	-1.316E-2	-1.316E-2	-1.36E-2	-1.315E-2	-1.315E-2	-1.315E-2
N timesteps	2000	1600	1200	800	400	100	33

The table shows that L remains constant regardless of maturity. The error that does occur is due to an error in the numerical solution, hence BS with deterministic volatility satisfies the Put-Call-parity.

Chapter 4

Volatility function

4.1 Method for deriving deterministic volatility

The VM informs about what the vol should have been if the option were to be priced correctly with BS.

It is a reasonable assumption that there is a correspondence between the different vol obtained in the VM (where vol is changing with E and time, and S is constant), and a vol dependent on S itself and time. The justification of this relationship is that in both cases, vol is an informer of the market level. It is therefore natural to exploit the possibilities for deriving an approximation of vol dependent on varying S and time, from the implied vol in the VM. Since there is no directly interpretations between the two different vol, it will be very important to find the most consistent method of approximation to avoid arbitrage possibilities. The VM contains 63 implied vol, this means that the actual values of all the different strikes and maturities can be obtained by the BS.

The problem can now be stated as follows.

At a certain time t=0, a set of option values are given

$$V_{ij}^{O}(S_0, E_i, T_j), \quad O \ for \ "observed" \tag{4.1}$$

$$with \quad V_{ij} = max(S - E_i) \ \ at \ \ t = T_j \tag{4.2}$$

for a Call. The task is to find a function $\sigma(S,t)$ such that solving

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

with final conditions (4.2) get as close as possible to the given values (4.1). One way of doing this would be to

$$minimize \ V_{min} = \sum_{j=1}^{7}\sum_{i=1}^{9}\left[V_{ij}^{O}(S_0, E_i, T_j) - V_{ij}(S, E_i, T_j, \sigma(S,t))\right]^2 \tag{4.3}$$

Certain assumptions are necessary for deriving such an approximation, these will be clarified underway.

The restriction on the use of $\sigma(S,t)$ in the valuation of options, is that it is only based on information in a certain area around S_0, since the VM only is defined for

$$-30 \leq \frac{S_0 - S}{S_0} \leq 30$$

The method used will first be applied to Call options with the nine different strikes, and afterwards to Put options with the same nine strikes. The reason for this procedure is that Call options with strikes that are higher than S_0 and with short maturities will have very small values. Consequently any method used will suffer from the fact that errors in the numeric approximation bias the solution. Using both Call and Put options, means that vol function will be valid for the whole surface, defined by the VM. The following approaches will all start using Call options. The philosophy is that if a method is not applicable for a Call option, the obtained vol function will lose its generality, and hence value.

Due to the fact that the use of Call and Put options will lead to corrections on the vol function in areas, that in each case is reasonably known beforehand, a weight could be introduced in formula (4.3). The reason for not doing so, is that these correction areas have to cover as much as possible of the whole volatility surface, and hence it is not possible to make any statements about how much each of the correction areas have to cover. Using a weighting, might therefore prevent this development.
Dividend is not incorporated in these valuations, although there is a dividend on the ftse it will not change the optimal vol function since the dividend for later valuations will be approximated as a continuous dividend yield.
The methods used will be bounded by certain restrictions, some more obvious than others.

4.1.1 Path dependence

Using BS with deterministic volatility leads to a path dependence that gives restrictions on further use of implied vol. To prove this a small example is given.
Using an arbitrary vol function given by $\sigma(S_{rel})$, S_{rel} from (3.5)

$$S_{rel} \leq -30 \rightarrow \sigma(S_{rel}) = 0.21$$

$$-30 < S_{rel} < 30 \rightarrow \sigma(S_{rel}) = \frac{(18 - \frac{7}{60}S_{rel})}{100} \tag{4.4}$$

$$S_{rel} \geq 30 \rightarrow \sigma(S_{rel}) = 0.145$$

the example is carried out for a European Call, with r=5%, E=100, $S_0 = 90$ and is solved with the Crank-Nicolson solver.

N (timesteps)	T (years)	V (value)	σ_{impl} (implied vol)
2000	1.00	4.5802	0.185504
1000	0.50	2.0325	0.185699
500	0.25	0.7303	0.185851

the σ_{impl} decrease as T increase, this can be extended to the limit

$$\sigma_{impl} \rightarrow 0.145 \quad as \quad T \rightarrow \infty$$

This limit occurs because as $S \rightarrow \infty$ $S_{rel} \rightarrow -\infty$, hence as can be seen from formula (4.4) $\sigma(S_{rel}) = 0.145$ so that the implied volatilty goes toward this value. This path dependence means that further use of implied vol in the derivation of a deterministic vol will lead to mispricing. Although this example is carried out for volatility being a function of asset price only, the small changes in σ_{impl} show that the use of the volatility function (3.6) with coefficients in table 3.1 is a good starting point for an iterative scheme.

4.1.2 The parametric form

The vol function (3.6) is used in a parametric form so that the variable $S_{rel} = 100 - S$ and $t = \tau * 12$. This parameterisation is necessary because the vol function is derived with the variables from the VM where $S_{rel} = \frac{S_0 - S}{S_0} * 100$ and $t = time\ to\ maturity$ in months. This rescaling gives the volatility surface shown on figure 3.2. When finding an optimal vol function, only values for one asset price $S_0 = 100$ is used in the optimization. The consequence of this is, that when using an optimal vol function in the further analysis, the parameters must be scaled so that $S_0 = 100$.

The derivation of the vol function means, that it essentially only will be applicable for investigating option values for one asset price at a time. Therefore when obtaining a whole surface of option values as a function of S and t, its accordance with, what would have been observed in the marked, will be difficult substantiate.

4.1.3 Secant method

For making the problem applicable for root seeking methods, define the function

$$V_{ij}^{O}(S_0, E_i, T_j) - V_{ij}(S, E_i, T_j, \sigma(S, t)) = 0$$

To avoid the derivation and calculation of $\frac{\partial V_{ij}(\sigma(S,t))}{\partial \sigma}$ when using Newton's method, the Secant method is used in the form

$$\sigma_{n+1} = \sigma_n - [V_{ij}^O - V_{ij}(\sigma_n)]\frac{(\sigma_n - \sigma_{n-1})}{(V_{ij}(\sigma_n) - V_{ij}(\sigma_{n-1}))}$$

In this approximation two initial guesses of vol functions are needed for the first iteration $n = 1$. The first iteration will then give 63 new vol (σ_2) that will be used as data points for a new surface fit accomplished by the least squares method described earlier. This surface will then take the σ_n position in the following iteration leading to 63 new vol etc. The procedure is continued until convergence is obtained. For making optimal guesses on initial vol functions the following values of V_{min} have been obtained. Only differences from the function (3.6) and values of $a_1, a_2, ..., a_6$ from table 3.1 have been noted. The table shows that the vol function used to approximate the

vol function	coefficients	V_{min}
formula (3.6)	table 3.1	5.5644
formula (3.6)	table 3.1, a1=1.05*16.5926	4.8951
formula (3.6)	table 3.1, a1=0.95*16.5926	22.7720
$\sigma = 0.20$	given in formula	54.5606
formula (4.4)	given in formula	9.0974

Table 4.1: initial parameters

implied vol in the VM gives a good initial guess for the secant method. This means that the formula (3.6) will be used to fit new data points from each iteration, unless the new data points are significantly different so that a new function is required.

Two specifications are needed in using the secant method for this problem.

1. To avoid the biases in the area where the value of option is very small, a minimum difference is required to reset the vol, else it will remain the same.
2. Only 15% of deviation between the old and new vol is allowed.

The results from the secant method have been written into table 4.2 below.

	least squares	σ_0	σ_1	σ_2	σ_3
a_1	16.5926	16.5926	1.05*16.5926	18.7684	20.9918
a_2	-0.0682	-0.0682	-0.0682	-0.0554	-0.00795
a_3	0.0011	0.0011	0.0011	0.00067	-0.00151
a_4	0.02154	0.02154	0.02154	0.00149	-0.02388
a_5	-0.6732	-0.6732	-0.6732	-0.4136	-0.17758
a_6	0.0560	0.0560	0.0560	0.00687	-0.0298
formula (3.7)	18.1176	N/A	N/A	23.6463	28.2487
formula (3.8)	0.2875	N/A	N/A	0.3753	0.4483
V_{min} (4.3)	5.5644	5.5644	4.8951	18.7962	60.8396

Table 4.2: results from secant method

It can be inferred from the table that the secant method gives a V_{min}, that diverges. Although only two iterations have been shown in this table (σ_2, σ_3) the last iteration gives a V_{min} that is ten times larger than V_{min} for σ_0, the method will therefore never converge. The change in the coefficients $a_1, a_2, ..., a_6$ shows that the method almost leads to a constant vol. The two rows containing the errors in using the least squares method (formula (3.7), (4.3)) shows that the vol function (3.6) makes a sufficient background for approximating the new data points, because these errors does not increase remarkable from the initial approach.

The conclusion is that the secant method not is applicable for this problem. The reason for this, is that the method corrects the vol at discrete points, so that the difference between a real value and an approximated value leads to a correction in the vol only at the point of valuation. The consequence is that the method eventually leads to a constant vol, this also shows to be the case.

4.1.4 Functional derivative

In obtaining the optimal initial vol functions for the secant method, it showed to be possible to attain a relation between the change of V_{min} with respect to change in a_1. This can more rigourously be stated to find some relation in the form.

$$\frac{\partial V_{min}}{\partial a_k} = \frac{1}{\epsilon}[V_{min}(\sigma(a_1,..,a_k+\epsilon,..,a_6)) - V_{min}(\sigma(a_1,...,a_6))]$$

where $\epsilon = \sqrt{u} = 10^{-7}$, u being machine accuracy. This approach can be obtained with the NAG-routine E04JAF ref [17], but it requires a continuous first and second derivative of the function $\sigma(a_1,..,a_6)$. That condition is not satisfied in this case, since the surface have been assumed to continue constantly out of the boundaries, hence the first and second derivatives are discontinuous at the boundaries.

A similar approach can be obtained by a small change in the computer program (appendix B). By changing the parameters $a_1, a_2, ..., a_6$ one at a time the derivative

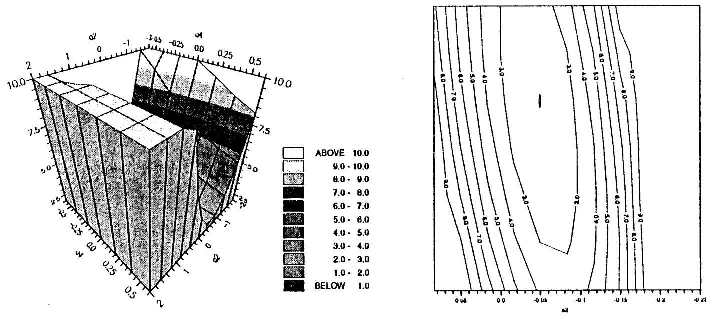
$$\frac{\partial^2 V_{min}}{\partial a_k \partial a_p} \quad k \neq p$$

can be approximated discretely, so that the optimal set of coefficients will be derived from a procedure that picks up the minimum V_{min} from each surfaces with the axis a_k, a_p. Figure 4.1 shows such a surface and its contour diagram, where an optimal relation between a_2 and a_4 is found for the vol function using a Put option. The contour diagram shows the V_{min} for the whole surface as this can not be seen on the graph. The coefficients are investigated in the area $a_2 \pm 200\%$ and $a_4 \pm 60\%$ The computing time required for calculating a surface with 7 different a_k and 11 different a_p, giving a surface of 77 V_{min}, is approximately 8 hours on a SUN SPARC station 1+. The optimal procedure would be to obtain all six derivatives in one procedure so that the absolute minimum of V_{min}, for the vol func (3.6) could be obtained. This is due to the amount of computing time required for this process not possible within the reach of this project. The restriction on this procedure is, that it is limited to only deal with one function, meaning that if the optimal vol function has another shape it will not be discovered by this method.

