DEGREE PROJECT IN MATHEMATICS,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2017

Calibrating the Hull-White model using Adjoint Algorithmic Differentiation

SIMON CARMELID

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ENGINEERING SCIENCES

Calibrating the Hull-White model using Adjoint Algorithmic Differentiation

SIMON CARMELID

Degree Projects in Financial Mathematics (30 ECTS credits)
Degree Programme in Mathematics
KTH Royal Institute of Technology year 2017
Supervisor at KTH: Henrik Hult
Examiner at KTH: Henrik Hult

TRITA-MAT-E 2017:57
ISRN-KTH/MAT/E--17/57--SE

Royal Institute of Technology
School of Engineering Sciences
KTH SCI
SE-100 44 Stockholm, Sweden
URL: www.kth.se/sci

Abstract

This thesis includes a brief introduction to Adjoint Algorithmic Differentiation (AAD), accompanied by numerical examples, step-by-step explanations and runtime comparisons to a finite difference method. In order to show the applicability of AAD in a stochastic setting, it is also applied in the calculation of the arbitrage free price and partial derivatives of a European call option, where the underlying stock has Geometric Brownian motion dynamics.

Finally the Hull-White model is calibrated using a set of zero coupon bonds, and a set of swaptions. Using AAD, the partial derivatives of the model are found and used in a Newton-Raphson method in order to find the market's implied volatility. The end result is a Monte Carlo simulated short rate curve and its derivatives with respect to the calibration parameters, i.e. the zero coupon bond and swaption prices.

Sammanfattning

Denna uppsats innehåller en introduktion till Adjungerad Algoritmisk Differentiering (AAD), tillsammans med numeriska exempel, steg-för-steg beskrivningar samt körtidsjämförelser med en finit differensmetod. För att illustrera applicerbarheten av AAD i ett stokastiskt ramverk, tillämpas metoden i beräkningen av det arbitragefria priset och de partiella derivatorna av en europeisk köp-option, där den underliggande aktien har geometrisk Brownsk dynamik.

Slutligen kalibreras Hull-White-modellen genom ett antal nollkupongsobligationer och swap-optioner. Via AAD beräknas de partiella derivatorna för modellen som sedan används i Newton-Raphsons metod för att finna markandens implicita volatilitet. Slutresultatet är en Monte Carlo-simulerad räntekurva och dess derivator med avseende på kalibreringsparametrarna, dvs. nollkupongs- och swap-optionspriserna.

Acknowledgements

First of all, I'd like to acknowledge the helpful guidance and input from professor Henrik Hult. You course corrected me when needed and provided much needed input in times of trouble.
Secondly, my family for your continued support throughout my studies. Especially my fiancé Nina. This would not have been possible without your help and backing!

Contents

Chapter 1

Introduction

Risk management has always been of interest for banks and investors, and following the financial crisis of 2008, emphasis on risk management and reporting has increased with several new regulations introduced in the EU. Also, with the electrification of securities trading that began in the 1970's and led to most of Europe's exchanges being fully automated in the 1990's, high-frequency-trading emerged as a way of gaining an edge on the market (see (10)). Thus, with more and more complicated products, traded at higher frequency and an increased emphasis on risk management, there is a need to develop faster methods for getting sensitivities with respect to model parameters (see (1)). While several approaches have been proposed (see for example (9), (7)), this thesis will focus on Adjoint Algorithmic Differentiation (AAD for short). AAD is sometimes referred to as Automatic Differentiation, and the idea was first introduced as early as 1964 (see (15)). An excellent source on AAD is the book by Griewank & Walter (see (11)).

A model often encountered in financial mathematics is the Hull-White one factor model for short rates. The model has several appealing properties since it is possible to fit the model to an initial yield curve and have the volatility time dependent. However, existing studies on calibrating the Hull-White model with time-dependent parameters are rather scarce (see (13)). The objective of this thesis is to propose a method for calibrating the Hull-White model and to find its sensitivities with respect to the calibration parameters.

1.1 Stochastic Differential Equations

For most pricing problems, there exists an element of uncertainty about what the future value will be. The value of some financial contract is thus derived from some unpredictable process, which determines the value of the contract at maturity. Contracts on this form are called derivatives and in a sense, almost any financial contract can be viewed as a derivative of some process. A stock price is a derivative of the market valuation of the company, and a savings account is a derivative of the interest rate. Even lottery tickets can be viewed as derivatives, in which case the underlying process is the numbers drawn in the lottery. In all of these processes, the investor estimates the behavior of the underlying process

in order to determine if the listed price is fair or not. In an arbitrage-free market, the expected outcome according to the market, is represented in the current price of the asset. Since the underlying process is unknown it is often modeled by a stochastic process, with the general representation

$$dX(t, X(t)) = \alpha(t, X(t))dt + \sigma(t, X(t))dW(t). \tag{1.1}$$

The function $\alpha(t, X(t))$ represents a drift function (seasonal drift, mean reverting drift, constant drift, etc.), and $\sigma(t, X(t))$ represents the volatility at time t, with the underlying process at $X(t)$. $W(t)$ is a one-dimensional Wiener-process, responsible for the *randomness* in the process. The general representation (1.1) is an example of a stochastic differential equation (SDE). SDEs represent a central part of financial mathematics and modeling and hence, this thesis.

1.2 Monte Carlo simulations

Another central part of financial modeling is to today try and determine the value of an asset at a future time T. For deterministic processes this is simply a matter of calculating the result, but in financial markets there are seldom any deterministic assets. Instead, one *simulates* the value process of a given asset based on a set of assumptions, including drift and volatility. In order to simulate a stochastic process such as (1.1) on a computer, a Wiener process is generated such that the value of (1.1) can be computed. However, with only one Wiener process generated, the result is just some random outcome out of all possible outcomes of the simulation and does not say much about the distribution of possible values at time T. However, when simulating the process a large number of times and averaging the result, the mean will tend toward the expected value of the process. Each simulated process, $\mathbf{X}_j$ is often referred to as a *path*, and the expected value of $\mathbf{X}(T)$ after a large number of simulations N is then

$$\mathbb{E}[\mathbf{X}(T)] \approx \frac{1}{N}\sum_{j=1}^{N}\mathbf{X}_j(T). \tag{1.2}$$

The result from generating many random samples will converge towards the true probability distribution of the process. However, the convergence can sometimes be slow, and a way of speeding it up is to use antithetic sampling (see (9)). In Figure 1.1, 2000 standard normal variables are plotted, as well as the first 1000 of those with their antithetic (or symmetric) counterparts. The antithetic sample converges faster towards the standard normal density function, and will be used in all Monte Carlo simulations in this thesis.

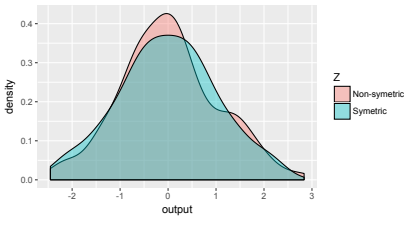

Figure 1.1: Sample of 2000 $iid \sim N(0, 1)$, and the first 1000 together with their antithetic counterparts.

Chapter 2

Adjoint Algorithmic Differentiation

Typically when using a computer to compute derivatives of a function a finite difference method, also known as a *bump-and-run* technique, is used. These methods are based on the assumption that during a sufficiently small interval $x + \Delta x$, a function $F(x)$ is approximately linear. These methods are explained further in Section 2.2. When using algorithmic differentiation however, there is no need for the linear assumptions associated with finite difference approximations.

The benefits of using algorithmic differentiation is twofold. First of all the desired derivatives are computed at machine accuracy, and truncation errors are avoided (14). Second, the computation of first order gradients can be completed at a small constant multiple of the cost of computing the original function. According to Griewank ((12)) and Capriotti ((4),(5)), this multiple is bounded at around 4, however Neuman ((14)) claims it typically ranges between 3 and 30. Regardless, it is often a vast improvement over the finite difference methods, which scale linearly with the number of inputs n.

For complex models, with a large number of inputs N, there is a limiting factor in terms of computing power to use the "brute force" methods from the past, which is a reason as to why *algorithmic differentiation*, or AD for short, has gained interest in recent years. Before explaining further what AD is, let's examine the following function

$$f(x_1, x_2, x_3, x_4) = \left(\frac{x_1 x_2}{x_3} + e^{\sin(x_4)} - \sin(x_4)\cos(x_3) + \frac{x_1}{x_2} \right) (x_1 - x_2 x_3)^2 . \tag{2.1}$$

The partial derivatives of (2.1) can be derived analytically in this example, but doing so can in many real world situations be a very tedious task. Furthermore, the resulting expressions can be even more complex than the original model. As an example, consider

$$\begin{aligned} \frac{\partial f}{\partial x_3} = &\left(\frac{-x_1 x_2}{x_3^2} + \sin(x_4)\sin(x_3) \right) (x_1 - x_2 x_3)^2 + \\ &(-2x_2) \left(\frac{x_1 x_2}{x_3} + e^{\sin(x_4)} - \sin(x_4)\cos(x_3) + \frac{x_1}{x_2} \right) (x_1 - x_2 x_3) , \end{aligned} \tag{2.2}$$

which is a lot more complex than the original function (2.1). However, notice the similarities between the two expressions (2.1) and (2.2). With a bit of clever design, there are many terms which need only be calculated once, and then reused to speed up the calculations. The idea of reusing past calculations will be exemplified in Section 2.3.

2.1 Inner workings of the CPU

In order to fully appreciate the advantages and drawbacks of AAD, a brief introduction into how the computer actually calculates is in order. It is common knowledge that the computer works with ones and zeros, or rather high and low voltage signals, in order to process what the user asks from it. Most of these signals are handled within the central processing unit, CPU.

The central processing unit contains various parts, such as a clock, program counter, registers, cache memory and arithmetic logic unit (ALU). The various parts of the CPU communicate and send data to each other via a central bus. What the CPU does is determined by the codes in the currently loaded program which are interpreted by the control logic to instruct other parts of the CPU weather to load or store data from the central bus.

The clock in the CPU determines when an operation is performed. The clock is binary, and sends a high and low voltage signal one after the other at a frequency often between 2 and 4 GHz on modern CPUs. Each time the clock signal goes high, the CPU does something, but only one thing. For our purposes, the registers, cache and ALU are of most importance. The ALU is the part of the CPU that does the actual math. In order to do so, there are two registers connected to it. The registers are small 'memories' of 8-, 16-, 32- or 64-bits depending on the type of CPU. When data is loaded into the two registers, the ALU can output the sum, difference, product or quotient of the two numbers onto the bus the next time the clock signal goes high. Then some other instruction determines what happens to the output during the next clock signal.

Let's illustrate the process of adding and multiplying some numbers together by an example. Consider the following program

```
example{
    a=3
    b=4
    c=2
    print((a+b)*c)
}
```

Figure 2.1: Example of a simple program

Table 2.1 shows an example of how the small program in Figure 2.1 could be computed. By observing that loading data into the registers, and storing data from the bus to some memory location is just as *slow* as doing multiplication, some take-aways are as follows;

Clock tick	Operation	On Bus
1	Load 3 into RegA	0000 0011
2	Load 4 into RegB	0000 0100
3	Compute Sum	0000 0111
4	Load output into RegA	0000 0111
5	Load 2 into RegB	0000 0010
6	Compute Product	0000 1110
7	Output bus to XXX	0000 1110

Table 2.1: Example of operations performed in the CPU in order to process the code in Figure 2.1.

- The ALU only operates on two numbers at a time
- Storing or fetching data is just as time consuming as doing a calculation of two numbers

While the cost of computing derivatives using AAD, in theory, should only be about 2 times the cost of calculating the original function, the number is offset by the extra number of operations performed relating to storing and fetching data(see (11),(12),(14)).

2.2 Finite Difference Methods

In Edsberg's book ((8)) a number of methods are presented on how to numerically calculate derivatives of functions, or rather, approximations of derivatives. The numerical solutions are often obtained by discretizing the function and using a finite difference method (FDM). Examples of known FDMs are Euler's method and Runge-Kutta's method.

Finite difference methods are sometimes reffered to as bump-and-run techniques and are based on the idea that in a sufficiently small interval $[x_1, x_1 + \Delta x]$, the function is essentially linear, and the derivative can therefore be approximated with

Definition 2.1. Euler forward

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1 + \Delta x, x_2, x_3, x_4) - f(x_1, x_2, x_3, x_4)}{\Delta x} + \mathcal{O}(\Delta x),$$

Definition 2.2. Euler backward

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1, x_2, x_3, x_4) - f(x_1 - \Delta x, x_2, x_3, x_4)}{\Delta x} + \mathcal{O}(\Delta x),$$

Definition 2.3. Central difference method

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1 + \Delta x, x_2, x_3, x_4) - f(x_1 - \Delta x, x_2, x_3, x_4)}{2\Delta x} + \mathcal{O}(\Delta x^2),$$

and similarly for $x_2, x_3, \ldots, x_N$. For smaller perturbations Δx, the error becomes smaller, however this leads us into the topic of truncation errors. Truncation errors arise from approximating a value with high precision to fewer digits.

This comes from the fact that Δx needs to become infinitely small in order for the difference quotient to converge to the analytical result. However, since the computer only has a finite number of bits to represent the values, the truncation error will distort the calculated value from the analytical solution. With Algorithmic Differentiation however, there are no truncation errors since the derivatives are never approximated in the first place(see (11)).

2.3 Adjoint Algorithmic Differentiation

Adjoint Algorithmic Differentiation takes advantage of the fact that any operation in a computer is a series of simpler operations, that are combined into a large model. As mentioned earlier the Arithmetic Logic Unit (ALU) inside the computer's CPU takes two inputs at a time and performs some operation on those, creating a large set of temporary outcomes (or states) along the way. The main idea is that a vector of inputs, IN, are inserted into the function or *method* to produce a vector of outputs OUT. Depending on the complexity of the method, there are a number of intermediate steps $I_0, I_1, I_2, \ldots, I_S$ produced "under the hood", before the output is presented. The process of going from the vector IN to the vector OUT can be schematically described as

$$\underbrace{IN \rightarrow I_0 \rightarrow I_1 \rightarrow I_2 \rightarrow \ldots I_{s-1} \rightarrow I_s \rightarrow OUT}_{\text{METHOD}(IN)\rightarrow OUT}.$$

Suppose we are interested in some partial derivative

$$\frac{\partial OUT}{\partial IN_i},$$

then it can be expressed using the chain rule as

$$\frac{\partial OUT}{\partial IN_i} = \frac{\partial OUT}{\partial I_s}\frac{\partial I_s}{\partial I_{s-1}} \cdots \frac{\partial I_1}{\partial I_0}\frac{\partial I_0}{\partial IN_i}.$$

As an example, consider equation (2.1) which could be performed in the follow-

ing steps

$$
\begin{aligned}
1) &= \left(\frac{\overbrace{x_1 x_2}^{=v_1}}{x_3} + e^{\overbrace{\sin(x_4)}^{=v_2}} - \sin(x_4)\underbrace{\cos(x_3)}_{=v_3} + \underbrace{\frac{x_1}{x_2}}_{=v_4} \right) \left(x_1 - \underbrace{x_2 x_3}_{=v_5} \right)^2 \\
2) &= \left(\underbrace{\frac{v_1}{x_3}}_{=v_6} + \underbrace{e^{v_2}}_{=v_7} - \underbrace{v_2 v_3}_{=v_8} + v_4 \right) \left(\underbrace{x_1 - v_5}_{=v_9} \right)^2 \\
3) &= \left(\underbrace{v_6 + v_7}_{v_{10}} + \underbrace{(-v_8 + v_4)}_{v_{11}} \right) \underbrace{v_9^2}_{v_{12}} \\
4) &= \left(\underbrace{v_{10} + v_{11}}_{v_{13}} \right) v_{12} \\
5) &= \underbrace{v_{13} v_{12}}_{v_{14}} \\
6) &= v_{14} = f(x_1, x_2, x_3, x_4).
\end{aligned}
$$

These steps exemplify how equation (2.1) is actually a result of (in this case) 14 separate rather simple operations. AAD takes advantage of this fact with the help of the chain rule in order to produce the partial derivatives of f. Let's define the steps described above, as a series of intermediate vector valued functions $\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_s$ defined as

$$
\begin{aligned}
\mathbf{I}_0(\mathbf{x}) &= \begin{bmatrix} x_1 & x_3 & x_1 x_2 & \sin(x_4) & \cos(x_3) & \frac{x_1}{x_2} & x_2 x_3 \end{bmatrix} \\
&= \begin{bmatrix} x_1 & x_3 & v_1 & v_2 & v_3 & v_4 & v_5 \end{bmatrix} \\
\mathbf{I}_1(\mathbf{I}_0) &= \begin{bmatrix} v_4 & \frac{v_1}{x_3} & e^{v_2} & v_2 v_3 & x_1 - v_5 \end{bmatrix} \\
&= \begin{bmatrix} v_4 & v_6 & v_7 & v_8 & v_9 \end{bmatrix} \\
\mathbf{I}_2(\mathbf{I}_1) &= \begin{bmatrix} v_6 + v_7 & -v_8 + v_4 & v_9 v_9 \end{bmatrix} = \begin{bmatrix} v_{10} & v_{11} & v_{12} \end{bmatrix} \\
\mathbf{I}_3(\mathbf{I}_2) &= \begin{bmatrix} v_{10} + v_{11} & v_{12} \end{bmatrix} = \begin{bmatrix} v_{13} & v_{12} \end{bmatrix} \\
\mathbf{I}_4(\mathbf{I}_3) &= \begin{bmatrix} v_{13} v_{12} \end{bmatrix} = \begin{bmatrix} v_{14} \end{bmatrix},
\end{aligned}
\tag{2.3}
$$

giving

$$
f(\mathbf{x}) = \mathbf{I}_4(\mathbf{I}_3(\mathbf{I}_2(\mathbf{I}_1(\mathbf{I}_0(\mathbf{x}))))).
$$

Looking at the partial derivatives of f,

$$
\begin{aligned}
\frac{\partial f}{\partial x_1} &= \frac{\partial \mathbf{I}_4}{\partial \mathbf{I}_3} \frac{\partial \mathbf{I}_3}{\partial \mathbf{I}_2} \frac{\partial \mathbf{I}_2}{\partial \mathbf{I}_1} \frac{\partial \mathbf{I}_1}{\partial \mathbf{I}_0} \frac{\partial \mathbf{I}_0}{\partial x_1}, \\
\frac{\partial f}{\partial x_2} &= \frac{\partial \mathbf{I}_4}{\partial \mathbf{I}_3} \frac{\partial \mathbf{I}_3}{\partial \mathbf{I}_2} \frac{\partial \mathbf{I}_2}{\partial \mathbf{I}_1} \frac{\partial \mathbf{I}_1}{\partial \mathbf{I}_0} \frac{\partial \mathbf{I}_0}{\partial x_2}, \\
\frac{\partial f}{\partial x_3} &= \frac{\partial \mathbf{I}_4}{\partial \mathbf{I}_3} \frac{\partial \mathbf{I}_3}{\partial \mathbf{I}_2} \frac{\partial \mathbf{I}_2}{\partial \mathbf{I}_1} \frac{\partial \mathbf{I}_1}{\partial \mathbf{I}_0} \frac{\partial \mathbf{I}_0}{\partial x_3}, \\
\frac{\partial f}{\partial x_4} &= \frac{\partial \mathbf{I}_4}{\partial \mathbf{I}_3} \frac{\partial \mathbf{I}_3}{\partial \mathbf{I}_2} \frac{\partial \mathbf{I}_2}{\partial \mathbf{I}_1} \frac{\partial \mathbf{I}_1}{\partial \mathbf{I}_0} \frac{\partial \mathbf{I}_0}{\partial x_4},
\end{aligned}
$$

it can be noted that the right hand side of the equations look very similar. In fact, they can be written as

$$\frac{\partial f}{\partial x_i} = \left(\frac{\partial \mathbf{I}_4}{\partial \mathbf{I}_3} \frac{\partial \mathbf{I}_3}{\partial \mathbf{I}_2} \frac{\partial \mathbf{I}_2}{\partial \mathbf{I}_1} \frac{\partial \mathbf{I}_1}{\partial \mathbf{I}_0} \right) \frac{\partial \mathbf{I}_0}{\partial x_i},$$

where the part in parenthesis is equal for all i. This is the fundamental idea of AAD. Once the parenthesis is calculated, the CPU time required for each partial derivative is drastically reduced.

In accordance with previous literature, we will be using the notation $\bar{v}_i = \partial y / \partial v_i$ for the so called *adjoint* of v_i. The adjoint variables in the context of algorithmic differentiation are defined as

Definition 2.4. Adjoint vector
The adjoint of some intermediate vector $\mathbf{I}_i$ is

$$\bar{\mathbf{I}}_i(\mathbf{I}_i, \bar{\mathbf{I}}_{i+1}) = \sum_{j=i+1}^{s} \bar{\mathbf{I}}_j \cdot \frac{\partial \mathbf{I}_j}{\partial \mathbf{I}_i}, \qquad \bar{\mathbf{I}}_s \equiv (1, 1, 1, \ldots, 1)^T,$$

where s is the total number of intermediate steps.

So when constructing the adjoint vectors, one has to go recursively backwards through the program, starting at $\bar{\mathbf{I}}_{s-1}$

2.3.1 Example

Looking back at (2.1),

$$f(x_1, x_2, x_3, x_4) = \left(\frac{x_1 x_2}{x_3} + e^{\sin(x_4)} - \sin(x_4)\cos(x_3) + \frac{x_1}{x_2} \right) (x_1 - x_2 x_3)^2,$$

and using the intermediate steps defined in (2.3), the adjoint vectors become

$$\begin{aligned}
\bar{\mathbf{I}}_4 &\equiv 1 = \bar{v}_{14} \\
\bar{\mathbf{I}}_3 &= \begin{bmatrix} \bar{v}_{14}\frac{\partial v_{14}}{\partial v_{13}} & \bar{v}_{14}\frac{\partial v_{14}}{\partial v_{12}} \end{bmatrix} = \begin{bmatrix} v_{12} & v_{13} \end{bmatrix} = \begin{bmatrix} \bar{v}_{13} & \bar{v}_{12} \end{bmatrix} \\
\bar{\mathbf{I}}_2 &= \begin{bmatrix} \bar{v}_{13}\frac{\partial v_{13}}{\partial v_{10}} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_{10}} \\ \bar{v}_{13}\frac{\partial v_{13}}{\partial v_{11}} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_{11}} \\ \bar{v}_{13}\frac{\partial v_{13}}{\partial v_{12}} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_{12}} \end{bmatrix} = \begin{bmatrix} \bar{v}_{13} \\ \bar{v}_{13} \\ \bar{v}_{12} \end{bmatrix} = \begin{bmatrix} \bar{v}_{10} \\ \bar{v}_{11} \\ \bar{v}_{12} \end{bmatrix} \\
\bar{\mathbf{I}}_1 &= \begin{bmatrix} \bar{v}_{10}\frac{\partial v_{10}}{\partial v_4} + \bar{v}_{11}\frac{\partial v_{11}}{\partial v_4} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_4} \\ \bar{v}_{10}\frac{\partial v_{10}}{\partial v_6} + \bar{v}_{11}\frac{\partial v_{11}}{\partial v_6} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_6} \\ \bar{v}_{10}\frac{\partial v_{10}}{\partial v_7} + \bar{v}_{11}\frac{\partial v_{11}}{\partial v_7} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_7} \\ \bar{v}_{10}\frac{\partial v_{10}}{\partial v_8} + \bar{v}_{11}\frac{\partial v_{11}}{\partial v_8} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_8} \\ \bar{v}_{10}\frac{\partial v_{10}}{\partial v_9} + \bar{v}_{11}\frac{\partial v_{11}}{\partial v_9} + \bar{v}_{12}\frac{\partial v_{12}}{\partial v_9} \end{bmatrix} = \begin{bmatrix} \bar{v}_{11} \\ \bar{v}_{10} \\ \bar{v}_{10} \\ -\bar{v}_{11} \\ 2\bar{v}_{12}v_9 \end{bmatrix} = \begin{bmatrix} \bar{v}_4 \\ \bar{v}_6 \\ \bar{v}_7 \\ \bar{v}_8 \\ \bar{v}_9 \end{bmatrix}
\end{aligned}$$

$$
\begin{aligned}
\bar{\mathbf{I}}_0 &= \begin{bmatrix}
\bar{v}_4\frac{\partial v_4}{\partial x_1} + \bar{v}_6\frac{\partial v_6}{\partial x_1} + \bar{v}_7\frac{\partial v_7}{\partial x_1} + \bar{v}_8\frac{\partial v_8}{\partial x_1} + \bar{v}_9\frac{\partial v_9}{\partial x_1} \\
\bar{v}_4\frac{\partial v_4}{\partial x_3} + \bar{v}_6\frac{\partial v_6}{\partial x_3} + \bar{v}_7\frac{\partial v_7}{\partial x_3} + \bar{v}_8\frac{\partial v_8}{\partial x_3} + \bar{v}_9\frac{\partial v_9}{\partial x_3} \\
\bar{v}_4\frac{\partial v_4}{\partial v_1} + \bar{v}_6\frac{\partial v_6}{\partial v_1} + \bar{v}_7\frac{\partial v_7}{\partial v_1} + \bar{v}_8\frac{\partial v_8}{\partial v_1} + \bar{v}_9\frac{\partial v_9}{\partial v_1} \\
\bar{v}_4\frac{\partial v_4}{\partial v_2} + \bar{v}_6\frac{\partial v_6}{\partial v_2} + \bar{v}_7\frac{\partial v_7}{\partial v_2} + \bar{v}_8\frac{\partial v_8}{\partial v_2} + \bar{v}_9\frac{\partial v_9}{\partial v_2} \\
\bar{v}_4\frac{\partial v_4}{\partial v_3} + \bar{v}_6\frac{\partial v_6}{\partial v_3} + \bar{v}_7\frac{\partial v_7}{\partial v_3} + \bar{v}_8\frac{\partial v_8}{\partial v_3} + \bar{v}_9\frac{\partial v_9}{\partial v_3} \\
\bar{v}_4\frac{\partial v_4}{\partial v_4} + \bar{v}_6\frac{\partial v_6}{\partial v_4} + \bar{v}_7\frac{\partial v_7}{\partial v_4} + \bar{v}_8\frac{\partial v_8}{\partial v_4} + \bar{v}_9\frac{\partial v_9}{\partial v_4} \\
\bar{v}_4\frac{\partial v_4}{\partial v_5} + \bar{v}_6\frac{\partial v_6}{\partial v_5} + \bar{v}_7\frac{\partial v_7}{\partial v_5} + \bar{v}_8\frac{\partial v_8}{\partial v_5} + \bar{v}_9\frac{\partial v_9}{\partial v_5}
\end{bmatrix} = \\
&= \begin{bmatrix}
\bar{v}_9 \\
-\bar{v}_6\frac{v_1}{x_3^2} \\
\bar{v}_6\frac{1}{x_3} \\
\bar{v}_7 e^{v_2} + \bar{v}_8 v_3 \\
\bar{v}_8 v_2 \\
\bar{v}_4 \\
-\bar{v}_9
\end{bmatrix} = \begin{bmatrix}
\bar{x}_1 \\ \bar{x}_3 \\ \bar{v}_1 \\ \bar{v}_2 \\ \bar{v}_3 \\ \bar{v}_4 \\ \bar{v}_5
\end{bmatrix} \\
\bar{\mathbf{x}} &= \begin{bmatrix}
\bar{x}_1 + \bar{v}_1 x_2 + \bar{v}_4\frac{1}{x_2} \\
\bar{v}_1 x_1 - \bar{v}_4\frac{x_1}{x_2^2} + \bar{v}_5 x_3 \\
\bar{x}_3 - \bar{v}_3\sin(x_3) + \bar{v}_5 x_2 \\
\bar{v}_2\cos(x_4)
\end{bmatrix} = \begin{bmatrix}
\bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4
\end{bmatrix}.
\end{aligned}
\tag{2.4}
$$