To obtain a vol function that is valid for the whole area of interest ($S_0 \pm 30\%$), the procedure needs to be carried out with Call options and Put options with the same strikes. The reason for this is, as mentioned earlier, that the procedure will not adjust the vol function for low S_0 using Call options, and vice versa for Put options. The final result will therefore be two vol functions where the vol func obtained using a Call will be used for pricing options where $S_0 > E$, and the vol func obtained with a Put is used for option pricing with $S_0 < E$. The coefficients for the vol function (3.6) (repeated below) leading to a minimal V_{min} is given in table (4.3).

$$\sigma(S_{rel}, t) = a_1 + a_2 S_{rel} + a_3 S_{rel}^2 + a_4 t^2 + a_5 \left(\frac{S_{rel}}{t+2}\right) + a_6 \left(\frac{S_{rel}}{t+2}\right)^2$$

This gave the following results for Call and Put options with $S_0 = 100$,



a) b)

Figure 4.1: a) V_{min} surface, b) Contourplot.

$$V_{min}(Call) = 3.096077 \quad V_{min}(Put) = 2.760749$$

The tables 4.4 a,b on the next page show

$$\frac{V_{ij}^O - V_{ij}(\sigma(S,t))}{V_{ij}^O}$$

	Coef from table 3.1	Call	relative	Put	relative
a_1	16.592607	17.007422	1.025	17.00742	1.025
a_2	-0.068240	-0.068240	1.0	-0.075064	1.1
a_3	0.001144	0.001144	1.0	0.001144	1.0
a_4	0.021546	0.019205	0.891	0.020357	1.058
a_5	-0.673210	-0.693406	1.03	-0.693406	1.03
a_6	0.056017	0.056017	1.0	0.056017	1.0

Table 4.3: optimal coefficients for V_{min}

where numbers $\leq$ 0.01 have been marked with bold. The sum of the relative error gave,

$$\sum_{j=1}^{7}\sum_{i=1}^{9}\frac{V_{ij}^{O}-V_{ij}(\sigma(S,t))}{V_{ij}^{O}} = 3.794378 \quad Call$$

$$do = 9.111E9 \quad Put$$

The enormous difference in the sum of relative errors occurs solely because of the error in the upper left corner of table 4.4 b, which is deeply out of the money with a short maturity, and therefore the least relevant value to minimize with the Put option. Comparing the numbers marked with bold in the two tables shows where each of the vol functions are applicable. The area around at-the-money with short maturity, does not go below 1%, hence none of the vol function can be used in this area.

In figure 4.2, the graphs of the two tables can be seen. It must be notified that either of the two axis are equidistant. This means that the axis "$T-t$" in the graphs going from 1 to 7 correspond to the 7 different maturities in the VM. The "100-S" axis going from -30 to 30 has the 9 different strikes in the VM, with equal distances in the graphs. The reason for not interpolating this into equal distances, is that the quite rapid changes in the short maturities not would be noticeable. Only relative errors smaller than $\pm 10\%$ have been shown on the graphs.

a	$\frac{S_{const}-E}{S_{const}} * 100$, Call								
T-t	-30	-20	-10	-5	0	5	10	20	30
1	0.97966	0.82228	0.20644	-.08390	-.10038	-.01921	-.00391	-.00155	-.00101
3	0.49376	0.21010	-.01513	-.05354	-.05184	-.01832	-.00587	-.00124	-.00081
12	0.14889	0.01466	-.01753	-.02362	-.02211	-.00906	-.00647	-.00217	-.00098
24	0.12056	0.02425	-.00789	-.01980	-.01660	-.00685	-.00371	-.00250	-.00162
36	0.08468	0.01101	-.00702	-.00870	-.00992	-.00618	-.00253	-.00122	-.00114
48	0.05250	0.00224	-.00793	-.01002	-.00729	0.00092	0.00220	0.00087	-.00046
60	0.03436	0.00211	-.00274	-.00681	-.00277	0.00340	0.00234	0.00331	0.00063

b	$\frac{S_{const}-E}{S_{const}} * 100$, Put								
T-t	-30	-20	-10	-5	0	5	10	20	30
1	0.00148	0.00280	0.00692	-.00312	-.10561	-.32934	-1.0481	-83.125	-.9E+09
3	0.00290	0.00453	0.00049	-.01507	-.06686	-.11297	-.20395	-1.2207	-47.483
12	0.01302	0.00287	-.00848	-.02180	-.04162	-.03183	-.05349	-.11156	-.88810
24	0.03084	0.01064	-.00834	-.03217	-.04387	-.02676	-.02010	-.05564	-.31360
36	0.03742	0.00683	-.01177	-.01961	-.03274	-.02905	-.01206	-.00104	-.04404
48	0.03326	-.00017	-.01807	-.03182	-.02953	0.01045	0.03078	0.05203	0.03640
60	0.02851	-.00042	-.00965	-.02565	-.01318	0.02702	0.03197	0.10323	0.10591

Table 4.4: a,b Results from optimal vol function Call, Put

The tables and graphs shows that minimization with the Call option gives a much more homogeneous surface than the Put option. This difference occurs for two reasons, firstly because more time was spent on optimizing the vol function for the Call option, and secondly, as it can be seen on the graphs 3.5, the value of a Put option in the limit is $V(Put) \to 0 \;\; as \;\; t \to \infty$ whether for a Call it is $V(Call) \to \infty \;\; as \;\; t \to \infty$. This means that the area where the options has a value that can be used to correct the vol function is smaller for a Put- than a Call option. This procedure of minimization was stopped, when these reasonable results was attained, meaning that a better set of coefficients giving a smaller V_{min} probably exists.

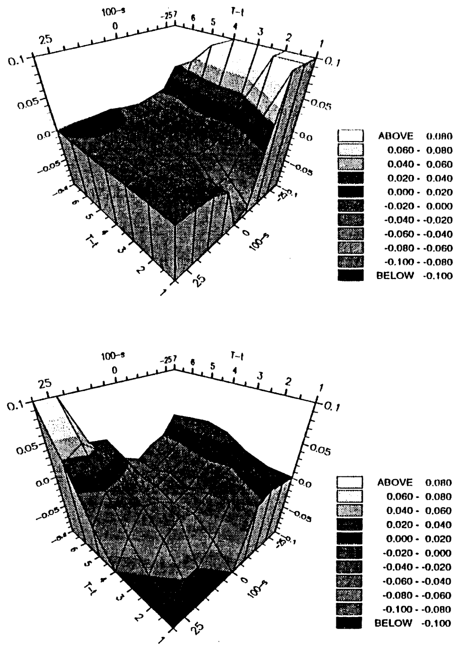


Figure 4.2: Graph for Call (top) and Put

The results described here will be used in the further analysis, notifying the fact, that in certain areas of the vol surface it will not be advisable to use the vol functions to price options, because the error is to big. This means that certain kind of options will be uninteresting to price with the vol func.

Chapter 5

Pricing with volatility function

5.1 Parameter estimation

To make the evaluation as realistic as possible, the parameters interest rate and dividend must be estimated empirically from information dated around the 24 May 1994 which is the date of the VM used to approximate the vol functions. These parameters will be assumed constant regardless of time or maturity of consideration, but will make the most realistic background for valuation of options on the 23/5-94 and with expiry 21/9-94, a 4 month maturity. Estimating the parameters around that date, will make it possible to benchmark the prices on options obtained with deterministic volatility, and the actual observed prices in the market.

Dividend :
This table contains the 13 dividends paid on the ftse100 in the time between 23/5-94 and 21/9-94. The 23/5-94 is used because the VM, from 24/5-94 is based on closing prices on the 23/5-94.

6/6	20/6	4/7	18/7	25/7	1/8	8/8	15/8	22/8	30/8	5/9	12/9	19/9
11.705	5.051	1.838	3.576	7.116	1.156	7.032	4.000	0.756	1.602	1.843	1.709	9.23

The estimate of an annualized average of the dividend is obtained by

$$\frac{\sum_{n=1}^{13} D_n}{TS} = D_0 = 0.05449711$$

where the ftse100 on 23 May 94 was S=3108.4. There are 122 days between 23/5 - 21/9 and 365 days in 1994 this gives T=$\frac{122}{365}$.

To incorporate a continuous dividend yield in BS ref [5] changes (3.10) into

$$\frac{\partial u}{\partial \tau} = \frac{1}{2}\sigma^2(S, T-\tau)S^2\frac{\partial^2 u}{\partial S^2} + (r - D_0)S\frac{\partial u}{\partial S} \tag{5.1}$$

The initial and boundary conditions remain the same, so that the Crank-Nicolson solver only changes slightly. The solution of BS with constant volatility changes to

$$V(S,t) = Se^{-D_0(T-t)}N(d_1) - Ee^{-r(T-t)}N(d_2)$$

where d_1 and d_2 change to

$$d_1 = \frac{\log(S/E) + (r - D_0 + \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

$$d_2 = \frac{\log(S/E) + (r - D_0 - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}$$

Interest rate :

For making the most realistic estimation of the interest rate, the period rate closest to the 4 month maturity will be used in an annualized form. The 6 month LIBOR at the 23 May 94 $R = 5.3125\%$, annualizing this interest rate by $e^{rT} = 1 + RT$, which gives

$$r = 0.0526588$$

In the further analysis a small computing time will be less important, therefore will a more accurate version of the Crank-Nicolson solver be appropriate. From subsection (3.2.4) the table shows that using $S_{max} = 10 * S_0$, J=500 (spacesteps), N=2000 (timesteps) gives a good accuracy.

5.2 Option pricing

5.2.1 Benchmarking with observed values

The information provided from BZW to carry out the benchmarking with observed prices, is 9 Call option prices on the ftse100, observed on the closing of the market 23/5-94, all with maturity 21/9-94. The results obtained are given in table 5.1 below. The BS values use the volatilities obtained with the coefficients given in table 3.1. The first three strikes are out-the-money so to obtain these $V(\sigma(S,t))$ the coefficients obtained with the the Put options have been used. The last six strikes are in-the-money, hence the coefficients obtained with the Call options have been used. The ftse100 on the 23/5-94 have been used as S_0

The results show that the observed values are different from the ones obtained with BS, this should not be the case since the implied volatilities from that day have been used to calculate the BS values. The reason for this bias is most likely due to lack of information in how to obtain the implied volatilities in the VM, so that this procedure could be used backward to obtain the observed values. This lack of information could be the procedure on how to interpolate the implied vol into the x- and y-labels in the VM. This interpolation is essentially carried out when making a surface fit to the 63 data points in the VM. If the interpolation used in this project deviates remarkable from the one used to actually obtain the VM, errors will be inevitable.