The final adjoint vector $[\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4]$ is precisely

$$
\begin{pmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{x}_3 \\ \bar{x}_4 \end{pmatrix} = \begin{pmatrix} \frac{\partial f}{\partial x_1} \\ \frac{\partial f}{\partial x_2} \\ \frac{\partial f}{\partial x_3} \\ \frac{\partial f}{\partial x_4} \end{pmatrix},
$$

although expressed in a series of new variables v_i and $\bar{v}_i$. As a numeric example, consider the forward pass denoted "$\rightarrow$" and backward pass "$\leftarrow$" on page 13. So for some values of $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix}$, lets say $\begin{pmatrix} 1.5 & 2.5 & 1.0 & 0.5 \end{pmatrix}$, the result is

$$
\mathbf{x}\begin{cases} x_1 = 1.5000 \\ x_2 = 2.5000 \\ x_3 = 1.0000 \\ x_4 = 0.5000 \end{cases}
$$

$$
\rightarrow \quad \mathbf{I}_0\begin{cases} x_1 = 1.5000 \\ x_3 = 1.0000 \\ v_1 = 3.7500 \\ v_2 \approx 0.3894183 \\ v_3 \approx 0.5403023 \\ v_4 = 0.6000 \\ v_5 = 2.5000 \end{cases}
$$

$$
\rightarrow \quad \mathbf{I}_1\begin{cases} v_4 = 0.6000 \\ v_6 = 3.7500 \\ v_7 \approx 1.4761 \\ v_8 \approx 0.2104 \\ v_9 = -1.0000 \end{cases}
$$

$$
\rightarrow \quad \mathbf{I}_2\begin{cases} v_{10} \approx 5.2261 \\ v_{11} \approx 0.3896 \\ v_{12} = 1.0000 \end{cases}
$$

$$
\rightarrow \quad \mathbf{I}_3\begin{cases} v_{13} \approx 5.6157 \\ v_{12} = 1.0000 \end{cases}
$$

$$
\rightarrow \quad \mathbf{I}_4\left\{ v_{14} \approx 5.6157 \right.
$$

$$
\leftarrow \quad \bar{\mathbf{I}}_4\left\{ \bar{v}_{14} = 1.0000 \right.
$$

$$
\leftarrow \quad \bar{\mathbf{I}}_3\begin{cases} \bar{v}_{13} = 1.0000 \\ \bar{v}_{12} \approx 5.6157 \end{cases}
$$

$$
\leftarrow \quad \bar{\mathbf{I}}_2\begin{cases} \bar{v}_{10} = 1.0000 \\ \bar{v}_{11} = 1.0000 \\ \bar{v}_{12} \approx 5.6157 \end{cases}
$$

$$
\leftarrow \quad \bar{\mathbf{I}}_1\begin{cases} \bar{v}_4 = 1.0000 \\ \bar{v}_6 = 1.0000 \\ \bar{v}_7 = 1.0000 \\ \bar{v}_8 = -1.0000 \\ \bar{v}_9 \approx -11.2314 \end{cases}
$$

$$
\leftarrow \quad \bar{\mathbf{I}}_0\begin{cases} \bar{x}_1 \approx -11.2314 \\ \bar{x}_3 = -3.7500 \\ \bar{v}_1 = 1.0000 \\ \bar{v}_2 \approx 0.9358 \\ \bar{v}_3 \approx -0.3894 \\ \bar{v}_4 = 1.0000 \\ \bar{v}_5 \approx 11.2314 \end{cases}
$$

$$
\bar{\mathbf{x}}\begin{cases} \bar{x}_1 \approx -8.3314 \\ \bar{x}_2 \approx 12.4914 \\ \bar{x}_3 \approx 24.6563 \\ \bar{x}_4 \approx 0.8619 \end{cases}
$$

2.4 Chaining

To illustrate the power of using the chain rule in algorithmic differentiation, we introduce a concept called chaining. When using AAD, the task is to in each step along the way, find the relationship between input variables and output variables. Most times, the output of one method, is the input to the next method, which is the input in the method after that and so on. In order to find the relationship between the original input variables and the final results, there needs to be a way of linking (or chaining) these methods together. With AAD, it is simply a matter of multiplying them together.

Let's once again return to (2.1),

$$f(x_1, x_2, x_3, x_4) = \left(\frac{x_1 x_2}{x_3} + e^{\sin(x_4)} - \sin(x_4)\cos(x_3) + \frac{x_1}{x_2} \right) (x_1 - x_2 x_3)^2 .$$

With the notation introduced in (2.3), $f(\mathbf{x})$ is a mapping of $\mathbf{x} \to v_{14}$, and the partial derivatives $\partial v_{14}/\partial x_i$ are defined in (2.4).

What if v_{14} is the input to yet another function? Or what if x_1 is the output of some earlier procedure? We illustrate what happens by the following example

$$g(y) = \sin\left(e^{0.01y^3}\right). \tag{2.5}$$

By deconstructing (2.5) in the same manner as earlier, the following can be shown

$$\left.\begin{aligned} \mathbf{I}_0(y) &= 0.01y^3 = v_1 \\ \mathbf{I}_1(v_1) &= e^{v_1} = v_2 \\ \mathbf{I}_2(v_2) &= \sin(v_2) = v_3 \end{aligned}\right\} g(y) = \mathbf{I}_2(\mathbf{I}_1(\mathbf{I}_0(y))).$$

As usual, $\bar{v}_3 = 1$, and the full adjoint of g is,

$$\begin{aligned} \bar{v}_2 &= \bar{v}_3 \frac{dv_2}{dv_3} = \cos(v_2), \\ \bar{v}_1 &= \bar{v}_2 \frac{dv_1}{dv_2} = \bar{v}_2 e^{v_1} = \cos(v_2) v_2, \\ \bar{y} &= \bar{v_1} \frac{dv_1}{dy} = \bar{v_1}\left(0.03y^2\right) = 0.03y^2 v_2 \cos(v_2). \end{aligned} \tag{2.6}$$

Note that $\bar{y} = 0.03y^2 e^{0.01y^3}\cos(e^{0.01y^3})$ is equal to the analytical expression of dg/dy, which is expected. The adjoint expression is therefore not an approximation of the derivative of $g(x)$ at x, but rather the exact solution.
If $f(x_1, x_2, x_3, x_4) = v_{14} = y$ is the input to (2.5), the partial derivatives of

$$\frac{\partial g(f(x_1, x_2, x_3, x_4))}{\partial x_i},$$

can be calculated by means of *chaining*. That means the adjoints $\bar{x}_i$ calculated in (2.4), can simply be multiplied by the adjoint $\bar{y}$ to produce the desired partial derivatives since *(remember that $f(\mathbf{x}) = y$)*

$$\bar{x}_i \bar{y} = \frac{\partial f(\mathbf{x})}{\partial x_i} \underbrace{\frac{dy}{df(\mathbf{x})}}_{=1} \frac{dg(y)}{dy} = \frac{\partial g(y)}{\partial x_i}.$$

Using the same numerical example values as before $\begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} = \begin{pmatrix} 1.5 & 2.5 & 1.0 & 0.5 \end{pmatrix}$, the result is $v_{14} = y = 5.706112$ and

$$g(5.706112) = \sin\left(e^{0.01 \cdot 5.706112^3}\right) \approx \sin\left(e^{1.857893}\right)$$
$$\approx \sin(6.410218) \approx 0.1266917 = y.$$

The question now is how $g(y)$ is changed when some of the initial x_is are changed?
Since $v_2 = e^{0.01y^3}$, Equation (2.6) gives

$$\bar{y} = 0.03y^2 v_2 \cos(v_2) \approx 6.210992,$$

and multiplying $\bar{y}$ with the adjoints in (2.4) yields

$$\bar{x}_1\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_1} \approx -52.869346, \qquad \bar{x}_2\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_2} \approx 78.707071,$$
$$\bar{x}_3\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_3} \approx 156.417491, \qquad \bar{x}_4\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_4} \approx 5.858607.$$

When implementing

$$\frac{\partial g(f(\mathbf{x}))}{\partial x_1} \approx \frac{g(f(x_1 + h, x_2, x_3, x_4)) - g(f(x_1, x_2, x_3, x_4))}{h}, \qquad h = 0.0001 \tag{2.7}$$

the results are

$$\bar{x}_1\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_1} \approx -52.857276 \qquad \bar{x}_2\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_2} \approx 78.737702$$
$$\bar{x}_3\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_3} \approx 156.528886 \qquad \bar{x}_4\bar{y} = \frac{\partial g(f(\mathbf{x}))}{\partial x_4} \approx 5.858979$$

which are similar, but the runtime of doing the finite difference method is slower, and that is discussed more in Section 2.5.

Furthermore, suppose x_3 is the result of some earlier input, such that

$$x_3 = \gamma(a_1, a_2, a_3, a_4, a_5, a_6) = \frac{2a_1 - a_2a_3}{a_4a_5} + a_6.$$

With $a_1 = 3$, $a_2 = a_3 = \sqrt{2}$, $a_4 = 4$, $a_5 = 2$ and $a_6 = 0.5$, x_3 is still

$$\gamma(3, \sqrt{2}, \sqrt{2}, 4, 2, 0.5) = \frac{6 - 2}{8} + 0.5 = 1. \tag{2.8}$$

The adjoint variables are

$$\bar{a}_1 = \frac{2}{a_4a_5} = 0.25, \qquad \bar{a}_2 = \frac{-a_3}{a_4a_5} \approx -0.1767767,$$
$$\bar{a}_3 = \frac{-a_2}{a_4a_5} \approx -0.1767767, \qquad \bar{a}_4 = \frac{a_2a_3 - 2a_1}{a_4^2a_5} = -0.125,$$
$$\bar{a}_5 = \frac{a_2a_3 - 2a_1}{a_4a_5^2} = -0.25, \qquad \bar{a}_6 = 1.$$

It is now possible to calculate

$$\frac{\partial g(f(x_1, x_2, \gamma(a_1, a_2, a_3, a_4, a_5, a_6), x_4))}{\partial a_1},$$

by simply multiplying $\bar{x}_3\bar{y} \approx 156.417491$ with the corresponding adjoint. Numerical results at the point

$$\begin{bmatrix} x_1 \\ x_2 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ x_4 \end{bmatrix} = \begin{bmatrix} 1.5 \\ 2.5 \\ 3.0 \\ \sqrt{2} \\ \sqrt{2} \\ 4.0 \\ 2.0 \\ 0.5 \\ 0.5 \end{bmatrix} \quad \text{are} \quad \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \\ \bar{a}_1 \\ \bar{a}_2 \\ \bar{a}_3 \\ \bar{a}_4 \\ \bar{a}_5 \\ \bar{a}_6 \\ \bar{x}_4 \end{bmatrix} = \begin{bmatrix} -52.869346 \\ 78.707071 \\ 39.104373 \\ -27.650967 \\ -27.650967 \\ -19.552186 \\ -39.104373 \\ 156.417491 \\ 5.858607 \end{bmatrix}. \tag{2.9}$$

As expected $\partial g(y)/\partial x_3 = \partial g(y)/\partial a_6$ since $\bar{a}_6 = 1$.

Hopefully this clarifies what is meant by chaining in this context, and the power of chaining when looking for derivatives which are the result of multiple functions feeding each other.

2.5 Performance of AAD

By continuing with the functions presented in Section 2.4, specifically

$$g(y) = \sin\left(e^{0.01y^3}\right),$$

where

$$y = f(x_1, x_2, x_3, x_4) = \left(\frac{x_1 x_2}{x_3} + e^{\sin(x_4)} - \sin(x_4)\cos(x_3) + \frac{x_1}{x_2}\right)(x_1 - x_2 x_3)^2,$$

where in turn,

$$x_3 = \gamma(a_1, a_2, a_3, a_4, a_5, a_6) = \frac{2a_1 - a_2 a_3}{a_4 a_5} + a_6.$$

The partial derivatives

$$\frac{\partial g(y)}{\partial x_i}\ i \in [1, 2, 4] \qquad \text{and} \qquad \frac{\partial g(y)}{\partial a_i}\ i \in [1, 2, 3, 4, 5, 6],$$

were calculated both through the finite difference method corresponding to Euler forward, and by adjoint algorithmic differentiation. Both methods were relatively quick in solving these partial derivatives, so the computation was repeated 10 000 times for each method in order to get a good estimate. In order to get a standard deviation of the estimates, the entire computation of 10 000 repetitions was done 100 times, and the standard deviation as well as mean runtime of 10 000 repetitions could thus be computed. In table 2.2 the runtime

results of computing the partial derivatives using both techniques are given, along with the computation of only the function value.

Function	Runtime (s)	Std. Dev. (s)
$g(f(x_1, x_2, \gamma(a_1, a_2, a_3, a_4, a_5, a_6), x_4))$	0.14286	0.008735184
AAD derivatives	0.54379	0.02163158
FDM derivatives	1.45272	0.01780142

Table 2.2: Table shows standard deviation and mean runtime per cycle, over 100 cycles, where each cycle executes the function 10000 times. The runtime is therefore exaggerated by a factor of 10000, on a MacBook Pro 13 inch mid 2009.

Note that the FDM-method is not only slower, but also only an approximation, since the function $g(f(x_1, x_2, \gamma(a_1, a_2, a_3, a_4, a_5, a_6), x_4))$ is not linear. Greater accuracy can be achieved with smaller perturbation h, but that will lead to more truncation errors and is not the point of this demonstration. Rather, observe that the time required for the FDM-method is $1.45272/0.14286 \approx 10.2$ times slower than the time to compute only the value of the function. This is because there are 9 variables, giving 9 partial derivatives, and the $n+1$ runtime is thus expected. For the AAD method, the runtime is only $0.54379/0.14286 \approx 3.8$, which is within Griewank's proposed limits of 4-5x the original runtime (see (12)).

2.6 Notes on recursion

First of all, consider the function

$$y_{i+1} = \alpha x + \beta x y_i,$$

and the partial derivative of y_{i+1} with respect to x,

$$\bar{x} = \frac{\partial y_{i+1}}{\partial x} = \alpha + \beta y_i + \beta x \frac{\partial y_i}{\partial x} = \alpha + \beta y_i + \beta x \left(\alpha + \beta y_{i-1} + \beta x \frac{\partial y_{i-1}}{\partial x} \right) = \dots.$$

The structure of the above equation suggests that it needs to be solved recursively, however Griewank (see (11)) proposes a set of rules regarding AAD, and one of those (rule 10) states that there is only need for one forward sweep and one backward sweep. In other words, recursive functions can either be solved from "right-to-left", implying that i is on the further right along the grid than $i-1$, or from "left-to-right". This can be illustrated by the fact that

$$\frac{\partial y_i}{\partial x} = \frac{\partial y_i}{\partial y_{i-1}} \frac{\partial y_{i-1}}{\partial y_{i-2}} \cdots \frac{\partial y_1}{\partial y_0} \frac{\partial y_0}{\partial x} = \frac{\partial y_0}{\partial x} \frac{\partial y_1}{\partial y_0} \cdots \frac{\partial y_{i-1}}{\partial y_{i-2}} \frac{\partial y_i}{\partial y_{i-1}}.$$

Merely an aesthetic difference perhaps, but the implication suggests that for some applications, the adjoint variables can be constructed along with the original function in the forward pass. As an example, for some starting value y_0, the next value y_1 is

$$y_1 = \alpha x + \beta x y_0,$$

giving the adjoint variables as

$$\bar{x}_1 = \alpha + \beta y_0, \quad \bar{y}_0 = \beta x.$$

The adjoint variables are computed along with the value y_1, and carried along into the next propagation giving

$$y_2 = \alpha x + \beta x y_1,$$

with adjoint variable, (omitting $\bar{y}$)

$$\bar{x}_2 = \frac{\partial y_2}{\partial x} = (\alpha + \beta y_1) + \left(\beta x \underbrace{\frac{\partial y_1}{\partial x}}_{=\bar{x}_1} \right) = \alpha + \beta y_1 + \beta x \bar{x}_1.$$

These values are again carried along to create

$$\bar{x}_3 = \frac{\partial y_3}{\partial x} = (\alpha + \beta y_2) + \left(\beta x \underbrace{\frac{\partial y_2}{\partial x}}_{=\bar{x}_2} \right) = \alpha + \beta y_2 + \beta x \bar{x}_2,$$

and so on. The subscript in the adjoint variables suggests that $\bar{x}_k$ is the partial derivative $\partial y_k / \partial x$. When implementing this method in code it is of course up to the programmer if there is a need to save the intermediate $\bar{x}_k$s, or if only the final $\bar{x}_N$ is of interest, in which case the variable can be overwritten as

$$\bar{x} \leftarrow \alpha + \beta y_i + \beta x \bar{x},$$

where the arrow suggests assignment in code.

Chapter 3

AAD in the stochastic setting

In order to evaluate the feasibility of AAD in financial settings, we must consider the stochastic case as well, as financial mathematics is very much dependent on stochastic processes. Let's consider a european call-option on a geometric Brownian motion, for which we can analytically derive the derivatives, and hence see if the numerical results are in line with what is to be expected. The value of such a call-option, will be priced using Black-Scholes option pricing formula.

Definition 3.1. Black-Scholes formula
For a stock with dynamics as in Definition 3.2, the arbitrage free price at time t for a European call option with strike K and maturity $T > t$ is

$$c(S(t), t) = S(t)\phi(d_1) - e^{-r(T-t)}K\phi(d_2)$$

where

$$d_1 = \frac{1}{\sigma\sqrt{T-t}}\left(\ln\left(\frac{S(t)}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)(T-t)\right)$$
$$d_2 = d_1 - \sigma\sqrt{T-t}$$

and r is the risk free rate, σ the volatility of the underlying stock and $\phi(\cdot)$ is the standard normal cumulative distribution function.

For simplicity, consider the case when $r = t = 0$ and $T = 1$, in which case

$$C_0 = S_0\phi(\mu_1) - K\phi(\mu_2), \tag{3.1}$$

where

$$\mu_1 = \frac{\ln(S_0/K)}{\sigma} + \frac{\sigma}{2},$$
$$\mu_2 = \frac{\ln(S_0/K)}{\sigma} - \frac{\sigma}{2},$$

giving delta and vega as

$$\frac{\partial C_0}{\partial S_0} = \phi(d_1), \qquad \frac{\partial C_0}{\partial \sigma} = S_0\phi'(\mu_1).$$

3.1 Geometric Brownian Motion

In the usual Black-Scholes model, the underlying asset is assumed to have log-normal daily returns, that is, the underlying asset is a geometric Brownian motion GBM, defined as

$$S(t) = S_0 e^{(\mu - \frac{\sigma^2}{2})t + \sigma W(t)}. \tag{3.2}$$

Here is the formal definition:

Definition 3.2. Geometric Brownian Motion
A stochastic process S_t is said to follow a Geometric Brownian Motion, GBM, if it solves

$$dS(t) = \mu S(t)dt + \sigma S(t)dW(t), \quad S(0) = S_0,$$

where μ is the drift, σ is the volatility, and W is a Wiener process.

Using Ito calculus, the process in Definition 3.2 can be written as

$$S(t) = S_0 + \int_0^t \mu S(u)du + \int_0^t \sigma S(u)dW(u). \tag{3.3}$$

However, the Wiener process in Definition 3.2 (and subsequently Equation (3.3)) is Q-Wiener, and in order to compare the simulated results the local rate of return must equal the short rate. This can be done via a Girsanov transformation using the kernel

$$h = -\frac{\mu - r}{\sigma}.$$

With $r = 0$ as in (3.1), the dynamics of (3.3) become

$$dS(t) = S(t)\left((\mu + \sigma h)dt + \sigma dW^P(t)\right) = S(t)\sigma dW^P(t).$$

Furthermore, recall that

$$\int_t^{t+dt} dS(t) = S(t+dt) - S(t) \quad \Rightarrow \quad S(t+dt) = S(t) + \int_t^{t+dt} dS(t).$$

This is still considered in a continuous-time case, but if a Monte Carlo simulation is to be applied, the time axis is divided into $t \in [0, t_1, \ldots, t_n]$. The continuous time Wiener process is not available, however, for a sufficiently small interval dt, a first-order Euler approximation becomes

$$S(t+dt) \approx S(t) + \sigma S(t)\left(W^P(t+dt) - W^P(t)\right).$$

And from the fundamental properties of Wiener processes it is evident that

$$W(t) - W(s) \sim N(0, \sqrt{t-s}),$$

and therefore

$$S(t+dt) \approx S(t)\left(1 + \sigma N(0, \sqrt{dt})\right). \tag{3.4}$$

A simple propagation algorithm for (3.4) could then be

```
prop(i,S,dt,sigma){
S[i+1]=S[i]*(1+sigma*rnorm(1,mean=0,sd=sqrt(dt)))
}
```

and in the context of algorithmic differentiation, the propagation is a mapping of the state vector at time t_i to the state at t_{i+1} as

$$\begin{pmatrix} S_i \\ \sigma_i \\ Z_i \end{pmatrix} \underbrace{\rightarrow}_{\text{prop}} \begin{pmatrix} S_i(1+\sigma_i Z_i) \\ \sigma_i \\ N(0,\sqrt{dt}) \end{pmatrix} = \begin{pmatrix} S_{i+1} \\ \sigma_{i+1} \\ Z_{i+1} \end{pmatrix}, \tag{3.5}$$

such that Z is a vector of n $iid \sim N(0,\sqrt{dt})$. Finally, the value of a European call option at maturity T is

$$V(S_{n,k},K) = \begin{cases} S_{n,k}-K & \text{if } S_{n,k} > K \\ 0 & \text{otherwise} \end{cases} \tag{3.6}$$

for each Monte-Carlo simulated path k. The arbitrage free value at time $t = ydt$ is therefore

$$C(S_y, ydt) = \frac{e^{-r(ndt-ydt)}}{m} \sum_{k=1}^{m} V(S_{n,k},K),$$

where m is the number of simulated paths. The Monte-Carlo simulation further implies that

$$\lim_{m\to\infty} \frac{1}{m} \sum_{k=1}^{m} V(S_{n,k},K) = C_0$$

where C_0 is defined as in (3.1).

In order to calculate Delta and Vega using AAD, we need the adjoints

$$\bar{S}_0 = \frac{\partial V(S_n,K)}{\partial S_0}, \qquad \bar{\sigma}_0 = \frac{\partial V(S_n,K)}{\partial \sigma_0}.$$

However, $V(S_n,K)$ is not differentiable at $S_n = K$. In a practical sense, this does not matter, since the grid can be designed as to not include the point K. By setting $\bar{V} \equiv 1$, the first adjoint becomes

$$\bar{S}_n = \begin{cases} \bar{V}\frac{\partial V}{\partial S_n} = 1 & \text{if } S_n > K \\ undefined & \text{if } S_n = K \\ 0 & \text{if } S_n < K \end{cases}.$$

The rest of the adjoints are then computed by taking the adjoint of (3.5) such that

$$\begin{aligned} \bar{S}_i &= \bar{S}_{i+1}\frac{\partial S_{i+1}}{S_i} = \bar{S}_{i+1}(1+\sigma_i Z_i) \\ \bar{\sigma}_i &= \bar{S}_{i+1}\frac{\partial S_{i+1}}{\partial \sigma_i} + \bar{\sigma}_{i+1}\frac{\partial \sigma_{i+1}}{\partial \sigma_i} = \bar{S}_{i+1} S_i Z_i + \bar{\sigma}_{i+1} \end{aligned} \tag{3.7}$$

3.2 Numerical results

In order to test the feasibility of Adjoint Algorithmic Differentiation in a stochastic setting, the derivatives using AAD should coincide with the analytical values of Definition 3.1. Using Definition 3.1, with values $r = 0$, $t = 0$, $T = 1$, $\sigma = 0.25$, $S(0) = 100$ and $K = 102$, the arbitrage free price of the option becomes

$$\begin{aligned} c(S(0),0) &= 100\phi\left(\frac{\ln\left(\frac{100}{102}\right)}{0.25} + \frac{0.25}{2}\right) - 102\phi\left(\frac{\ln\left(\frac{100}{102}\right)}{0.25} - \frac{0.25}{2}\right) \\ &\approx 9,078459. \end{aligned}$$

Furthermore, the delta and vega of the call option are

$$\begin{aligned} \frac{\partial c(S(0),0)}{\partial S(0)} &= \phi(d_1) \approx 0.518261, \\ \frac{\partial c(S(0),0)}{\partial \sigma} &= S(0)\phi'(d_1)\sqrt{1-0} \approx 39.852427. \end{aligned}$$

The process defined in (3.4) was then simulated by discretizing the grid to 250 steps between $t = 0$ and T, such that $dt = 1/250$, and the value computed at maturity according to (3.6). In Table 3.1 the results are presented for different numbers of Monte Carlo paths, along with the deviation in percent from the analytical results.

MC paths	$c(S(0),0)$ (±%)	$\frac{\partial c}{\partial S}$ (±%)	$\frac{\partial c}{\partial \sigma}$ (±%)
10 000	8,939778 (-1,53%)	0,51545 (-0,54%)	39,57476 (-0,70%)
5 000	9,02042 (-0.64%)	0,52044 (+0.42%)	39,89564 (+0,11%)
1 000	8,76619 (-3,44%)	0,51810 (-0,03%)	38,64311 (-3,03 %)
500	9,14848 (+0,77%)	0,51580 (-0,47%)	40,36231 (+1,28%)
100	9,96143 (+9,73%)	0,54841 (+5,82%)	45,53559 (+14,26%)

Table 3.1: Monte Carlo simulated European call option prices along delta and vega for the option. The deviation from the analytical value in parenthesis.

What is not obvious from table 3.1 is that the results become less volatile with increased number of simulated Monte Carlo paths. The fact that 500 paths has similar accuracy as 10 000 paths is simply chance. This phenomenon is illustrated in figure 3.1 where 200 Monte Carlo simulations were performed, half with 1000 paths, the other half with 100 paths.

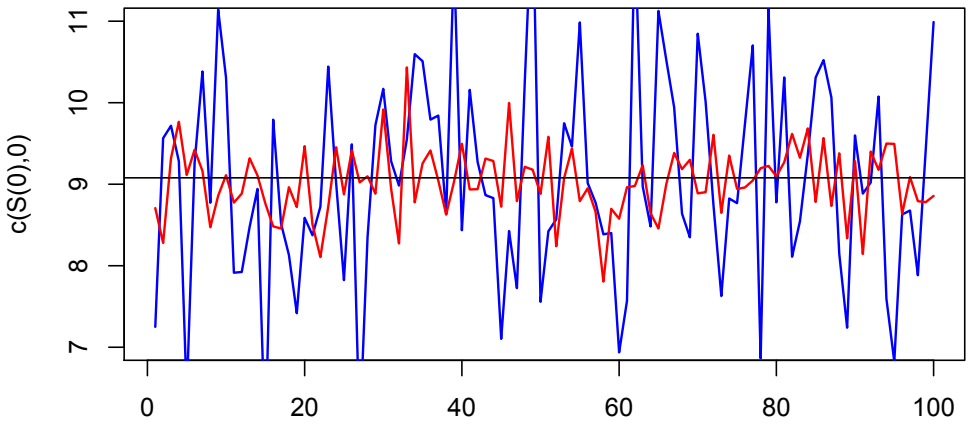

Figure 3.1: 100 simulations of call prices using 1000 Monte Carlo paths (red) and 100 MC paths (blue). Analytical price in black.