The conclusion on this matter is that the remaining part of the pricing will be carried out on the basis of comparison between BS values with constant volatilities from the vol func using coefficients in table 3.1, and values obtained with the vol functions. This does not mean that the vol function loose its generality, but only that the values used for benchmarking will be different. The graph 5.1 shows $\frac{V_{BS}-V(\sigma(S,t))}{V_{BS}}$ where the option evaluated has E=94.09. At S=100 on the graph, the relation between E an S correspond to the option with E=2925 on the 23/5-94 where the ftse100 was 3108.4, meaning the option is 5.9% in-the-money. The V_{BS} is solved with a constant volatility from the VM at T=4 years and S=100, obtained with the vol function using coefficients from table 3.1. The $V(\sigma(S,t))$ is computed with the vol func derived with Call options. This means that the surface shows the difference between constant- and deterministic volatility, where the constant volatility is chosen so that the relative difference is minimal at T=4 and S=100.

Strike	3325	3225	3125	3025	2925	2825	2725	2625	2525
% in-the-money	-6.96	-3.75	-0.53	2.68	5.90	9.11	12.33	15.55	18.77
Observed price	31	57.5	98.5	154	223	301.5	387.5	480.5	580.5
$V(\sigma(S,t))$	51.87	76.74	112.50	160.89	222.63	298.26	383.63	476.13	572.21
BS value	51.13	75.04	109.53	157.14	219.21	295.01	381.63	475.06	571.77

Table 5.1: results from benchmarking

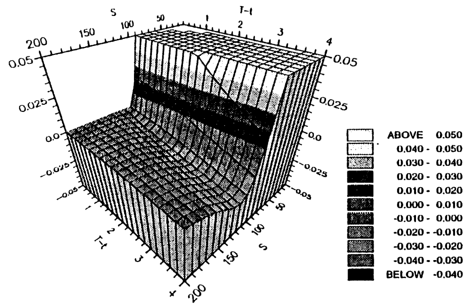


Figure 5.1: relative difference, E=94.09

5.2.2 Down-and-out call

The down-and-out option is a barrier option that ceases to exist if the underlying asset reaches a certain level, denoted by B. In this project the down-and-out Call option have been evaluated, in the two cases $B < E$ and $B > E$. The payoff of such

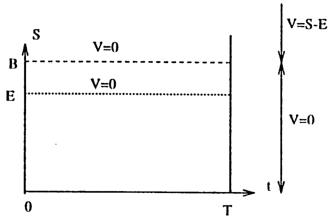


Figure 5.2: down-and-out barrier, $B > E$

an option for $B > E$ can be seen on figure 5.2.

case 1 : $B < E$

This option can be priced with the Crank-Nicolson solver by changing the boundary, and initial conditions,

bc : $S < B \quad U_j = 0 \qquad U_J = (S - Ee^{-r\tau})e^{r\tau}$

ic : $S \leq E \quad U_j^0 = 0 \qquad S > E \quad U_j^0 = S - E.$

When pricing with BS constant volatility, a small trick must be applied. The price of a down-and-out Call option is equal to the price of a Call subtracted the price of the correspondent down-and-in Call option. The price of an down-and-in option solved with BS is

$$Se^{-DT}(B/S)^{2\lambda}N(y) - Ee^{-rT}(B/S)^{2\lambda-2}N(y - \sigma\sqrt{T})$$

where

$$\lambda = \frac{r - D + \frac{\sigma^2}{2}}{\sigma^2}, \qquad y = \frac{ln[B^2/SE]}{\sigma\sqrt{T}} + \lambda\sigma\sqrt{T}.$$

In this case E=90, B=80, and the volatility for the analytic BS solution were obtained at t=12 months and $S_{rel} = 10$ in formulae (3.6). The results obtained in this case can be seen on the two graphs on next page.

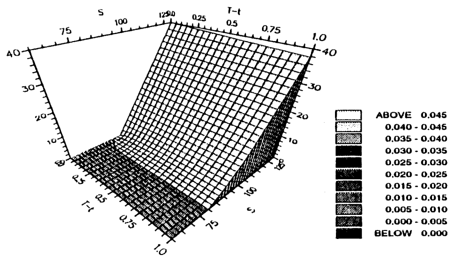


Figure 5.3: down-and-out Call E=90, B=80

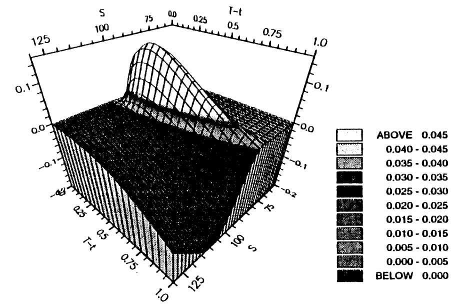


Figure 5.4: $V_{BS} - V(\sigma(S,t))$.

case 2 : $B > E$

The reason for solving the rather unusual case $B > S$ is, that the option is bounded to pay out a minimum of $B - E$ at expiry, hence the value of the option in the area $S > B$ will always be in-the-money. This means that when using the vol function it will only be applied where the error was small, see table 4.4.

In this case the boundary and initial conditions change to

bc : $S < B \quad U_j = 0 \qquad U_J = (S - Ee^{-r\tau})e^{r\tau}$

ic : $S \leq B \quad U_j^0 = 0 \qquad S > B \quad U_j^0 = S - E.$

Since the analytic solution assumes that $E > B$, it will not be applicable for this case. This means that both constant and deterministic vol will be solved with the

Crank-Nicolson solver The volatility chosen for solving BS, was for t=12 months and and $S_{rel} = 20$ in formulae (3.6). This results can be seen on the two graphs.

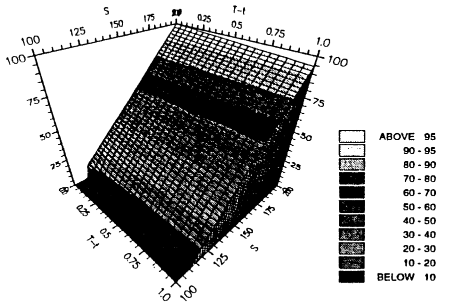


Figure 5.5: down-and-out-Call E=100, B=110

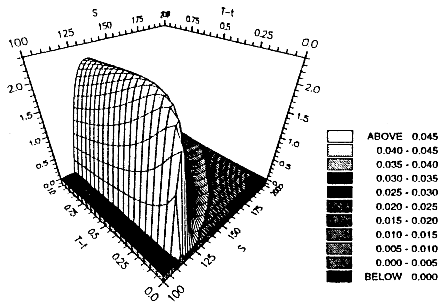


Figure 5.6: $V_{BS} - V(\sigma(S,t))$.

Chapter 6

Final remarks

The vol function has been biased due to several steps and assumptions in its derivation. When using the least squares method to fit the data points in the VM, the nature of this method has the consequence that error for such a large set of data points is inevitable. Therefore when using the initial vol surface to obtain values for evaluating BS with constant volatilities, and assumimg these are the observed values, two dimensions of error will be passed on in further analysis. The error in the surface fit, will lead to an optimal V_{min}, that is based on observed values that are not 100% correct. The second error is that the vol function takes a certain path between the discrete values in the VM. This path might not be similar to the path that would occur if information were available.
A suggestion for improvement would be to obtain the vol surfaces, using a method that would give smaller errors. Such a method might be cubic spline interpolation. This method would fit the data points exactly, to eliminate one of the errors.

When initially using the secant method to obtain a vol function, the very time consuming procedure, means that its lack of applicability first was discovered after a considerable amount of the project time was dedicated to it. The consequence of this was, that when a successful method was applied the time available for iterations to obtain an overall practicable vol surface was limited. This gave restrictions when pricing options so that certain contracts would be uninteresting to price due to that the area where such options are interesting to examine, would contain too large errors.

The results of the pricing with the vol function show that a quite large difference between constant and variable vol is obained. Some of the relations between the constant vol and vol function can be traced in the graphs, while some of the local behaviour in the graphs seems difficult to explain.

The time a vol function will be practicable to price options, depends on how much the VM changes over time. For a long period of time it may be reasonable to assume the VM to be constant, but for a shorter period a certain trend might be conspicuous, so that an even more consistent pricing method may be obtained if this is taken into consideration. These thoughts are beyond the scope of this project.

Chapter 7

Introduction to Bond pricing

7.1 Description of bond contracts

A bond is a contract that gives the owner a prescribed amount on a known date in the future, called the maturity date ($t = T$). When a bond changes hands the new owner pays for it when the contract is being transferred to him. A bond may pay out a dividend (called a coupon) at fixed times during the life of the contract. These bonds are known as coupon bonds. Bonds that pay no dividend are called zero-coupon bonds. The latter is easier to model and extensive literature may be found on this topic.
Evaluating a, say, zero-coupon bond that pays out DKr 1000 in 10 years from now, corresponds to asking the question ;

How much would one pay today to receive a guaranteed DKr 1000 in 10 year's time ?

The answer to this question depends on how much the amount will return from a bank deposit receiving an interest rate. If the bank interest rate implies that the amount accumulates to more than DKr 1000 in 10 years, one would rather deposit in the bank than invest in the bond.
A bond can therefore be described as an interest rate derivative product, so that the

price of a bond depends on time to maturity and the interest rate. The notation $P(t,T,r)$, for the price at time t, expiry at time T and interest rate r, will be used in the further analysis.

7.2 Derivation of the bond pricing formula

Supposing that the interest rate follows a one factor stochastic differential equation of the general form

$$dr = \mu(r,t)dt + \sigma(r,t)dw, \tag{7.1}$$

where w is a continous Brownian motion (ref [3]). The functions μ, σ are the instantaneous drift and standard deviation, of the process $r(t)$.