As expected, the results become less volatile and converge towards the analytical solution as the number of Monte Carlo paths simulated are increased. However, the main result is the derivatives, which are in line with the expected accuracy of a Monte Carlo simulation (see (9)), and thus show the feasibility of AAD in financial mathematics.

Chapter 4

The Hull-White model

A central part of financial mathematics is of course interest rates, which are used not only to discount future payments into todays equivalent value, but as underlying instrument for various assets. In order to simulate the future behavior of interest rates some model of their movements are needed. Since the future movements are unknown today, some kind of stochastic process could serve as a model.

4.1 Short Rate Models

There are several different stochastic representations of short rate dynamics. The general formula is

$$dr(t) = \mu(t, r(t))dt + \sigma(t, r(t))dW(t), \quad r(0) = r_0, \tag{4.1}$$

where $W(t)$ is a Wiener process. This general formulation has several variations with different properties. The classic Vasiček model describes the short rate through an Ornstein-Uhlenbeck process (see(9)) and is defined as

Definition 4.1. Vasiček model

$$dr(t) = \alpha(b - r(t))dt + \sigma dW(t),$$

where α, b, and σ are positive constants and W is a Wiener process.

This gives a so-called mean reverting process, since if $r(t) > b$, the drift will be negative, and if $r(t) < b$ the drift is positive. In other words, the process will tend towards the mean where $r(t) = b$.

The Cox-Ingersoll-Ross model expands upon the Vasiček model by having the variance proportional to the square root of the rate (see (6)). In other words, the greater the rate, the grater the variance.

Definition 4.2. Cox-Ingersoll-Ross

$$dr(t) = \alpha(b - r(t))dt + \sigma\sqrt{r(t)}dW, \quad \alpha > 0,$$

where W is a Wiener process.

The Cox-Ingersoll-Ross model, or CIR for short, is also a mean reverting process.

Both the Vasiček model and the CIR model have constant parameters a, b and σ. This can make it difficult for the models to precisely represent the yield curve given by a set of bonds. Therefore, there is also the Ho-Lee model

Definition 4.3. Ho-Lee

$$dr(t) = \theta(t)dt + \sigma dW(t),$$

where W is a Wiener process.

The Ho-Lee model can preform well when fitted against a yield curve of bond prices, however, since the volatility σ is constant, there are yet another model I would like to present, namely the Hull-White model.

The Hull-White model is a popular and commonly used model in financial modeling (see (13)). There are two main definitions of the model,

Definition 4.4. Hull-White (extended Vasiček)

$$dr(t) = (\Theta(t) - \alpha(t)r(t))dt + \sigma(t)dW(t), \quad \alpha(t) > 0,$$

where W is a Wiener process.

Definition 4.5. Hull-White (extended CIR)

$$dr(t) = (\Theta(t) - \alpha(t)r(t))dt + \sigma(t)\sqrt{r(t)}dW(t), \quad \alpha(t) > 0,$$

where W is a Wiener process.

They are similar to the Vasiček and Cox-Ingersoll-Ross models, but with time dependent variables instead of constants. In the rest of this thesis Definition 4.4 will be used.

4.2 Choosing $\theta(t)$

The θ in Definition 4.4 need be chosen such that the initial term structure of discount bonds is replicated by the model. First of all, assume that the instantaneous short rate under the risk adjusted measure Q is given by

$$r(t) = x(t) + \varphi(t), \qquad r(0) = r_0, \tag{4.2}$$

where

$$dx(t) = -ax(t)dt + \sigma(t)dW(t), \qquad x(0) = 0.$$

Which implies that $\varphi(0) = r(0) - x(0) = r_0$. Using a filtration $\mathcal{F}_s$ at s together with Proposition A.1, (4.2) can be written as

$$r(t) = x(s)e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\sigma(u)dW(u) + \varphi(t). \tag{4.3}$$

Meaning that $r(t)$ conditional on $\mathcal{F}_s$ is normally distributed with mean

$$\mathbb{E}\left[r(t)|\mathcal{F}_s\right] = x(s)e^{-a(t-s)} + \varphi(t), \tag{4.4}$$

and variance

$$\begin{aligned}
Var\left\{r(t)|\mathcal{F}_s\right\} &= \mathbb{E}\left[r(t)^2|\mathcal{F}_s\right] - \mathbb{E}\left[r(t)|\mathcal{F}_s\right]^2 \\
&= \mathbb{E}\left[\left(x(s)e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\sigma(u)dW(u) + \varphi(t)\right)^2 \middle| \mathcal{F}_s\right] \\
&\quad - \left(x(s)e^{-a(t-s)} + \varphi(t)\right)^2 \\
&= \mathbb{E}\left[\left(\int_s^t e^{-a(t-u)}\sigma(u)dW(u)\right)^2 \middle| \mathcal{F}_s\right] \\
&\quad + \mathbb{E}\left[2\left(x(s)e^{-a(t-s)} + \varphi(t)\right)\left(\int_s^t e^{-a(t-u)}\sigma(u)dW(u)\right) \middle| \mathcal{F}_s\right] \\
&\quad + \mathbb{E}\left[\left(x(s)e^{-a(t-s)} + \varphi(t)\right)^2 \middle| \mathcal{F}_s\right] \\
&\quad - \left(x(s)e^{-a(t-s)} + \varphi(t)\right)^2.
\end{aligned}$$

The second expectation is equal to zero, which follows from the fundamental properties of Wiener processes. The third expectation is deterministic, and thus the expectation can be removed, making it equal to the fourth term. Left behind is

$$\begin{aligned}
Var\left\{r(t)|\mathcal{F}_s\right\} &= \mathbb{E}\left[\left(\int_s^t e^{-a(t-u)}\sigma(u)dW(u)\right)^2 \middle| \mathcal{F}_s\right] \\
&= \int_s^t e^{-2a(t-u)}\sigma(u)^2 du.
\end{aligned} \tag{4.5}$$

The price $p(t,T)$ of a zero-coupon bond at time t, maturing at time T, and with unit face value is denoted

$$p(t,T) = \mathbb{E}\left[e^{-\int_t^T r(u)du} \middle| \mathcal{F}_t\right], \tag{4.6}$$

where $\mathbb{E}$ denotes the expectation under Q. Inserting (4.2) this can be written as

$$p(t,T) = \mathbb{E}\left[e^{-\int_t^T x(u)du - \int_t^T \varphi(u)du} \middle| \mathcal{F}_t\right].$$

With $\varphi(t)$ being a deterministic function, it can move outside the expectation, giving

$$p(t,T) = e^{-\int_t^T \varphi(u)du}\mathbb{E}\left[e^{-\int_t^T x(u)du} \middle| \mathcal{F}_t\right].$$

Using Lemma A.6 with $Y(t) = 0$, the exponent inside the expectation has the representation

$$I(t,T) = \int_t^T X(u)du = \int_t^T -x(u)du,$$

where $I(t,T)$ conditional on the sigma filed $\mathcal{F}_t$ is normally distributed with mean

$$M(t,T) = \frac{1-e^{-a(T-t)}}{a}X(t) = -\frac{1-e^{-a(T-t)}}{a}x(t),$$

and variance

$$V(t,T) = \int_t^T \frac{\sigma(u)^2}{a^2}\left(1 - e^{-a(T-u)}\right)^2 du.$$

Furthermore, with $I(t,T)$ being a normal random variable with the above mentioned mean and variance, then

$$\mathbb{E}\left[e^{I(t,T)}\middle|\mathcal{F}_t\right] = e^{M(t,T)+\frac{1}{2}V(t,T)}. \tag{4.7}$$

Therefore, equation (4.6) can be expressed as

$$p(t,T) = \exp\left\{-\int_t^T \varphi(u)du - \frac{1-e^{-a(T-t)}}{a}x(t) + \frac{1}{2}\int_t^T \frac{\sigma(u)^2}{a^2}\left(1-e^{-a(T-u)}\right)^2 du\right\}.$$

The objective is to find a function $\varphi(t)$ which accurately reproduces the observed term structure in the market. The prices of discount bonds can only be observed at time $t = 0$, for some maturity T_i, and we denote these observable prices by $p^\star(0,T_i)$. Therefore $\varphi(t)$ must solve the following equation

$$p^\star(0,T_i) = \exp\left\{-\int_0^{T_i}\varphi(u)du + \frac{1}{2}\int_0^{T_i}\frac{\sigma(u)^2}{a^2}\left(1-e^{-a(T-u)}\right)^2 du\right\},$$

since $x(0) = 0$. From the relationship between forward rates and bond prices

$$f(0,T) = -\frac{\partial \ln\left(p(0,T)\right)}{\partial T},$$

it follows that

$$f^\star(0,T_i) = \varphi(T) - \frac{\partial}{\partial T}\left\{\frac{1}{2}\int_0^{T_i}\frac{\sigma(u)^2}{a^2}\left(1-e^{-a(T-u)}\right)^2 du\right\}.$$

This results yields Proposition 4.6.

Proposition 4.6. ***Term structure fit***
The model defined in (4.2) *fits the currently observed term structure of discount factors if and only if, for each* T

$$\varphi(T) = f^\star(0,T) + \frac{1}{2}\frac{\partial}{\partial T}\left(\int_0^T \frac{\sigma(u)^2}{a^2}\left(1-e^{-a(T-u)}\right)^2 du\right).$$

However, this does not answer the question of how to choose $\theta(t)$ in Definition 4.4. The question can be answered by applying Itôs lemma to (4.2), namely

$$\begin{aligned} dr(t) &= \left(\frac{\partial\varphi(t)}{\partial t} - ax(t)\right)dt + \sigma(t)dW(t) \\ &= \left(\frac{\partial\varphi(t)}{\partial t} - a\left(r(t)-\varphi(t)\right)\right)dt + \sigma(t)dW(t) \\ &= \left(\left(\frac{\partial\varphi(t)}{\partial t} + a\varphi(t)\right) - ar(t)\right)dt + \sigma(t)dW(t). \end{aligned}$$

It follows that the model is fully specified by setting

$$\theta(t) = \frac{\partial \varphi(t)}{\partial t} + a\varphi(t). \tag{4.8}$$

Calculating the derivative of $\varphi(t)$ might seem daunting, however, calculating $\theta(t)$ explicitly is not necessary. By again using $x(t) = r(t) - \varphi(t)$, equations (4.3), (4.4) and (4.5) become

$$r(t) = \varphi(t) + r(s)e^{-a(t-s)} - \varphi(s)e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\sigma(u)dW(u), \tag{4.9}$$

$$\mathbb{E}\left[r(t)|\mathcal{F}_s\right] = \varphi(t) + r(s)e^{-a(t-s)} - \varphi(s)e^{-a(t-s)}, \tag{4.10}$$

$$Var\left\{r(t)|\mathcal{F}_s\right\} = \int_s^t e^{-2a(t-u)}\sigma(u)^2 du, \tag{4.11}$$

where $r(t)$ conditional on $\mathcal{F}_s$ is still normally distributed. Hence there is no need for an explicit expression for $\theta(t)$, and thus no need to find the partial derivative $\partial\varphi(t)/\partial t$. In fact, (4.9), (4.10) and (4.11) is all that is needed in order to propagate from $r(t_i)$ to $r(t_{i+1})$, but more on this in Chapter 5

4.3 Volatility-Calibration

In the previous section, we showed how to $\theta(T)$ is a determined by the forward rate $f^\star(0,T)$, the mean reverting parameter a and the volatility $\sigma(t)$. Since the forward rate is observed, $\theta(T)$ will depend on our choice of a and $\sigma(t)$. Since we have already let the market decide $\theta(t)$ for us through Proposition 4.6, it would be satisfactory if the market could also decide the volatility $\sigma(t)$. This can be done by considering swap options commonly known as swaptions. We will be using the following definitions found in Björk (see (2)).

Definition 4.7. LIBOR forward rate
Let $p_i(t)$ denote the zero coupon bond price $p(t,T_i)$ and let $L_i(t)$ denote the LIBOR forward rate contracted at t, for the period $[T_i - 1, T_i]$,i.e,

$$L_i(t) = \frac{1}{T_i - T_{i-1}} \cdot \frac{p_{i-1}(t) - p_i(t)}{p_i(t)}, \quad i = 1, \ldots, N.$$

Definition 4.8. Payer swap contract
The payments in a $T_n \times (T_N - T_n)$ payer swap are a follows:

- Payments will be made and received at $T_{n+1}, T_{n+2}, \ldots, T_N$.
- For every elementary period $[T_i, T_{i+1}]$, $i = n, \ldots, N-1$, the LIBOR rate $L_{i+1}(T_i)$ is set at time T_i, and the floating leg

  $$(T_{i+1} - T_i) \cdot L_{i+1}(T_i) = \frac{p_i(T_i) - p_{i+1}(T_i)}{p_{i+1}(T_i)} = \frac{1}{p_{i+1}(T_i)} - 1,$$

  is received at T_{i+1}.
- For the same period the fixed leg $(T_{i+1} - T_i) \cdot K$ is payed at T_{i+1}.

The value of such a payer swap at time $t < T_n$ is denoted $\mathbf{PS}_n^N(t;K)$.

The payer swap defined above is a so called fixed for floating swap, allowing the holder of the contract to know what they will be paying, regardless of future interest rate movements. For a **receiver swap** the payments go in the other direction, so the holder receives a fixed rate and pays floating payments.
With these two definitions, we have the following proposition.

Proposition 4.9. ***Value of floating payment***
The arbitrage free value at time t of a floating payment $(T_{i+1} - T_i) \cdot L_{i+1}(T_i)$ at time T_{i+1} is

$$p(t, T_i) - p(t, T_{i+1}).$$

Proof. The discounted value of the payment at time T_{i+1} is

$$\begin{aligned} p_{i+1}(t)(T_{i+1} - T_i) \cdot L_{i+1}(t) &= p_{i+1}(t)(T_{i+1} - T_i) \cdot \underbrace{\frac{1}{T_{i+1} - T_i} \cdot \frac{p_i(t) - p_{i+1}(t)}{p_{i+1}(t)}}_{\text{Definition 4.7}} \\ &= p_{i+1}(t) \frac{p_i(t) - p_{i+1}(t)}{p_{i+1}(t)} \\ &= p_i(t) - p_{i+1}(t) = p(t, T_i) - p(t, T_{i+1}). \end{aligned}$$

□

Using Proposition 4.9 the total value of the floating payments at time $t \leq T_n$ as

$$\sum_{i=n}^{N-1} [p(t, T_i) - p(t, T_{i+1})] = p(t, T_n) - p(t, T_N) = p_n(t) - p_N(t).$$

Furthermore, the total value of the fixed payments are at time $t < T_n$

$$\sum_{i=n}^{N-1} p(t, T_{i+1})(T_{i+1} - T_i)K = K \sum_{i=n}^{N-1} (T_{i+1} - T_i) p_{i+1}(t).$$

The sum in the right-hand side is an important object, deserving of its own definition

Definition 4.10. Accrual factor
Fore each pair n, N with $n < N$, the process $S_n^N(t)$ is defined by

$$S_n^N(t) = \sum_{i=n+1}^{N} (T_i - T_{i-1})\, p(t, T_i).$$

S_n^N is referred to as the **accrual factor** or as **the present value of a basis point**.

The value of the payer swap $\mathbf{PS}_n^N(t; K)$ is the difference between the floating leg and the fixed leg, thus given by

$$\mathbf{PS}_n^N(t; K) = p_n(t) - p_N(t) - K S_n^N(t).$$

For some K, the value of the swap is 0, which gives us the next definition

Definition 4.11. Forward Swap Rate
The forward swap rate $R_n^N(t)$ of the $T_n \times (T_N - T_n)$ swap is the value K for which $\mathbf{PS}_n^N(t;K) = 0$, i.e.

$$R_n^N(t) = \frac{p_n(t) - p_N(t)}{S_n^N(t)}.$$

Using Definition 4.11, the arbitrage free price of a payer swap with swap rate K can be rewritten as

$$\mathbf{PS}_n^N(t;K) = \left(R_n^N(t) - K\right) S_n^N(t).$$

An option written on the swap described by $\mathbf{PS}_n^N(t;K)$, gives the holder the right but not the obligation to enter the swap contract at T_n. Such options are known as swap-options, or swaptions, and can be priced using Black's formula for swaptions.

Definition 4.12. Black's Formula for Swaptions
The Black-76 formula for a $T_n \times (T_N - T_n)$ payer swaption with strike K is defined as

$$\mathbf{PSN}_n^N(t) = S_n^N(t)\left\{R_n^N(t)N[d_1] - KN[d_2]\right\},$$

where

$$d_1 = \frac{1}{\sigma_{n,N}\sqrt{T_N - t}}\left[\ln\left(\frac{R_n^N(t)}{K}\right) + \frac{1}{2}\sigma_{n,N}^2(T_N - t)\right],$$
$$d_2 = d1 - \sigma_{n,N}\sqrt{T_N - t}.$$

The constant $\sigma_{n,N}$ is known as the **Black volatility**. Given a market price for the swaption, the Black volatility implied by the Black formula is referred to as the **implied Black volatility**, and we denote it by σ_{imp}.

Furthermore, according to Wilmott (see (16)), the Black-Scholes formulae are still valid when volatility is time dependent provided we use

$$\sqrt{\frac{1}{T-t}\int_t^T \sigma(s)^2 ds}, \tag{4.12}$$

in place of σ, i.e. now we use

$$\begin{aligned} d_1 &= \frac{\ln\left(\frac{R_n^N(t)}{K}\right) + \frac{1}{2}\int_t^{T_N}\sigma(s)^2 ds}{\sqrt{\int_t^{T_N}\sigma(s)^T ds}}, \\ d_2 &= \frac{\ln\left(\frac{R_n^N(t)}{K}\right) - \frac{1}{2}\int_t^{T_N}\sigma(s)^2 ds}{\sqrt{\int_t^{T_N}\sigma(s)^2 ds}}. \end{aligned} \tag{4.13}$$

Now, for some fixed a, t, T and a set of market factors such as the prices of various discount bonds, it is clear that the function in Definition 4.12 depends only on σ, and it can be written as

$$\mathbf{PSN}_n^N(\sigma; t, K, a, \mathbf{p}^\star).$$

Suppose now that there is a swaption with price $C^\star$, on the underlying swap $\mathbf{PS}_n^N$ with swap rate K. Using Definition 4.12, we should therefore have

$$C^\star = \mathbf{PSN}_n^N(\sigma_{imp}; 0, K, a, \mathbf{p}^\star). \tag{4.14}$$

How to determine σ_{imp} is discussed in Section 5.7.

4.4 The case when $\sigma(t)$ is piecewise constant

Suppose there are a set of n observable swaptions in the market with maturities $T_1 < T_2 < \cdots < T_n$, and that between each maturity, the volatility is constant. The expression for σ then becomes

$$\sigma(t) = \left(\sum_{i=0}^{n} 1_{\{t \in [T_{i-1}, T_i]\}} \sigma_i \right) + 1_{\{t \in [T_n, \infty]\}} \sigma_n, \tag{4.15}$$

where $T_0 = 0$. We further define

$$\begin{aligned} T_+(t) &= \min \{i : T_i \geq t\}, \\ T_-(t) &= \max \{i : T_i \leq t\}, \end{aligned}$$

in other words, $T_+(t)$ is the index of the first maturity *after* (or at) t, and $T_-(t)$ for the last maturity *before* (or at) t. Note that if

$$t = T_i, \quad \text{then} \quad T_+(t) = T_-(t) = i.$$

With the expression for $\sigma(t)$ in (4.15) inserted into Proposition 4.6, the result is

$$\begin{aligned} \varphi(t) &= f^\star(0,t) + \frac{1}{2a^2} \frac{\partial}{\partial t} \left(\sum_{i=0}^{T_-(t)-1} \sigma_i^2 \int_{T_i}^{T_{i+1}} \left(1 - e^{-a(t-u)}\right)^2 du \right. \\ &\qquad \left. + \sigma_{T_-(t)}^2 \int_{T_{T_-(t)}}^{t} \left(1 - e^{-a(t-u)}\right)^2 du \right) \\ &= f^\star(0,t) + \frac{1}{2a^2} \frac{\partial}{\partial t} \left(\sum_{i=0}^{T_-(t)-1} \sigma_i^2 \left[u - \frac{2e^{-a(t-u)}}{a} + \frac{e^{-2a(t-u)}}{2a} \right]_{T_i}^{T_{i+1}} \right. \\ &\qquad \left. + \sigma_{T_-(t)}^2 \left[u - \frac{2e^{-a(t-u)}}{a} + \frac{e^{-2a(t-u)}}{2a} \right]_{T_{T_-(t)}}^{t} \right) \\ &= f^\star(0,t) + \frac{1}{2a^2} \left(\sum_{i=0}^{T_-(t)-1} \sigma_i^2 \left[2e^{-a(t-u)} - e^{-2a(t-u)} \right]_{T_i}^{T_{i+1}} \right. \\ &\qquad \left. + \sigma_{T_-(t)}^2 \left[2e^{-a(t-u)} - e^{-2a(t-u)} \right]_{T_{T_-(t)}}^{t} \right) \\ &= f^\star(0,t) + \frac{1}{2a^2} \left(\sum_{i=0}^{T_-(t)-1} \sigma_i^2 \left(\begin{array}{l} 2e^{-a(t-T_{i+1})} - e^{-2a(t-T_{i+1})} \\ \quad -2e^{-a(t-T_i)} + e^{-2a(t-T_i)} \end{array} \right) \right. \\ &\qquad \left. + \sigma_{T_-(t)}^2 \left(2 - 1 - 2e^{-a(t-T_{T_-(t)})} + e^{-2a(t-T_{T_-(t)})} \right) \right). \end{aligned} \tag{4.16}$$

In the same manner, the variance (4.5) becomes

$$
\begin{aligned}
Var[r(t)|\mathcal{F}_s] =& \int_s^t e^{-2a(t-u)}\sigma(u)^2 du \\
=& \int_s^{T_{T_+(s)}} e^{-2a(t-u)}\sigma(u)^2 du + \int_{T_{T_+(s)}}^{T_{T_-(t)}} e^{-2a(t-u)}\sigma(u)^2 du \\
&+ \int_{T_{T_-(t)}}^{t} e^{-2a(t-u)}\sigma(u)^2 du \\
=& \sigma^2_{T_+(s)} \left[\frac{e^{-2a(t-u)}}{2a}\right]_s^{T_{T_+(s)}} + \sum_{i=T_+(s)}^{T_-(t)} \sigma_i^2 \left[\frac{e^{-2a(t-u)}}{2a}\right]_{T_i}^{T_{i+1}} \\
&+ \sigma^2_{T_-(t)} \left[\frac{e^{-2a(t-u)}}{2a}\right]_{T_{T_-(t)}}^{t} \\
=& \frac{\sigma^2_{T_+(s)}}{2a}\left(e^{-2a(t-T_{T_+(s)})} - e^{-2a(t-s)}\right) \\
&+ \sum_{i=T_+(s)}^{T_-(t)} \frac{\sigma_i^2}{2a}\left(e^{-2a(t-T_{i+1})} - e^{-2a(t-T_i)}\right) \\
&+ \frac{\sigma^2_{T_-(t)}}{2a}\left(1 - e^{-2a(t-T_{T_-(t)})}\right).
\end{aligned}
\tag{4.17}
$$

Furthermore, the expressions for d_1 and d_2 in Equations (4.13) become

$$
d_1 = \frac{\ln\left(\frac{R_n^N(t)}{K}\right) + \frac{1}{2}\left(\sigma^2_{T_-(t)}\left(T_{T_+(t)} - t\right) + \sum_{i=T_+(t)}^{N-1}\sigma_i^2\left(T_{i+1} - T_i\right)\right)}{\sqrt{\sigma^2_{T_-(t)}\left(T_{T_+(t)} - t\right) + \sum_{i=T_+(t)}^{N-1}\sigma_i^2\left(T_{i+1} - T_i\right)}}, \tag{4.18}
$$

and

$$
d_2 = \frac{\ln\left(\frac{R_n^N(t)}{K}\right) - \frac{1}{2}\left(\sigma^2_{T_-(t)}\left(T_{T_+(t)} - t\right) + \sum_{i=T_+(t)}^{N-1}\sigma_i^2\left(T_{i+1} - T_i\right)\right)}{\sqrt{\sigma^2_{T_-(t)}\left(T_{T_+(t)} - t\right) + \sum_{i=T_+(t)}^{N-1}\sigma_i^2\left(T_{i+1} - T_i\right)}}, \tag{4.19}
$$

which comes from the fact that

$$
\begin{aligned}
\int_t^{T_N} \sigma(s)^2 ds &= \int_t^{T_{T_+(t)}} \sigma(s)^2 ds + \int_{T_{T_+(t)}}^{T_N} \sigma(s)^2 ds \\
&= \sigma^2_{T_-(t)} \int_t^{T_{T_+(t)}} ds + \sum_{i=T_+(t)}^{N-1} \sigma_i^2 \int_{T_i}^{T_{i+1}} ds \\
&= \sigma^2_{T_-(t)}\left(T_{T_+(t)} - t\right) + \sum_{i=T_+(t)}^{N-1} \sigma_i^2\left(T_{i+1} - T_i\right).
\end{aligned}
$$

Chapter 5

Calibrating the Hull-White model

In Chapter 4 we derived an expression for θ in Definition 4.4, and showed how σ is related to swaption prices. In this chapter we will develop a method for calibrating the model in Definition 4.4 using Adjoint Algorithmic Differentiation. At the end, we will derive the partial derivatives of the calibration parameters with respect to the final short rate curve. The Chapter is structured as to present the forward pass equations, then the corresponding adjoint versions in a subsection before moving on to the next method.

5.1 Notations

Before going further into the calibration and adjoint versions of equations discussed in Chapter 4, some notations must first be introduced.
Maturity T^s: The time of maturity of traded swaptions are denoted by T^s. Furthermore, since the code is implemented in R, the first index of a vector is 1, not 0 as in many other coding languages. Therefore, the vector of maturities $\mathbf{T}^s = [T_1^s = 0, T_2^s, T_3^s, \ldots, T_n^s]$, where $T_i^s < T_j^s$ for $i < j$.
Maturity T^r: The time of maturity for a quoted discount bond is denoted T^r, and the corresponding vector of maturities is $\mathbf{T}^r\,[T_1^r, T_2^r, \ldots, T_n^r]$, where $T_i^r < T_j^r$ for $i < j$. Note that the length of the vectors $\mathbf{T}^r$ and $\mathbf{T}^s$ may differ.
Superscript $\sigma^{(i)}$: Between two maturities T_i^s and T_{i+1}^s, a constant σ denoted by $\sigma^{(i)}$ applies.
Subscript σ_i: For any time t_i along the grid, there is a corresponding σ_i. It follows from above that $\left\{\sigma_i : t_i \in [T_j^s, T_{j+1}^s)\right\} = \sigma^{(j)}$.
Short rate r: The short rate $r(t)$ at time t_i is denoted r_i.
Less-than function $L(t_k)$: The function $L(t_k)$ returns the index of the swaption maturity that is nearest, but *less than*, or equal, to t_k. It is the function

$$L(t_k) = \max\left\{i : T_i^s \leq t_k\right\}. \tag{5.1}$$

Hence, for any σ_k,

$$\sigma_k = \sigma^{(L(t_k))}. \tag{5.2}$$

Greater-than function $G(t_k)$: The function $G(t_k)$ returns the index of the swaption maturity that is nearest, but *more than* t_k. It is the function

$$G(t_k) = \min\{i : T_i^s > t_k\}. \tag{5.3}$$

5.2 The Assignment Function

As mentioned earlier, in order to simulate the short rate at some future time T, the time axis need to be discretized. When doing so, we end up with a time array

$$\mathbf{t} = [t_1 = 0, t_2, t_3, \ldots, t_{N-1}, t_N],$$

where t_N is the time at the end of the simulation. For each time t_i, there is a corresponding short rate r_i, such that $r(t_i) = r_i$, producing a similar vector for the short rates as

$$\mathbf{r} = [r_1 = r(0), r_2, r_3, \ldots, r_{N-1}, r_N].$$

How these r_i's are determined is discussed in Section 5.5 below. Furthermore, the vector of φ's

$$\varphi = [\varphi_1, \varphi_2, \varphi_3, \ldots, \varphi_{N-1}, \varphi_N],$$

is assigned using equation (4.16), and how that is done is discussed in further detail in Section 5.4 below. One easy assignment function however, is the one assigning the various σ's into a vector.