Now setting up a portfolio Π containing one bond with price P_1 and maturity T_1, and a number $-\Delta$ of a bond with price P_2 and maturity T_2, we get

$$\Pi = P_1 - \Delta P_2.$$

Now using Itô's lemma for P, given by

$$dP = \sigma\frac{\partial P}{\partial r}dw + \left(\mu\frac{\partial P}{\partial r} + \frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial r^2} + \frac{\partial P}{\partial t}\right)dt = \frac{\partial P}{\partial t}dt + \frac{\partial P}{\partial r}dr + \frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial r^2}dt \tag{7.2}$$

we can obtain an expression for the change in the portfolio in a time step dt by

$$d\Pi = \frac{\partial P_1}{\partial t}dt + \frac{\partial P_1}{\partial r}dr + \frac{1}{2}\sigma^2\frac{\partial^2 P_1}{\partial r^2}dt - \Delta\left(\frac{\partial P_2}{\partial t}dt + \frac{\partial P_2}{\partial r}dr + \frac{1}{2}\sigma^2\frac{\partial^2 P_2}{\partial r^2}dt\right),$$

where Δ is held constant during the time step dt. Now by eliminating the random component dw in $d\Pi$ we can obtain a deterministic expression where there is no risk. This will naturally create an optimal situation for investment purposes. So by choosing

$$\Delta = \frac{\partial P_1}{\partial r}\Big/\frac{\partial P_2}{\partial r},$$

so that

$$d\Pi = \left(\frac{\partial P_1}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P_1}{\partial r^2} - \frac{\partial P_1/\partial r}{\partial P_2/\partial r}\left(\frac{\partial P_2}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P_2}{\partial r^2}\right)\right)dt. \tag{7.3}$$

The change in the portfolio is now risk-free. This means that the return on the portfolio is equal to the risk-free interest rate, so that

$$\frac{\partial P}{\partial t} = rP,$$

in addition risk-free change means that

$$\frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial r^2} = 0.$$

This means that $d\Pi$ can be written as

$$d\Pi = r\left(P_1 - \frac{\partial P_1\big/\partial r}{\partial P_2\big/\partial r}P_2\right)dt = r\Pi dt. \tag{7.4}$$

Now using (7.3) and (7.4) and gathering all terms in P_1 on the left-hand side and all terms in P_2 on the right-hand side we find that

$$\left(\frac{\partial P_1}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P_1}{\partial r^2} - rP_1\right)\Big/\frac{\partial P_1}{\partial r} = \left(\frac{\partial P_2}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P_2}{\partial r^2} - rP_2\right)\Big/\frac{\partial P_2}{\partial r}.$$

This equation contains two unknowns P_1 and P_2. Since the left-hand side is a function of T_1 and the right-hand side a function of T_2, thus this equation only makes sense if there is independence of maturity dates. Hence the subscript on P can be dropped, and a shorter notation may be introduced

$$\left(\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial r^2} - rP\right)\Big/\frac{\partial P}{\partial r} = I(r,t). \tag{7.5}$$

Now specify the function $I(r,t)$ as

$$I(r,t) = \sigma\lambda - \mu, \tag{7.6}$$

The unknown function $\lambda(r,t)$ will be explained later in the derivation. Using (7.5) and (7.6) the zero-coupon bond equation is given by

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2\frac{\partial^2 P}{\partial r^2} + (\mu - \sigma\lambda)\frac{\partial P}{\partial r} - rP = 0. \tag{7.7}$$

To solve this equation boundary conditions and final conditions at $(t = T)$ must be applied. The boundary conditions depends on μ and σ, and they will therefore be dependent on the choice of the spot rate model (7.1). Final conditions corresponding to payout at maturity ($t = T$),

$$P(t, T, r) = M.$$

The unknown function λ given in (7.6) can be explained by isolating $\mu\frac{\partial P}{\partial r}$ in formula (7.7) and substitute this expression into Itô's lemma given in (7.2). From this derivation the following expression is obtained

$$dP - rPdt = \mu\frac{\partial P}{\partial r}(dw + \lambda dt) \tag{7.8}$$

This expression may be interpreted as follows; in return of accepting an extra risk dw the portfolio profits by an extra λdt. For this reason the function $lambda(r, t)$ is called the "market price of risk". It is quite common (ref [24], [13]) to write $\lambda(r, t)$ as

$$\lambda(r, t) = \frac{\mu - r}{\sigma} \tag{7.9}$$

where r is the current spot rate.

7.3 Objective

When pricing bonds the shapes of the functions σ and μ that enter into the interest rate model (7.1) must be determined. Various models can and has been suggested to model the nature of the interest rate. The next step in retrieving a valid model for bond pricing is to estimate the parameters that take part in this model. A natural way to estimate these parameters would be to use a data set of previous interest rates. This would lead to a model, giving the best possible representation of the interest rate.

When the interest rate model is used to price bonds, a model developed from a data set of previous interest rates would exhibit the biases that occur in the approximations used to derive the bond pricing equation. The consequence of this is that regardless of how good the model is to model interest rates it would lead to biases when pricing bonds.

On this background it would be more reasonable to model the interest rate implied by the bond pricing formulae. The task is therefore, to estimate the parameters in a given interest rate model, from a set of observed bond prices.

Chapter 8

Theoretical background

8.1 Interest rate model

The interest rate model used in this context, is the simple case where $\mu(r,t) = 0$ and $\sigma(r,t)$ is assumed to be constant, in the spot rate model (7.1), so that it can be written as

$$dr = \sigma dw. \tag{8.1}$$

This is a random walk and σ is called the volatility of the interest rate. The solution of (8.1) can easily be found to give

$$r(t) = r_0 + \sigma w(t), \quad r_0 = r(t_0), \quad w(t_0) = 0. \tag{8.2}$$

It follows that $w(t_0) = 0$ because w is a Wiener process (ref [12]). Now if we set the "market price of risk" $\lambda = 0$ in (7.7) and use the interest rate model (8.1), the zero-coupon bond equation changes into

$$\frac{\partial P}{\partial t} + \frac{1}{2}\sigma^2 \frac{\partial^2 P}{\partial r^2} - rP = 0. \tag{8.3}$$

8.1.1 Solution of the bond pricing equation

Now assuming that the solution of (8.3) is of the form

$$P(t,T,r(t)) = Me^{-(T-t)R(t,T)} \tag{8.4}$$

where $P(t,T,r(t))$ is the price of a bond at time t, maturity at time T and a known spot rate $r(t)$ at time t. M is the payout at maturity T, so that $P(T,T,r(t)) = M$. $R(t,T)$ is the term structure of interest rate. Substituting (8.4) into (8.3) leads to

$$R(t,T) + (T-t)\frac{dR(t,T)}{dt} + \frac{1}{2}\sigma^2 - r(t) = 0.$$

This is an ordinary differential equation and it is solved by

$$R(t,T)) = r(t) - \frac{1}{6}\sigma^2(T-t)^2.$$

Hence the price is given by

$$P(t,T,r(t)) = Me^{-(T-t)(r(t)-\frac{1}{6}\sigma^2(T-t)^2)}. \tag{8.5}$$

8.2 Stochastic representation of the bond price

The price of a zero-coupon bond can also be represented as the expected value (ref [6], [7], [24]). The price of a zero-coupon bond can generally be written on the form

$$\frac{dP}{P} = \mu(r,t)dt + \sigma(r,t)dw. \tag{8.6}$$

The market price of risk λ given by formula (7.9) can be substituted into (8.6) obtaining

$$\frac{dP}{P} = (r(t) + \lambda\sigma)dt + \sigma dw \tag{8.7}$$

The next step is to eliminate the market price of risk from occuring explicitly in the price process. This is obtained by transforming w into

$$\tilde{w}(t) = w(t) + \int_0^t \lambda(s)ds, \tag{8.8}$$

therefore substituting formula (8.8) into (8.7) leads to

$$\frac{dP}{P} = r(t)dt + \sigma d\tilde{w} \tag{8.9}$$

The transformation leads to a price process where the deterministic part of P only depends on r(t). By solving (8.9) the price of a zero-coupon bond can be found. Using the earlier final condition $P(T,T,r(t)) = M$, the solution is found to be

$$P(t,T,r(t)) = M\tilde{E}\{e^{-\int_t^T r(\tau)d\tau}|\mathcal{F}_t\}. \tag{8.10}$$

where $\tilde{E}$ is the expected value with respect to $\tilde{w}$ and $\mathcal{F}_t$ means conditioned on the information up to time t. The solution to (8.10) depends on the choice of interest rate model. Using a general one factor model given by

$$dr = (\eta(t)r + \theta(t))dt + \sigma(t)dw \tag{8.11}$$

where the functions $\eta(t), \theta(t), \sigma(t)$ are deterministic and bounded by $[0,\infty[$. In this general model it is not assumed that $r(t)$ is not equally bounded and this rather obvious requirement does not follow from the bounded functions due to the unbounded stochastic part, dw. The solution is given by

$$r(t) = e^{\int_{t_0}^t \eta(\psi)d\psi}\{r(t_0) + \int_{t_0}^t e^{-\int_{t_0}^s \eta(\psi)d\psi}\theta(s)ds + \int_{t_0}^t e^{-\int_{t_0}^s \eta(\psi)d\psi}\sigma(s)dw\}$$

The solution of (8.10) is given by

$$P(t,T,r(t)) = Mexp(-\tilde{E}\{\int_t^T r(\tau)d\tau|\mathcal{F}_t\} + \tilde{V}\{\int_t^T r(\tau)d\tau|F_t\}) \tag{8.12}$$

$$\tilde{E}\{\int_t^T r(\tau)d\tau|\mathcal{F}_t\} = r\int_t^T e^{\int_t^u \eta(\psi)d\psi}du + \int_t^T\int_t^u e^{\int_s^u \eta(\psi)d\psi}\theta(s)dsdu \tag{8.13}$$

$$\tilde{V}\{\int_t^T r(\tau)d\tau|\mathcal{F}_t\} = \int_t^T \left(\int_s^T (e^{\int_s^u \eta(\psi)d\psi}\sigma(s)du)^2 ds\right) \tag{8.14}$$

If we set $\eta(t) = 0$, $\theta(t) = 0$ and $\sigma(t) = \sigma$ the interest model given in (8.1) is obtained. Now substituting into (8.12) the zero-coupon bond price is obtained

$$P(t,T,r(t)) = Me^{-(T-t)(r(t)-\frac{1}{6}\sigma^2(T-t)^2)}. \tag{8.15}$$

It can now be seen that the bond price (8.5) that was found by solving (8.3) is identical to the bond price (8.15) found by solving (8.10) using (8.12).

8.3 Coupon bond pricing

A coupon bond, pays out a known amount of cash on certain dates in the life time of the bond. These bonds can easily be priced as a sum of zero-coupon bonds.

A coupon bond that pays out 3 coupons c, and M at expiry can be priced by

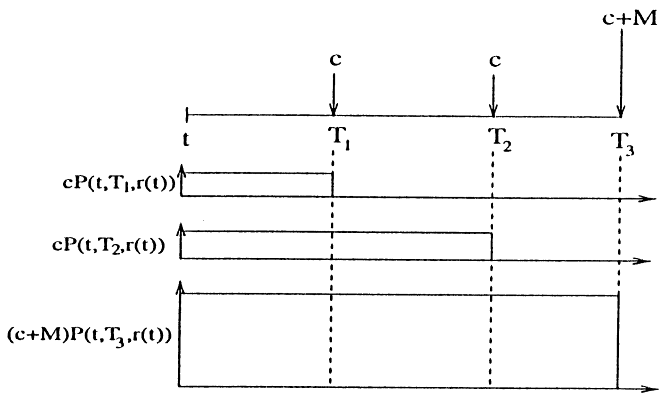


Figure 8.1: The cashflow and way of pricing a coupon bond

$$P(t,T_N,r(t)) = c\sum_{n=1}^{N} P(t,T_n,r(t)) + MP(t,T_N,r(t)).$$

Using (8.5) this can also be written as

$$P(t,T_N,r(t)) = c\sum_{n=1}^{N} e^{-(T_n-t)(r(t)-\frac{1}{6}\sigma^2(T_n-t)^2)} + Me^{-(T_N-t)(r(t)-\frac{1}{6}\sigma^2(T_N-t)^2)}, \quad (8.16)$$

where $N = 3$ or generally the number of coupons. The value of the coupon bond at time t (se figure 8.1) is therefore the value of the first coupon that is payed out at time T_1 plus the value of the second coupon payed out at time T_2 etc. The graph 8.2 is an example of a coupon bond. The jump in the price represents the payout of a coupon.