Suppose there are n swaptions with good enough liquidity that we can assume their prices correspond to the market's implied volatility. Suppose further that these swaptions have different times of maturity $T_1^s < T_2^s < \cdots < T_n^s$. Then all σ_is between two of these maturities are equal and the assignment function is

$$\sigma_i = \begin{cases} \sigma^{(1)} & \text{if } t_i < T_2^s \\ \sigma^{(2)} & \text{if } T_2^s \leq t_i < T_3^s \\ \vdots & \vdots \\ \sigma^{(n)} & \text{if } T_n^s \leq t_i \end{cases}.$$

For example, suppose there are 5 swaptions trading, with maturities at times t_{10}, t_{20}, t_{30}, t_{40} and t_{50}. Suppose further that the corresponding implied volatilities are 0.2, 0.3, 0.25, 0.3 and 0.35 respectively. Then the assignment function for σ would produce the graph seen in Figure 5.1.

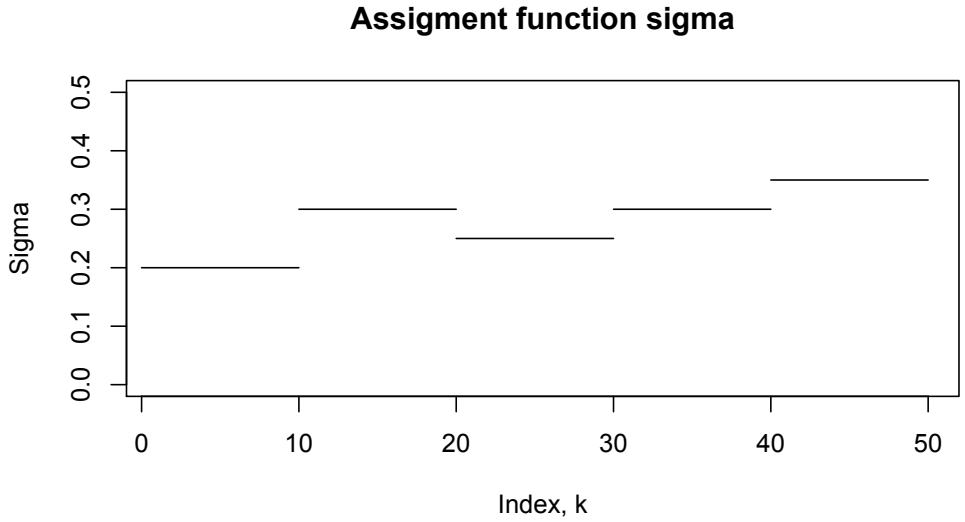

Figure 5.1: Example output of the σ-Assignment Function.

As the objective is to calibrate these $\sigma^{(k)}$s, the adjoint variables of the assignment function is needed. Since the σ_ks are just simple assignments, the adjoint variables are simply

$$\bar{\sigma}^{(k)} = \sum_{\left\{i:t_i \in \left[T^s_{k-1}, T^s_k\right)\right\}} \bar{\sigma}_i. \tag{5.4}$$

5.3 Choosing $f^\star(0,T)$

From Proposition 4.6, it is clear that we need the instantaneous short rate $f^\star(0,T)$ for all maturities T in order to compute the short rate $r(T)$ at that time. However, since there are only a finite number of quoted rates $r_i^\star$ and associated maturities T_i^r available in the market, a method for interpolating the instantaneous short rate to some $T \notin \mathbf{T}^r$ is needed. First of all, for some observed interest rate $r_i^\star$, associated with a bond with maturity T_i^r, the price of the bond is

$$p^\star(0,T_i^r) = e^{-\int_0^{T_i^r} r_i^\star du} = e^{-r_i^\star \int_0^{T_i^r} du} = e^{-r_i^\star T_i^r}.$$

From the relationship between the instantaneous forward rate and bond prices,

$$p(t,T) = e^{-\int_t^T f(t,u)du} \Rightarrow f(t,T) = -\frac{\partial \ln\left(p(t,T)\right)}{\partial T},$$

it is clear that $f(0,T)$ must satisfy

$$-\int_0^{T_i^r} f^\star(0,u)du = -\ln\left(p^\star(0,T_i^r)\right) = -\int_0^{T_i^r} r_i^\star du = -r_i^\star T_i^r, \tag{5.5}$$

for all maturities T_i^r. We will be using 2 assumptions regarding $f(0,T)$, specifically

- During the first period, $T < T_1^r$, the instantaneous short rate is equal to the rate corresponding to the first bond, i.e. $f(0,T) = r_1^\star, \quad T \leq T_1^r$.
- For the other periods, $f(0,T)$ is a linear function

$$f(0,T) = f(0,T_i^r)\left(\frac{T_{i+1}^r - T}{T_{i+1}^r - T_i^r}\right) + f(0,T_{i+1}^r)\left(\frac{T - T_i^r}{T_{i+1}^r - T_i^r}\right), \quad T \in \left[T_i^r, T_{i+1}^r\right),$$

  such that $f(0,T)$ fulfills the condition given by (5.5).

The assumptions about $f(0,T)$ imply that $f(0,T_1^r) = r_1^\star$ and

$$\int_0^{T_1^r} f(0,u)du = r_1^\star T_1^r.$$

Furthermore, $f(0,T_2^r)$ must solve

$$\begin{aligned}
\int_0^{T_2^r} f(0,u)du &= \int_0^{T_1^r} f(0,u)du + \int_{T_1^r}^{T_2^R} f(0,u)du \\
&= r_1^\star T_1^r + \frac{f(0,T_1^r)}{T_2^r - T_1^r}\left[T_2^r u - \frac{u^2}{2}\right]_{T_1^r}^{T_2^r} + \frac{f(0,T_2^r)}{T_2^r - T_1^r}\left[\frac{u^2}{2} - T_1^r u\right]_{T_1^r}^{T_2^r} \\
&= r_1^\star T_1^r + \frac{f(0,T_1^r)}{T_2^r - T_1^r}\left({T_2^r}^2 - \frac{{T_2^r}^2}{2} - T_2^r T_1^r + \frac{{T_1^r}^2}{2}\right) \\
&\quad + \frac{f(0,T_2^r)}{T_2^r - T_1^r}\left(\frac{{T_2^r}^2}{2} - T_1^r T_2^r - \frac{{T_1^r}^2}{2} + {T_1^r}^2\right) \\
&= r_1^\star T_1^r + \frac{f(0,T_1^r)}{2\left(T_2^r - T_1^r\right)}\left(T_2^r - T_1^r\right)^2 + \frac{f(0,T_2^r)}{2\left(T_2^r - T_1^r\right)}\left(T_2^r - T_1^r\right)^2 \\
&= r_1^\star T_1^r + \frac{f(0,T_1^r)}{2}\left(T_2^r - T_1^r\right) + \frac{f(0,T_2^r)}{2}\left(T_2^r - T_1^r\right) = r_2^\star T_2^r.
\end{aligned}$$

Thus

$$f(0,T_2^r) = \frac{2\left(r_2^\star T_2^r - r_1^\star T_1^r\right)}{T_2^r - T_1^r} - f(0,T_1^r) = \frac{2\left(r_2^\star T_2^r - r_1^\star T_1^r\right)}{T_2^r - T_1^r} - r_1^\star T_1^r.$$

Similar calculations show that

$$f(0,T_{i+1}^r) = \frac{2\left(r_{i+1}^\star T_{i+1}^r - r_i^\star T_i^r\right)}{T_{i+1}^r - T_i^r} - f(0,T_i^r), \text{ if } i > 0. \tag{5.6}$$

Hence $f(0,T)$ can be written as

$$f(T,\mathbf{r};\mathbf{T}^r) = \begin{cases} r_1^\star & \text{if } T \leq T_1^r \\ f(0,T_i^r)\left(\frac{T_{i+1}^r - T}{T_{i+1}^r - T_i^r}\right) + f(0,T_{i+1}^r)\left(\frac{T - T_i^r}{T_{i+1}^r - T_i^r}\right) & \text{if } T \in \left[T_i^r, T_{i+1}^r\right), \end{cases} \tag{5.7}$$

where $f(0,T_i^r)$ is defined by (5.6). By adjusting the notation, such that

$$f(0,T_i^r) = H_i,$$

allows (5.6) to be rewritten as

$$H_i = \frac{2\left(r_i^\star T_i^r - r_{i-1}^\star T_{i-1}^r\right)}{T_i^r - T_{i-1}^r} - H_{i-1}, \text{ if } i > 1,$$

with

$$H_1 = r_1^\star T_1^r.$$

5.3.1 Adjoint of $f^\star(0,T)$

To find the adjoint variables $\bar{r}_i^\star$ of $f(0,T)$, we will use the concept of chaining discussed in Section 2.4. First note that

$$\frac{\partial H_1}{\partial r_1^\star} = T_1^r, \text{ and } \frac{\partial H_1}{\partial r_i^\star} = 0, i > 1.$$

To find an expression for the derivatives of H_i, $i > 1$, consider these examples;

$$\begin{aligned}
&\frac{\partial H_2}{\partial r_1^\star} = \frac{-2T_1^r}{T_2^r - T_1^r} - T_1^r && \frac{\partial H_2}{\partial r_2^\star} = \frac{2T_2^r}{T_2^r - T_1^r} \\
&\frac{\partial H_3}{\partial r_1^\star} = -\frac{\partial H_2}{\partial r_1^\star} = (-1)\left(\frac{-2T_1^r}{T_2^r - T_1^r} - T_1^r\right) && \frac{\partial H_3}{\partial r_2^\star} = \frac{-2T_2^r}{T_3^r - T_2^r} - \frac{2T_2^r}{T_2^r - T_1^r} \\
&\frac{\partial H_4}{\partial r_1^\star} = -\frac{\partial H_3}{\partial r_1^\star} = \frac{\partial H_2}{\partial r_1^\star} && \frac{\partial H_4}{\partial r_2^\star} = -\frac{\partial H_3}{\partial r_2^\star} \\
&\quad\vdots && \quad\vdots \\
&\frac{\partial H_n}{\partial r_1^\star} = (-1)^n \frac{\partial H_2}{\partial r_1^\star} && \frac{\partial H_n}{\partial r_2^\star} = (-1)^{n+1} \frac{\partial H_3}{\partial r_2^\star}.
\end{aligned}$$

Hence the general formula can be expressed as

$$\frac{\partial H_i}{\partial r_j^\star} = (-1)^{i+j+1} \frac{\partial H_{j+1}}{\partial r_j^\star} = (-1)^{i+j+1} \left(\frac{-2T_j^r}{T_{j+1}^r - T_j^r} - \frac{\partial H_j}{\partial r_j^\star}\right), \quad \text{for } i > j,$$

where

$$\frac{\partial H_1}{\partial r_1^\star} = T_1^r \quad \text{and} \quad \frac{\partial H_j}{\partial r_j^\star} = \frac{2T_j^r}{T_j^r - T_{j-1}^r}, \text{ if } j > 1,$$

and

$$\frac{\partial H_i}{\partial r_j^\star} = 0 \text{ if } i < j.$$

In a more condensed form this can be written as

$$\frac{\partial H_i}{\partial r_j^\star} = \begin{cases} 0 & , i < j \\ T_1^r & , i = j = 1 \\ \frac{2T_j^r}{T_j^r - T_{j-1}^r} & , i = j > 1 \\ (-1)^{i+j+1}\left(\frac{-2T_j^r}{T_{j+1}^r - T_j^r} - \frac{\partial H_j}{\partial r_j^\star}\right) & , i > j \end{cases}. \tag{5.8}$$

Thus the adjoint of an observed rate $r_j^\star$ on $f(0,T)$ is

$$\bar{r}_j^\star = \frac{\partial f(0,T)}{\partial r_j^\star} = \frac{\partial H_i}{\partial r_j^\star}\left(\frac{T_{i+1}^r - T}{T_{i+1}^r - T_i^r}\right) + \frac{\partial H_{i+1}}{\partial r_j^\star}\left(\frac{T - T_i^r}{T_{i+1}^r - T_i^r}\right), \text{ for } T \in \left[T_i^r, T_{i+1}^r\right], \tag{5.9}$$

where $i > 0$. For $T < T_1^r$, the adjoint become

$$\bar{r}_j^\star = \frac{\partial f(0,T)}{\partial r_j^\star} = \begin{cases} 1 & \text{if } j = 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.10}$$

5.4 Adjoint of $\varphi(t)$

Recall (4.16) with piecewise constant σ, which using the notation introduced in Section 5.1 is written as

$$\varphi_k = f(0,t_k) + \frac{1}{2a^2}\left(\sum_{i=1}^{L(t_k)-1} {\sigma^{(i)}}^2 \begin{pmatrix} 2e^{-a(t_k-T^s_{i+1})} - e^{-2a(t_k-T^s_{i+1})} \\ -2e^{-a(t_k-T^s_i)} + e^{-2a(t_k-T^s_i)} \end{pmatrix} + {\sigma^{(L(t_k))}}^2 \left(1 - 2e^{-a(t_k-T^s_{L(t_k)})} + e^{-2a(t_k-T^s_{L(t_k)})}\right)\right).$$

To avoid cumbersome notations, we introduce the functions $g_i(t_k)$, and $h_i(t_k)$, defined as

$$g_i(t_k) = \begin{pmatrix} 2e^{-a(t_k-T^s_{i+1})} - e^{-2a(t_k-T^s_{i+1})} \\ -2e^{-a(t_k-T^s_i)} + e^{-2a(t_k-T^s_i)} \end{pmatrix}$$
$$h_i(t_k) = \left(1 - 2e^{-a(t_k-T^s_i)} + e^{-2a(t_k-T^s_i)}\right),$$

hence

$$\varphi_k = f(0,t_k) + \frac{1}{2a^2}\left(\sum_{i=1}^{L(t_k)-1} {\sigma^{(i)}}^2 g_i(t_k) + {\sigma^{(L(t_k))}}^2 h_{L(t_k)}\right).$$

Since $f(0,t_k)$ is a function of the observed market rates $r^\star_i$, so is φ_k, and the adjoint variables of the observed rates are

$$\bar{r}^\star_i = \frac{\partial \varphi_k}{\partial r^\star_i} = \underbrace{\frac{\partial \varphi_k}{\partial f(0,t_k)}}_{=1} \underbrace{\frac{\partial f(0,t_k)}{\partial r^\star_i}}_{\text{Equations (5.9) and (5.10)}},$$

hence, the expressions in (5.9) and (5.10) applies to φ_k as well without modification. Furthermore, the adjoint variables of σ become

$$\begin{aligned} \bar{\sigma}^{(i)}_{\varphi_k} &= \frac{\sigma^{(j)}}{a^2} g_i(t_k), && i \in [1, L(t_k)), \\ \bar{\sigma}^{(L(t_k))}_{\varphi_k} &= \frac{\sigma^{(j)}}{a^2} h_j(t_k), && i \in [L(t_k), L(t_k)+1), \\ \bar{\sigma}^{(L(t_k))}_{\varphi_k} &= 0, && i \geq L(t_k)+1, \end{aligned} \tag{5.11}$$

where the subscript is there to emphasise that these are the adjoint variables of sigma with respect to φ_k.