When estimating parameters the price of a coupon bond is a substantial extension from the zero-coupon bond (8.5). Most literature dealing with estimation of parameters only consider with zero-coupon bonds (ref [25], [24], [13]).

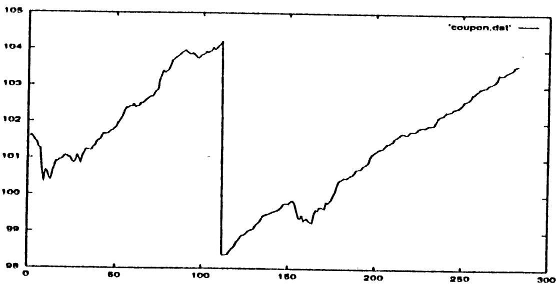


Figure 8.2: Prices from 2/8-94 to 8/9-95 for 6% Danske Stat St.Laan 1996

Chapter 9

Parameter estimation

9.1 The generalized approach

The problem can be stated as follows

Assuming that the interest rate follows a Brownian motion

$$dr = \sigma dw \tag{9.1}$$

that easily can be solved so that

$$r(t) = r_0 + \sigma w(t). \tag{9.2}$$

Now using formulae (9.1) an expression for the price of a coupon bond can be derived so that

$$P(t, T_N, r(t)) = c\sum_{n=1}^{N} e^{-(T_n-t)(r(t)-\frac{1}{6}\sigma^2(T_n-t)^2)} + Me^{-(T_N-t)(r(t)-\frac{1}{6}\sigma^2(T_N-t)^2)}. \tag{9.3}$$

With certain discrete equalized time steps, a set of different coupon bond prices are given

$$P_{ij}^{O}(t_i, T_{Nj}) \quad O \; for \; \text{"observed"} \tag{9.4}$$

The task is to estimate the parameters r_0 and σ in formula (9.2) such that the estimated prices given by formula (9.3) are as close as possible to the given values (9.4).

In general terms this problem can be formulated as

$$minimize\ e = \sum_i \sum_j \left(P^O_{ij}(t_i, T_{Nj}) - P(t_i, T_{Nj}, r(t))\right)^2$$

This representation of the problem is so general that a numerous of different methods can be suggested. In earlier approaches to this problem (ref [2]) the least squares method has been used, but this method does not necessarilly incorporate the fact that it is parameters in a stochastic differential equation that needs to be estimated. Furthermore it seems that various implementation of a (weighted, generalized, non-linear or ordinary) least squares method fail to take advantage of the fact that the volatility σ appears in the continuous time interest rate model (9.1) as well as in the discrete time bondpricing model (9.3).
A more intelligent approach would be to seek some sort of likelihood function for the bond price (ref [1]). This leads us to investigate the use of existing tools developed to solve similar problems in the technical sciences.

9.2 The use of CTLSM

CTLSM is short for Continuous Time Linear Stochastic Modelling, and it is a software package that can be used to estimate parameters in stochastic differential equations based on discrete time observations. It must be emphasized that this section is an explanation of the adaption of ctlsm to this problem and not a general manual (ref [8] , [18], [9]). The software package can be used for both linear and nonlinear models. In this case nonlinear models will be implemented. The stochastic state space model considered by ctlsm for nonlinear models is given by

$$dX = f(X, U, \theta)dt + dw(t) \tag{9.5}$$

where f is the drift term, X is the state space vector describing the system at time t and U is the input vector. The state space of a process is the set of possible values of an individual X_n in discrete time or $X(t)$ in continuous time. The term $w(t)$ is the noise of the process and this term is stochastic with independent increments it is assumed to be a Wiener process. Equation (9.5) is the stochastic non-linear state space model in continuous time.

In this case the state space model is used to model the spot rate given by (8.1), so that $X = r$, $f = 0$. When using ctlsm it is assumed that the function $f(X,U)$ is non zero ($f(X,U) \neq 0$). The consequence of this is that (8.1) cannot be represented by ctlsm, so that the state space model used takes the form

$$dr = ardt + dw(t), \quad f(X,U) = ar \tag{9.6}$$

The constant a will be assigned a very small value so that the consequences of the undesirable term will be minimized. The order of disturbance that a causes in the estimation will be determined. Y is the measurement equation in discrete time, it has the general form for nonlinear estimation

$$Y(t) = h(X,U,t) + e(t), \tag{9.7}$$

In this case $Y(t)$ is the observed bond prices, and $h(X,U,t)$ is the coupon bond pricing formulae given by (8.16). $e(t)$ is the measurement error, that is assumed to be Gaussian distributed with mean zero and covariance σ_e^2. $w(t)$ and $e(t)$ are assumed to be mutually independent. To fit the parameters in equation (9.7) to this case we set $X = r(t)$ and U will be specified in the next section. Equation (9.7) for this purpose therefore takes the form

$$Y(t) = P(t,T,r(t),U) + e(t) \tag{9.8}$$

The vector $Y(t)$ is the observed prices on different coupon bonds at time t. The reason for using different coupon bonds is to use their correlation, to make the best possible estimate of the parameter in the underlying nonobservable interest rates.

9.2.1 Disturbance from parameter a

To estimate the magnitude of interruption that the constant a from (9.6) causes in the estimation of the bond price, a small piece of algebra is necessary. Using the one factor model (8.11) where we set $\eta(t) = a$, $\theta(t) = 0$ and $\sigma(t) = \sigma$ so that the model (9.6) is reproduced. Now if we implement this model into the two moments contained in the stochastic representation of the bond price given by (8.12), we can give an estimate of the deviation in the price, as a result of the presence of a. Now the first moment $\tilde{E}$ given by (8.13) expanded as a Taylor series to $O(a)$

$$\tilde{E}\{\int_t^T r(\tau)d\tau|F_t\} = r(t)\int_t^T e^{\int_t^u a d\psi}du = r(t)((T-t)+\frac{1}{2}(T-t)^2 a). \qquad (9.9)$$

The second moment given by (8.14) expanded as a Taylor series to $O(a)$

$$\tilde{V}\{\int_t^T r(\tau)d\tau|F_t\} = \int_t^T (\int_s^T e^{\int_s^u a d\psi}\sigma du)^2)ds = \int_t^T \left(\sigma((T-s)+\frac{1}{2}(T-s)^2 a)\right)^2 \qquad (9.10)$$

From the equations (9.9) and (9.10) it can be seen that if we let $a \to 0$ the effect on the price can be neglected.

9.2.2 Conditions on the coupon bonds

The parameter estimations is made from prices on Danish Government bonds (in danish "Danske Stat St.Laan"). These bonds pay out a coupon once a year. The size of the coupon is constant throughout the lifetime of the bond. At maturity the bond pays out a coupon and the amount M, se graph 8.1.

Certain conventions are combined with buying Danish government bonds. The day of settlement is 3 business days after the day of agreement, see graph 9.2. Business days are days where the banks are open. In the last 30 "bond days" before a coupon is payed out, a new owner will not receive the coupon. By "bond days" is meant a year that is divided into days in which interest rate is paid. This "year" is divided into 12 months each containing 30 days. So that the full "year" is 360 days. When the calendar year is transformed into "coupon days" months containing 31 days are

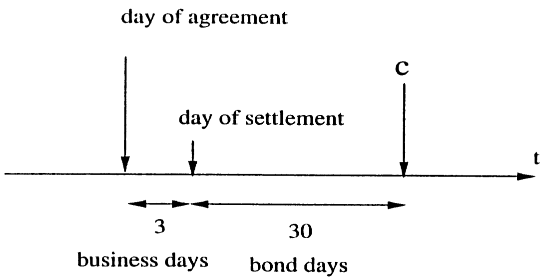


Figure 9.1: conventions on Danish government bonds

cut one day short and February is extended by 2 days. The 3 business days between agreement and settlement, is not relevant to this project, since it is the price at the day of payment that is being modelled.

Interest rates are as earlier described paid on a 360 days/year basis. The consequence of this is, that when the parameters in the interest rate model (9.6) are estimated, they will be retrieved from observed bond prices on business days. Business days are the days where the banks are open, and they are based on a calendar year with 365 days. To match up the dimensions of these two calendars, bond price observations on the last day in months containing 31 days should be dropped and in February the observation set would be two days short of observations.

It has been choosen to neglect these adjustments of the observations. This choice has been made for supporting a statistical as well as practical point of view not to rule out observations. It is also being reinforced by the fact that there are no observations on weekends.

9.2.3 Input parameters

When using ctlsm the set up is divided into three parts that each have their own input file. In the file "Defsys.f" (Appendix C) the state space model and the observation formulae are specified. In the file "Global.h" (Appendix D) the number of parameters

that take part in the model must be specified. In the file "readin" (Appendix E) the initial values of the parameters are specified, in this case r_0, σ^2 and σ_e^2.
The design of the parameters in (9.8) have to satisfy the conditions explained in the previous section. This means that when a bond change hands in the last 30 "bond days" before a coupon is payed out, the price drops the size of the coupon on the day before the last 30 "bond days". The incorporation of this has been handled by letting u_1 the first element in U in (9.8) be the time to the first coupon is payed out, see graph 9.2. u_2 is a trigger that is multiplied by the firstcoming coupon, equivalent to the first term in the sum in (8.16). When the last 30 "bond days" are reached the trigger turns off $u_2 = 0$.

When using ctlsm it is required that the observations are spread out with equal

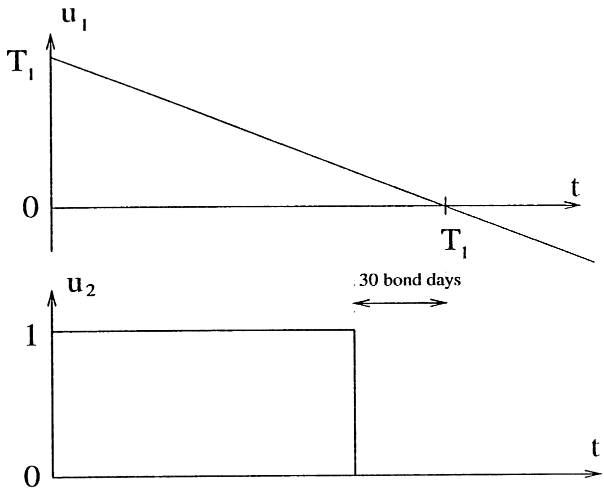


Figure 9.2: Inputs on the u vector

distances in time. In this case there are no observations on weekends and holidays, these will be treated as missing values. A missing value is predicted based on the current model and the earlier observations. This is carried out in a very simple manor by replacing values at time t with the values at time t-1.

In this case there are recorded one observation per day, stretched out over a period from 2/8-94 to 8/9-95. This gives 282 observations over the whole period. The missing values in weekends and holidays are predicted by ctlsm which results in 402 observations (see Appendix D).
The parameter a in (9.6) have been set to $a = 1d - 38$ in "Defsys.f" (Appendix C) throughout all the estimations. The typical initial values of r_0, σ^2 and $\hat{\sigma}_e^2$ are given in the last couple of lines in "readin" (Appendix E).

9.2.4 Presentation of results

Since it is the first time that ctlsm has been used for such a highly non linear observation formulae (9.8), it is advisable to compare estimations based on one bond price at a time. It can be inferred from the table that the $\hat{r}_0$ for the different bonds

	Estimated parameters		
Bond specification	$\hat{r}_0$	$\hat{\sigma}^2$	$\hat{\sigma}_e^2$
9% Danske Stat St.Laan 1996	0.684158e-1	0.230831e-6	-0.398046e-8
6.25% Danske Stat St.Laan 1997	0.692748e-1	0.216722e-6	-0.250063e-9
6% Danske Stat St.Laan 1996	0.664874e-1	0.250114e-6	-0.720442e-4
9% Danske Stat St.Laan 1995	0.656192e-1	0.331641e-6	-0.100815e-10
5.25% Danske Stat St.Laan 1996	0.674819e-1	0.208351e-6	-0.940145e-8

Table 9.1: Estimation of parameters from single dataseries

are quite equal. The estimates on $\hat{\sigma}$ deviates a little more, particularly "9% Danske Stat St.Laan 1995" diverge from the other values. This is most probably caused by the fact that this is the bond with the shortest time to maturity, and these bonds are traded in small volumes so that the observations that the results are based on not are very representative.
The estimates for $\hat{\sigma}_e^2$ are all negative. This should theoretically not be possible but it

only reflects the fact that the limits for which values $\hat{\sigma}_\epsilon^2$ can take have been specified as $-1d-7 < \hat{\sigma}_\epsilon^2 < 1d1$. The t-test that ctlsm makes for the different parameters shows that the probability of $\hat{\sigma}_\epsilon^2 = 0$ is $Pr(> |t|) = 0.997$ on average for the 5 different bonds. This is significantly proof for accepting the results and conclude that ctlsm is a well qualified program to solve this problem.

The next step in using ctlsm is to estimate the parameters over a cross section of bonds. This must be attained since the bonds are correlated by the fact that they are priced with the same formulae derived from one interest rate model.

A cross section of three bonds have been made, with the following results

$$\hat{r}_0 = 0.664876e-1 \qquad \hat{\sigma}^2 = 0.243401e-6$$

The values of $\hat{r}_0$ and $\hat{\sigma}^2$ are quite similar to the ones obtained in (9.1), but the dis-

Bond specification	$\hat{\sigma}_\epsilon^2$
9% Danske Stat St.Laan 1996	0.439369e0
9% Danske Stat St.Laan 1995	0.138900e-1
6% Danske Stat St.Laan 1996	-0.296877e-12

Table 9.2: Estimation of parameters from 3 dataseries

similarity in $\hat{\sigma}_\epsilon^2$ suggests that the interest model (9.1) is insufficient. This is reinforced by the relative large values of $\hat{\sigma}_\epsilon^2$ for the first two bonds, the last value is almost zero, but negative. The cause of the appearance of a negative value is similar to the cause explained under the estimation from single dataseries.

Estimating parameters using cross sections of bonds shows to be a slow process, since the estimations with ctlsm is quite time consuming. In addition ctlsm is very sensitive toward the initialization of parameters in "readin". A numerous amount of estimations breaks down before a right composition of initial values of parameters results in a successfull estimation.

Chapter 10

Final remarks

The results have shown that ctlsm is wellsuited for solving this problem. The simplicity of the interest rate model used has given the best possible foundation for retrieving results. It is, however a fact that the estimated parameters indicate that the model is insufficient, by being to simple. In this context it is worth mentioning that there are good prospects for using this software package to estimate parameters in more wellsuited interest rate models than the one used in this project. A suggestion might be models such as the "mean reverting process" given by

$$dr = c(\theta - r)dt + \sigma dw. \tag{10.1}$$

where c is the adjustment time coefficient, θ is the mean and σ^2 the variance of the Brownian motion.

The time consuming process of guessing initial values of parameters that leads to a succesfull estimation have meant that the quantity of results are quite sparse.

Considering the renowned CIR-models (ref [24], [2]) and others, where σ in (10.1) is a function $\sigma(r, t)$, an obvious improvement of ctlsm would be to let the state $\boldsymbol{X}$ in (9.5) enter into the stochastic term $\sigma(r, t)$.

A last remark must fall on the overall satisfactory achievement of estimating parameters in a continous stochastic differential equation based on observations in discrete time, modelled with a highly nonlinear model.

Appendix A

Least squares method

This program is a self updated version of the code given in "Numerical Recipes in C", specificaly the extension from one to two dimensions. The program is written in an Object Oriented way to give the most flexibility in definition of variables.

```
#include <stdio.h>
#include <math.h>
#define SWAP(a,b) {temp=(a);(a)=(b);(b)=temp;}

class Vector
{
public:
 int dimensions;
 float *v;
   Vector(int n);
};
Vector::Vector(int n)
   {
   dimensions=n;
   v=new float[n];
   for(int i=0;i<dimensions;i++) v[i]=0.;
}

class Matrix
{
public:
int dx,dy;
float **m;
Matrix(int x,int y);
```

```
};
Matrix::Matrix(int x,int y)
{
dx=x; dy=y;
m = new float *[dx];
for (int i=0;i<dx;i++)
m[i] = new float [dy];
for (int j=0;j<dx;j++)
for (i=0;i<dy;i++)
   m[i][j]=0.;
}

void gaussj(Matrix *a,Vector *b)

{
  int *indxc, *indxr, *ipiv, n;
  int i, icol, irow, j, k, l, ll;
  float big, dum, pivinv, temp;

  n = a->dx;
  indxc = new int[n];
  indxr = new int[n];
  ipiv  = new int[n];


  for (j=0;j<n;j++) ipiv[j]=0;
  for (i=0;i<n;i++)
{
    big=0.0;
for (j=0;j<n;j++)
    {
if (ipiv[j] != 1)
        {
for (k=0;k<n;k++)
{
if (ipiv[k] == 0)
{
if (fabs(a->m[j][k])>=big)
{
big=fabs(a->m[j][k]);
        irow=j;
        icol=k;
     }
   else if (ipiv[k] >1) printf("gaussj:singular Matrix-1");
}
```

```
}
     }
       }

     (ipiv[icol])++;

   if (irow != icol)
      {
for (l=0;l<n;l++)
SWAP(a->m[irow][l],a->m[icol][l])
SWAP(b->v[irow],b->v[icol])
      }

     indxr[i]=irow;
   indxc[i]=icol;
     if (a->m[icol][icol]==0.0) printf("gaussj: Singular Matrix-2");
     pivinv=1.0/a->m[icol][icol];
     a->m[icol][icol]=1.0;

   for (l=0;l<n;l++)
   a->m[icol][l]*=pivinv;
   b->v[icol] *= pivinv;


     for (ll=0;ll<n;ll++)
      {
if (ll != icol)
{
dum=a->m[ll][icol];
a->m[ll][icol]=0.0;
for (l=0;l<n;l++)
a->m[ll][l] -=a->m[icol][l]*dum;
b->v[ll]-=b->v[icol]*dum;
        }
      }
   }

for (l=n-1;l>=0;l--)
  {
if (indxr[l] != indxc[l])
      {
for (k=0;k<n;k++)
SWAP(a->m[k][indxr[l]],a->m[k][indxc[l]]);
        }
}
```

```
}

void basis_functions(Vector *afunc,float s, float t)
{
afunc->v[0]=1.;
afunc->v[1]=s;
afunc->v[2]=pow(s,2.);
afunc->v[3]=t;
afunc->v[4]=s/(t+2.);
afunc->v[5]=pow(s/(t+2.),2.);
}

void lfit(int ma,Vector *x,Vector *s,Vector *y,Vector *sig,
Vector *a)

{
int  i,j,k;
Vector afunc(ma);
Matrix alpha(ma,ma);

for (j=0;j<ma;j++)
{
for (k=0;k<ma;k++)
{
for (i=0;i< (x->dimensions) ;i++)
{
basis_functions(&afunc,x->v[i],s->v[i]);
alpha.m[j][k]+=(afunc.v[k]*afunc.v[j])/pow(sig->v[i],2.);
if (j==0) a->v[k] += y->v[i]*afunc.v[k]/pow(sig->v[i],2.);
          }
}
}
gaussj(&alpha,a);
}

void user_interface(int *ma,int *ndata)
{
printf("Surface fit for volatility data\n\n");
printf("Enter the number of points : ");
scanf("%d",ndata);
printf("Enter the number of coefficients to fit : ");
scanf("%d",ma);
}

void read_file(Vector *s,Vector *t,Vector *v,Vector *sig)
```

```
{
FILE *fptr;
fptr = fopen("a:tswvalu.txt","r");
        FILE *fptk;
fptk = fopen("a:a_new2.dat","r");
for (int j=0;j<s->dimensions;j++)
{
  // sig->v[j]=1.;
fscanf(fptr,"%f %f %f\n",&(t->v[j]),&(s->v[j]),&(sig->v[j]));
fscanf(fptk,"%f\n",&(v->v[j]));
    }
fclose(fptr);
fclose(fptk);
}

void results(int ma,Vector *s,Vector *t,Vector *v,Vector *a)
{
float sum,error;
Vector afunc(ma);

for (int j=0;j<ma;j++) printf("\na%d = %f",j,a->v[j]);
error=0.0;
for (int i=0;i<s->dimensions;i++)
{
basis_functions(&afunc,s->v[i],t->v[i]);
      sum=0.;
for (j=0;j<ma;j++)
sum += afunc.v[j]*a->v[j];
error+=fabs(sum-v->v[i]);
}
  printf("\n\n error = %f   average error = %f",error,error/s->dimensions);
}

void main(void)
{
int ma,ndata;
user_interface(&ma,&ndata);
Vector a(ma),s(ndata),t(ndata),v(ndata),sig(ndata);
read_file(&s,&t,&v,&sig);
lfit(ma,&s,&t,&v,&sig,&a);
results(ma,&s,&t,&v,&a);
}
```

Appendix B

BS with vol function

```
Valuation of Call option with volatility being a deterministic function
of time to maturity, and assetprice. A constant dividend yield is
incorporated
```

```
      IMPLICIT NONE
      DOUBLE PRECISION ds, T, E, Smax, r, dtau, D, vol(0:nx,0:ny),
   &  a0, a1, a2, a3, a4, a5, D, value(0:nx, 0:ny)
      INTEGER Jmax, Nmax

      PARAMETER nx=500, ny=2001

      COMMON /BLOCKA/ value
      COMMON /BLOCKB/ vol

      cccccccccccccccccccccccccccccccccccc
      SETTING VALUES
      T    = 0.33150684d0
      Nmax = 2000
      Jmax = 500
      Smax = 1000.0d0
      E    =  91.4505843d0
      r    = 0.052662d0
      D    = 0.0552908d0
      coefficient for vol func
      a0   = 17.00742d0
      a1   = -0.075064d0
      a2   = 0.001144d0
```

```
      a3    = 0.020357406d0
      a4    = -0.6934063d0
      a5    = 0.056017d0
c     ccccccccccccccccccccccccccccccccccc
c     step size determined
      dtau = T/Nmax
      ds    = Smax/Jmax
      CALL VOLATILITY(a0,a1,a2,a3,a4,a5,Nmax,Jmax,ds,dtau)
      CALL CRANK(Jmax, Nmax, E, r, D, ds, dtau)
      CALL RESULT
      STOP
      END

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE VOLATILITY(a0,a1,a2,a3,a4,a5,Nmax,Jmax,ds,dtau)
c     Reading in the volatility for each knots into vol(j,n)

       DOUBLE PRECISION vol(0:nx,0:ny), sr, tr, tau,
     & a0,a1,a2,a3,a4,a5,ds,dtau
       INTEGER n, j

       COMMON /BLOCKB/ vol
       do 100 n=0, Nmax+1, 1
         tau=n*dtau
         do 110 j=1, Jmax-1, 1
              s=j*ds
              ccccccccccccccccccccccccccccccccccccccccc
              VOLATILITY FUNCTION
              sr=100.0d0-s
              tr=tau*12.0d0
              if (sr .GT. 30.0d0) then
                sr=30.0d0
              endif
              if (sr .LT. -30.0d0) then
                sr=-30.0d0
              endif
              if (tr .LT. 1.0d0) then
                tr=1.0d0
              endif
              vol(j,n) =(a0 +a1*sr +a2*sr*sr +a3*tr +a4*sr/(tr+2.0d0)
     &        +a5*sr*sr/((tr+2.0d0)*(tr+2.0d0)) ) /100.0d0
              ccccccccccccccccccccccccccccccccccccccccc
 110     continue
 100   continue
```

```
      RETURN
      END

cccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

      SUBROUTINE CRANK(Jmax, Nmax, E, r, D, ds, dtau)
      DOUBLE PRECISION vol(0:nx,0:ny), Ej(0:500), Fj(0:500),
   &  U(0:500), Up(0:500), Oj,  alpha, beta, gamma,
   &  value(0:ny, 0:ny), s, tau, r, D, E, ds, dtau
      INTEGER n, j
      COMMON /BLOCKA/ value
      COMMON /BLOCKB/ vol

      ccccccccccccccccccccccccccccccccccccc
      Initial conditions for Thomas algorithm
      Ej(0)=0.0d0
      Fj(0)=0.0d0
      cccccccccccccccccccccccccccccccccccc
      Initial conditions (maturity)
      do  j=1, Jmax
          s=j*ds
          U(j)= s - E
          if (s .LT. E) then
          U(j)=0.0d0
          endif
      enddo
     cccccccccccccccccccccccccccccccccccccccccccccccccccccc
     Crank Nicolson solved with Thomas algorithm
     do 200 n=0, Nmax, 1
        tau=n*dtau
        do 210 j=1, Jmax-1, 1
             s=j*ds
             cccccccccccccccccccccccccccccccccccccccc
             boundary conditions at s=0 and s=Smax
             U(0)=0.0d0
             U(Jmax)=exp(-r*tau)*exp(r*tau)
             cccccccccccccccccccccccccccccccccccccccc
             coefficients alpha, beta, gamma for timestep n
             alpha =0.25d0*s*(dtau/ds)*(vol(j,n)*vol(j,n)*s/ds-(r-D))
             beta  =1.0d0-(dtau/(ds*ds))*(vol(j,n)*vol(j,n)*s*s)*0.5d0
             gamma =0.25d0*s*(dtau/ds)*(vol(j,n)*vol(j,n)*s/ds+(r-D))
             cccccccccccccccccccccccccccccccccccccccccccc
             Start of Thomas algorithm
             Oj    = alpha*U(j-1) + beta*U(j) + gamma*U(j+1)
             cccccccccccccccccccccccccccccccccccccccccccc
```

```
              alpha, beta, gamma for n+1 (note beta have a sign changing)
              alpha=0.25d0*s*(dtau/ds)*(vol(j,n+1)*vol(j,n+1)*
     &        s/ds-(r-D))
              beta  =1.0d0+(dtau/(ds*ds))
     &        *(vol(j,n+1)*vol(j,n+1)*s*s)*0.5d0
              gamma =0.25d0*s*(dtau/ds)*(vol(j,n+1)*vol(j,n+1)*
     &        s/ds+(r-D))
              ccccccccccccccccccccccccccccccccccccccccccc
              Ej(j) = gamma/(beta-alpha*Ej(j-1))
              Fj(j) = (Oj+alpha*Fj(j-1))/(beta-alpha*Ej(j-1))
 210      continue
          cccccccccccccccccccccccccccccccccccccccccccccccccc
          boundary conditions for Up at s=0 and s=Smax
          Up(0)       =  0.0d0
          Up(Jmax)    = exp(-r*(tau+dtau))*exp(r*(tau+dtau))
          cccccccccccccccccccccccccccccccccccccccccccccccccc
          Backward swep
          do 220  j=Jmax-1, 1, -1
              Up(j)  = Ej(j)*Up(j+1)+Fj(j)
              U(j)   = Up(j)
              value(j,n)=U(j)*exp(-r*tau)
 220      continue
 200    continue
        ccccccccccccccccccccccccccccccccccccc
        RETURN
        END

ccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccccc

        SUBROUTINE RESULT
        DOUBLE PRECISION  value(0:nx, 0:ny)
        INTEGER n, j
        COMMON / BLOCKA/ value
        open(7,file='EIG.DAT')
        write(7,500) ((value(j,n), j=2, 100, 2), n=0, 2000, 50)
        write(6,*) value(50,2000)
 500    format(1x,f15.7)
        close(7)
        RETURN
        END
```

Appendix C

Defsys.f

This is the print of the file Defsys.f that is a part of CTLSM.
In this file the state space model and observation equation is
specified.

```
c       @(#)Defsys.f    1.7     12/04/91
        subroutine defsys(x, nx, xm, id, nparam, xp0, xp, uo, t, a, b, c,
     $                    d, r1, r2, kf, n, m, s, kalm, naver, nn, f, h)
c-----------------------------------------------------------------------c
c       Specify the actual dynamic system.                              c
c                                                                       c
c       dx = a*x*dt + b*u*dt + dw  ;    E[dw*(dw)']  = r1               c
c                                                                       c
c       y  = c*x + d*u + e         ;    E[e*(e)']    = r2               c
c                                                                       c
c       x:  The parameter vector.                                       c
c       t:  The sampling time.                                          c
c       a:  The matrix a(n,n).                                          c
c       b:  The matrix b(n,m).                                          c
c       c:  The matrix c(s,n).                                          c
c       d:  The matrix d(s,m).                                          c
c       r1: The matrix r1(n,n).                                         c
c       r2: The matrix r2(s,s).                                         c
c       kf: The matrix kf(n,s).                                         c
c                                                                       c
c=======================================================================c
c       if nonlin=2 :                                                   c
c                                                                       c
c       dx = f(x,u) + dw                                                c
c       y  = h(x,u) + e                                                 c
c                                                                       c
c       then specify:  f(x,u), h(x,u),  a=df/dx, c=dh/dx                c
```

```
c     and skip:      b, d                                                c
c=====================================================================c
c                                                                     c
c     USAGE:                                                          c
c                                                                     c
c     Specify the system matrices in terms of x (variances            c
c     should be assigned the largest index of x).                     c
c     The samplingtime is Tsamp.                                      c
c---------------------------------------------------------------------c
      implicit none
c
      integer          i, j, nx, nparam, n, m, s, kalm, naver,
     $                 id(nparam), nn
c
      double precision x(nx), xm(nparam), t, Tsamp, xp0(n), xp(n),
     $                 uo(m), a(n, n), b(n, m), c(s, n), d(s, m),
     $                 r1(n, n), r2(s, s), kf(n, s), f(n), h(s)
c
c     .................................................................
c
      j = 1
      do 5 i = 1, nparam
         if (id(i) .ne. 0) then
            xm(i) = x(j)
            j = j + 1
         end if
 5    continue
      if (nn .eq. 1) goto 8
c
c     .................................................................
c                                                          Samplingtime
      Tsamp = 1d0
c
      t = Tsamp*naver
c
c     .................................................................
c                                                       initial states.
      xp0(1) = xm(1)
c     xp0(2) = xm(8)
c
c     .................................................................
c     Specify the actual dynamic system,
c     if (nonlin = 1) the matrices: a,b,c and d (not r1, r2 or kf) may
c     be dependant on the state vector: xp, and the input: uo(1)
c
```

```
  8   continue
c
      a(1, 1)  = 1d-38
c     a(1, 2)  = -xm(2)*xp(1)
c     a(2, 1)  = xm(4)*xp(2)
c     a(2, 2)  = xm(4)*xp(1) - xm(5) - 2d0*xm(6)*xp(2)
c
c bond 5035
      c(1, 1)  = -uo(1)*uo(3)*9d0*
     $          exp(-uo(1)*(xp(1)-1d0/6d0*xm(2)*uo(1)**2d0))-
     $       9d0*(uo(1)+1d0)*
     $    exp(-(uo(1)+1d0)*(xp(1)-1d0/6d0*xm(2)*(uo(1)+1d0)**2d0))-
     $       (9d0+100d0)*(uo(1)+2d0)*
     $    exp(-(uo(1)+2d0)*(xp(1)-1d0/6d0*xm(2)*(uo(1)+2d0)**2d0))
c
c bond 7916
      c(3, 1)  = -uo(2)*uo(4)*6d0*
     $          exp(-uo(2)*(xp(1)-1d0/6d0*xm(2)*uo(2)**2d0))-
     $       (6d0+100d0)*(uo(2)+1d0)*
     $    exp(-(uo(2)+1d0)*(xp(1)-1d0/6d0*xm(2)*(uo(2)+1d0)**2d0))
c
c     c(1, 2)  = 0d0
c     c(2, 1)  = 0d0
c     c(2, 2)  = 1d0
c
      f(1)     = 0d0
c     f(2)     = (xm(4)*xp(1)-xm(5)-xm(6)*xp(2))*xp(2)
c
c bond 5035
      h(1)     = uo(3)*9d0*exp(-uo(1)*(xp(1)-1d0/6d0*xm(2)*uo(1)**2))+
     $          9d0*
     $ exp(-(uo(1)+1d0)*(xp(1)-1d0/6d0*xm(2)*(uo(1)+1d0)**2d0))+
     $          (9d0+100.0d0)*
     $ exp(-(uo(1)+2d0)*(xp(1)-1d0/6d0*xm(2)*(uo(1)+2d0)**2d0))
c
c bond 7916
      h(3)     = uo(4)*6d0*exp(-uo(2)*(xp(1)-1d0/6d0*xm(2)*uo(2)**2))+
     $          (6d0+100d0)*
     $ exp(-(uo(2)+1d0)*(xp(1)-1d0/6d0*xm(2)*(uo(2)+1d0)**2d0))
c
c     h(2)     = xp(2)
c     ...................................................................
c                                            Variances (if kalm = 0).
      if (nn .eq. 1) goto 99
      if (kalm .ne. 1) then
```

```
         do 10 i = 1, n
         do 10 j = 1, n
 10         r1(i, j) = 0.0d0
c
         r1(1, 1) = xm(2)
c        r1(2, 2) = xm(10)
c
         do 20 i = 1, s
         do 20 j = 1, s
 20         r2(i, j) = 0.0d0
c
         r2(1, 1) = xm(3)
         r2(2, 2) = xm(4)
         r2(3, 3) = xm(5)
         r2(4, 4) = xm(6)
         r2(5, 5) = xm(7)
c     ...........................................................
      else
         do 30 i = 1, n
         do 30 j = 1, s
 30         kf(i, j) = xm(nparam-n*s+j+s*(i-1))
      end if
 99   return
      end
```

Appendix D

Global.h

This is the print of the file Global.h that is a part of CTLSM.
In this file the number of parameters is given.

```
c       @(#)Global.h    1.8     12/04/91
c----------------------------------------------------------------------c
c      Major global parameters.                                        c
c                                                                      c
c      maxobs: Maximum number of observations.                         c
c      nparam: Number of parameters in model, incl. noise terms.       c
c      n:      Order of the system of differential equations.          c
c      m:      Number of inputs.                                       c
c      s:      Number of outputs.                                      c
c----------------------------------------------------------------------c
      integer           maxobs, nparam, n, m, s
c
c----------------------------------------------------------------------c
      parameter         ( maxobs = 402 )
      parameter         ( nparam = 6 )
      parameter         ( n = 1 )
      parameter         ( m = 10 )
      parameter         ( s = 4 )
c----------------------------------------------------------------------c
```

Appendix E

readin

This is the print of the file readin that is a part of CTLSM. In this file the options for the estimation is specified. The initial guess of parameters and minimum and maximum values for each parameter are specified.

```
------Options-------     @(#)readin      1.2     12/17/92        ---------
'bondcros'  #01 filename for output files (max 8 characters).
0           #02 number of input records to be neglected.
0           #03 the output series will be moved the number of samples.
1           #04 the number of samples to be averaged.
1           #05 the k-step prediction used for estimation.
0           #06 order of interpolation between inputs (0/1).
0           #07 no action (0), mean removed (1), linear trend removed (2).
0           #08 estimate variances (0) or kalman gain directly (1).
2           #09 lin. timeinvar./lin. timevar./non-linear system (0/1/2).
1d1         #11 initial variance scaling.
0           #11 optimize (0), calc. hessian (1) ,output w/o optimize (2).
2000        #12 maximum number of function calls in optimization.
------Names for inputs and outputs-------------------------------------
-nr:--------name(max 13):----------------------------------------------
 1        'tau1          '
 2        'tau2          '
 3        'kupon1        '
 4        'kupon2        '
 5        'pris1         '
 6        'Pris2         '
------Parameters-------------------------------------------------------
-nr:-------name:------ini.value:-----min:----max:----(0/1/2):--std.dev.:-
 ------Initial states---------------------------------------------------
 1     'r-ini         '   9d-1          1d-4    2d1          1
```

```
------Variances or kalman gain------------------------------------------
 2    's11^2          '   7d-3         2d-10    1d1         1
 3    'se2^2          '   1d-2         -1d-7    1d1         1
------------------------------------------------------------------------
------Correlation matrix of prior density--(only for type=2)------------
```

Bibliography

[1] Braes Søren, Larsen Kent Stevens. 1989. Modellering af prisdannelse påobligationsmarkedet. *Lyngby, Eksamensprojekt, IMSOR.*

[2] Brown Roger H, Schaefer Stephen M. 1993. The term structure of real interest rates and the Cox, Ingersoll and Ross model. *Journal of Financial Economics 35 (1994) 3-42, North-Holland.*

[3] Cox D R, Miller H S. 1967. The theory of stochastic processes. *Chapman & Hall.*

[4] Dupire B. 1994. Pricing with a smile. *RISK, vol7, No1 January 1994.*

[5] Hull John C. 1993. Options, Futures and Other Derivative Securities. *Prentice hall International editions.*

[6] Jørgensen Peter Løchte. 1994. American Option Pricing. *Institut for Finansiering og kreditvæsen, Handelshøjskolen århus.*

[7] Kloeden Peter E, Platen Eckhard. 1980. Numerical Solution of stochastic Differential Equations. *Springer Verlag.*

[8] Madsen Henrik, Melgaard Henrik. 1993. CTLSM Continuous Time Linear Stochastic Modelling. *Technical report, No 1/1993, IMSOR.*

[9] Madsen Henrik, Melgaard H. 1991. The mathematical and numerical methods used in CTLSM. *Lyngby, IMSOR.*

[10] Madsen Henrik. 1989. Tidsrækkeanalyse. *Arbejdsudgave, IMSOR.*

[11] McKeague Ian W, Zhang Mei-Jie. Identification of nonlinear time series from first order cumulative characteristics. *The Annaks of Statistics 1994, vol 22, No. 1, 495-514.*

[12] Melgaard Henrik. 1994. Identification of physical models. *Lyngby 1994, IMM PHD.*

[13] Merton Robert C. 1990. Continuous-Time Finance. *Oxford, Blackwell.*

[14] Miltersen Kristian R, Sandmann Klaus, Sondermann Dieter. Closed form solutions for term structure derivatives with log-normal interest rates. *Discussion paper No. B-308*

[15] Morton K W. 1992. Lecture Notes on Finite Element Methods. *Oxford University Computing Laboratory.*

[16] Morton K W, Mayers D F. 1994. Numerical Solutions of Partial Differential Equations. *Cambridge, University Press.*

[17] NAG. 1988. The NAG Fortran Library Manual. *Mark 13, Volume 4.*

[18] Nielsen Jan Nygaard, Madsen Henrik. 1995. Estimation of embedded parameters in stochastic differential equations. *IMM, DTU, Denmark.*

[19] Pearson Neil D, Tong-Sheng Sun. 1994. Exploiting the Conditional Density in Estimating the Term Structure: An Application to the Cox, Ingersoll, and Ross Model. *The Journal of Finance, vol XLIX, No 4, September 1994.*

[20] Press W H, Teukolsky S A, Vetterling W T, Flannery B P. 1992. Numerical Recipes in C. *Cambridge, University Press.*

[21] Rubinstein M. 1994. Assesing new findings on implied binomial and trinomial tree structures. *University of California at Berkeley.*

[22] Sandmann Klaus, Sondermann Dieter. A term structure model and the pricing of interest rate derivative. *The review of futures markets, vol.12/2, 1993, pp.391-423.*

[23] Sløk Torsten. 1995. Intervention, valutakurs og renteudvikling under valutauro. *Nationaløkonomisk Tidsskrift 133 (1995) 178-188.*

[24] Vasicek Oldrich. 1977. An equilibrium characterization of the term structure. *Journal of Finacial Economics 5 (1977) 177-188 , North-Holland Publishing Company.*

[25] Wilmott P, Dewynne J, Howison S. 1993. Option Pricing. *Oxford Financial Press.*

Glossary

bs	boundary conditions
BS	Black Scholes
BZW	Barclay de Zoete Wedd Securities Limited
DAX	Deutche Aktie Index
E	Strike
ftsc	Financial Times Stock Exchange index
hedge	Eliminating risk
ic	initial conditions
LIBOR	London Interbank Offer Rate
pde	partial differential equation
portfolio	securities for investment purposes
S	assetprice
σ	Underlying asset volatility
vol	Underlying asset volatility
volatlity	standard deviation in the random walk
VM	volatility Matrix

Eksamensprojekter forår 1995 ved IMM.

1.	Allan Frese Rasmussen	Registrering af baneinventar ved hjælp af videofilm.
2.	Christian Henriksen	Modelling and control of underwater towed vehicle.
3.	Thomas Bagge Olesen	Organisation and strategy.
4.	Jens Kanstrup Kristensen	Ruteplanlægning og set partitioning.
5.	Dennis Schafroth	Søgestrategier i enumerationsmetoder.
6.	Henrik Holm Jørgensen Peter Lindstrøm	Optimal varmeproduktion.
7.	Alex Jensen	Modellering af selvsimilær opførsel med MAP-processer.
8.	Steen Strandskov Andersen Jesper Søegaard Christiansen	Design af et visuelt numerisk udviklingsværktøj.
9.	Per Albrektsen	Multipunktsmetode for optimeringsproblemer.
10.	Berno Wibbels	Power network design.
11.	Helene Moth-Poulsen	Længde- og artsestimering af fisk med vision.
12.	Thomas Stryger Olsen	A formal approach to risk assessment.
13.	Jakob Birkedal Nielsen	Jernbanekøretøjers dynamik i kurver med konstant radius.
14.	Jon Irgens	Mandskabsplanlægning - et case fra sundhedssektoren.
15.	Karina Schramm-Nielsen	Kvalitetssikring af kemiske referencematerialer.
16.	Jiri Dvorak	Solving of differential equations in a transputer network.
17.	Viet Nguyen	Multimedier og matematik.

Eksamensprojekter efterår 1995 ved IMM.

18.	Jens H. Kruuse	Statistisk analyse af scoliose operationer.
19.	Johan Doré Hansen	Interpolation og segmentering af skysekvenser.
20.	Bent Foss Pedersen	Visionsystem til lokalisering og måling af revner i beton.
21.	Nils V. Pedersen	Image analysis of fabric marks in paper.

22.	Thomas Gerd Hansen Thomas Thorbjørnsen	Optimeringsrutiner til billedklassifikation.
23.	Anne Sofie Østerdal	Analyse af chloridprofiler i beton.
24.	Kamilla Thingvad	Kulstyring på et kraftvarmeværk - et beslutningsstøttesystem.
25.	Helle Frimer-Larsen	Statistisk analyse af fiskemaver.
26.	Niels Henrik Thage	Simulering af beslutningsnetværk.
27.	Allan Larsen	Automatisk vognløbsplanlægning i et bybusselskab.
28.	Bjørn Hove	Rutelægning med tidsrestriktioner - udbringning og afhentning.
29.	Allan Skovgaard Hansen	The generalized knapsack problem - metaheuristical approaches.
30.	Morten Kjølby	Stochastic models for realtime control in urban drainage.
31.	Søren Lund	Rent parallaktisk stereo.
32.	Halldór Pálsson	Iterative methods for the Stokes problem.
33.	Thomas Hyldgaard Hansen	Value restriction and nonlinear constraints.
34.	Finn Kroijer	Volatility modelling in the option and bond pricing framework.
35.	Marianne Jacobsen	Nær-infrarød spektroskopi på kød - multivariat kalibrering.