For each φ_k, there will therefore be a vector of n adjoint $\sigma^{(i)}$s. For φ_1 this vector is simply

$$\bar{\sigma}_{\varphi_1} = \frac{1}{a^2}\begin{pmatrix} \sigma^{(1)} h_1(t_1) & 0 & 0 & \dots & 0 \end{pmatrix}^T,$$

and for φ_2 it is

$$\bar{\sigma}_{\varphi_2} = \frac{1}{a^2}\begin{pmatrix} \sigma^{(1)} h_1(t_2) & 0 & 0 & \dots & 0 \end{pmatrix}^T.$$

This will continue until $T_1^s < t_k \leq T_2^s$, in which case

$$\bar{\sigma}_{\varphi_k} = \frac{1}{a^2}\begin{pmatrix}\sigma^{(1)}g_1(t_k) & \sigma^{(2)}h_2(t_k) & 0 & \dots & 0\end{pmatrix}^T.$$

And similarly for $T_2^s < t_k \leq T_3^s$, the vector becomes

$$\bar{\sigma}_{\varphi_k} = \frac{1}{a^2}\begin{pmatrix}\sigma^{(1)}g_1(t_k) & \sigma^{(2)}g_2(t_k) & \sigma^{(3)}h_3(t_k) & 0 & \dots & 0\end{pmatrix}^T.$$

The full set of adjoint variables $\bar{\sigma}_\varphi$ can thus be represented by a matrix. With the n different $\sigma^{(i)}$s, and the N different φ_ks, the matrix become

$$\bar{\sigma}_\varphi = \begin{pmatrix} | & | & & | \\ \bar{\sigma}_{\varphi_1} & \bar{\sigma}_{\varphi_2} & \dots & \bar{\sigma}_{\varphi_N} \\ | & | & & | \end{pmatrix}_{[n\times N]}.$$

We will be referring to the adjoint variable $\bar{\sigma}_{\varphi_k}^{(i)}$, i.e. the derivative of the kth φ, with respect to the ith block of σs, as

$$\bar{\sigma}_{\varphi_k}^{(i)} = \bar{\sigma}_\varphi[i,k],$$

which is consistent with how elements are referenced in the programming language R.

5.5 Discretization of the diffusion process

The approach in this section will be similar to the one outlined in Section 3.1, but in this case the diffusion process is described by (4.9),

$$r(t) = \varphi(t) + r(s)e^{-a(t-s)} - \varphi(s)e^{-a(t-s)} + \int_s^t e^{-a(t-u)}\sigma(u)dW(u).$$

Recall that $r(t)$ conditional on $\mathcal{F}_s$ is normally distributed with mean

$$\mathbb{E}[r(t)|\mathcal{F}_s] = \varphi(t) + e^{-a(t-s)}(r(s) - \varphi(s)),$$

and variance

$$Var\{r(t)|\mathcal{F}_s\} = \int_s^t e^{-2a(t-u)}\sigma(u)^2du.$$

Because of the piecewise constant nature of our chosen $\sigma(t)$, the propagation from t_i to $t_{i+1} = t_i + dt$ can be written as

$$r(t_{i+1}) = \varphi(t_{i+1}) + e^{-adt}(r(t_i) - \varphi(t_i)) + \sigma(t_i)N(0,1)\sqrt{\int_{t_i}^{t_{i+1}} e^{-2a(t_{i+1}-u)}du}. \tag{5.12}$$

With the notations introduced in Section 5.1, including the vector $\mathbf{Z}$ of $iid \sim N(0,1)$, (5.12) can be written as

$$r_{i+1} = \varphi_{i+1} + e^{-adt}(r_i - \varphi_i) + \sigma_i Z_i\sqrt{\int_{t_i}^{t_{i+1}} e^{-2a(t_{i+1}-u)}du}.$$

The integral under the root is

$$\int_{t_i}^{t_{i+1}} e^{-2a(t_{i+1}-u)}du = \left[\frac{e^{-2a(t_{i+1}-u)}}{2a}\right]_{t_i}^{t_{i+1}} = \frac{1-e^{-2adt}}{2a}.$$

To avoid cumbersome notations, let $\alpha = e^{-adt}$, and $\beta = \sqrt{(1-\alpha^2)/2a}$, which allows (5.12) to be written as

$$r_{k+1} = \varphi_{k+1} + \alpha(r_k - \varphi_k) + \sigma^{(L(t_k)}Z_k\sqrt{\beta}.$$

In order to calculate the derivatives of r_k generated by the model, with respect to the quoted interest rates $r_i^\star$, the following equation needs to be solved;

$$\begin{aligned}\frac{\partial r_k}{\partial r_i^\star} &= \frac{\partial \varphi_k}{\partial r_i^\star} - \alpha\frac{\partial \varphi_{k-1}}{\partial r_i^\star} + \alpha\frac{\partial r_{k-1}}{\partial r_i^\star} \\ &= \frac{\partial f(0,t_k)}{\partial r_i^\star} - \alpha\frac{\partial f(0,t_{k-1})}{\partial r_i^\star} + \alpha\frac{\partial r_{k-1}}{\partial r_i^\star}.\end{aligned}$$

The first two terms on the right-hand side are defined by (5.9) and (5.10). From the last term on the right-hand side, it is clear that this method needs to be solved recursively. However, recall from Section 2.6, that it is possible to compute the adjoint variables during the forward pass in order to avoid recursion. Hence the adjoint variable of some $r_i^\star$ on r_k is simply

$$\bar{r}^\star_{i,r_k} = \frac{\partial f(0,t_k)}{\partial r_i^\star} - \alpha\frac{\partial f(0,t_{k-1})}{\partial r_i^\star} + \alpha\bar{r}^\star_{i,r_{k-1}}, \tag{5.13}$$

where $\bar{r}^\star_{i,r_{k-1}}$ is known when it is time to compute $\bar{r}^\star_{i,r_k}$.

The adjoint variable for $\sigma^{(i)}$, can be computed using the same reasoning as above, and realizing that $\sigma^{(i)}$ affects r_k both *explicitly* through $\sigma^{(L(t_k))}Z_k\sqrt{\beta}$ if $i = L(t_k)$, and *implicitly* by affecting φ_k, φ_{k-1} and potentially even r_{k-1}. The resulting adjoint is therefore

$$\begin{aligned}\bar{\sigma}^{(i)}_{r_k} &= \frac{\partial \varphi_k}{\partial \sigma^{(i)}} - \alpha\frac{\partial \varphi_{k-1}}{\partial \sigma^{(i)}} + \alpha\bar{\sigma}^{(i)}_{r_{k-1}} + Z_{k-1}\sqrt{\beta}\frac{\partial \sigma^{(L(t_k))}}{\partial \sigma^{(i)}} \\ &= \bar{\sigma}_\varphi[i,k] - \alpha\bar{\sigma}_\varphi[i,k-1] + \alpha\bar{\sigma}^{(i)}_{r_{k-1}} + Z_{k-1}\sqrt{\beta}\mathbf{1}_{[i=L(t_k)]},\end{aligned} \tag{5.14}$$

where again $\bar{\sigma}^{(i)}_{r_{k-1}}$ is known when computing $\bar{\sigma}^{(i)}_{r_k}$, and $\mathbf{1}$ denotes the indicator function.

5.6 Black's Formula for Swaptions

In Definition 4.12, Black's formula for swaptions is introduced. When using a time dependent (albeit piecewise constant) volatility, it becomes

$$\mathbf{PSN}_n^N(t_i) = S_n^N(t_i)\left(R_n^N(t_i)N[d_1] - KN[d_2]\right), \tag{5.15}$$

where

$$
\begin{aligned}
d_1 &= \frac{\ln\left(\frac{R_n^N(t_i)}{K}\right) + \frac{1}{2}\int_{t_n}^{t_N}\sigma(s)^2 ds}{\sqrt{\int_{t_n}^{t_N}\sigma(s)^2 ds}}, \\
d_2 &= \frac{\ln\left(\frac{R_n^N(t_i)}{K}\right) - \frac{1}{2}\int_{t_n}^{t_N}\sigma(s)^2 ds}{\sqrt{\int_{t_n}^{t_N}\sigma(s)^2 ds}}.
\end{aligned}
\tag{5.16}
$$

Using Algorithmic Differentiation, the aim is to find the volatilities $\sigma^{(i)}$ such that

$$
\mathbf{PSN}_{n_i}^{N_i}(0; K_i) = C_i^\star,
$$

for all observed swaption prices $C_i^\star$ with strike K_i, maturity t_{n_i} and tenor $t_{N_i} - t_{n_i}$. By realizing that the swaption price is computed at time $t = 0$, it follows that the accrual factor S_n^N and forward swap rate R_n^N are also computed at $t = 0$. This in turn implies that the discount factors $p(t, T)$ used in the computations of S_n^N and R_n^N will also be evaluated for $t = 0$. Recall that

$$
p(0, T) = e^{\int_0^T f(0,u)du},
$$

hence we can use the expression for $f(0, T)$ used in (5.7) to compute S_n^N and R_n^N.

5.6.1 Discount factors $p(0, T)$

With $f(0, T)$ as

$$
f(T, \mathbf{r}; \mathbf{T}) = \begin{cases} r_1^\star & \text{if } T \leq T_1^r \\ f(0, T_i)\left(\frac{T_{i+1}^r - T}{T_{i+1}^r - T_i^r}\right) + f(0, T_{i+1}^r)\left(\frac{T - T_i^r}{T_{i+1}^r - T_i^r}\right) & \text{if } T \in \left[T_i^r, T_{i+1}^r\right), \end{cases}
$$

we define $F(0, T)$ as

$$
\begin{aligned}
F(t) = \int_0^t f(0,u)du &= \underbrace{\int_0^{T_i^r} f(0,u)du}_{\text{Using (5.5)}} + \int_{T_i^r}^t f(0,u)du \\
&= r_i^\star T_i^r + \int_{T_i^r}^t \left(H_i\left(\frac{T_{i+1}^r - u}{T_{i+1}^r - T_i^r}\right) + H_{i+1}\left(\frac{u - T_i^r}{T_{i+1}^r - T_i^r}\right)\right) du \\
&= r_i^\star T_i^r + \frac{H_i}{T_{i+1}^r - T_i^r}\left[T_{i+1}^r u - \frac{u^2}{2}\right]_{T_i^r}^t + \frac{H_{i+1}}{T_{i+1}^r - T_i^r}\left[\frac{u^2}{2} - T_i^r u\right]_{T_i^r}^t \\
&= r_i^\star T_i^r + \frac{H_i}{T_{i+1}^r - T_i^r}\left(T_{i+1}^r t - \frac{t^2}{2} - T_{i+1}^r T_i^r + \frac{{T_i^r}^2}{2}\right) \\
&\quad + \frac{H_{i+1}}{T_{i+1}^r - T_i^r}\underbrace{\left(\frac{t^2}{2} - T_i^r t - \frac{{T_i^r}^2}{2} + {T_i^r}^2\right)}_{\frac{1}{2}\left(t - T_i^r\right)^2} \\
&= r_i^\star T_i^r + \frac{H_i\left(T_{i+1}^r t - \frac{t^2}{2} - T_{i+1}^r T_i^r + \frac{{T_i^r}^2}{2}\right)}{T_{i+1}^r - T_i^r} + \frac{H_{i+1}}{2}\frac{(t - T_i^r)^2}{T_{i+1}^r - T_i^r}.
\end{aligned}
$$

To avoid cumbersome notations, let

$$\Gamma_i(t) = \frac{T_{i+1}^r t - \frac{t^2}{2} - T_{i+1}^r T_i^r + \frac{{T_i^r}^2}{2}}{T_{i+1}^r - T_i^r},$$

and let $F(t_k) = F_k$, giving F_k as

$$F_k = \begin{cases} r_1^\star t_k & \text{if } t_k \leq T_1^r \\ r_i^\star T_i + H_i \Gamma_i(t_k) + \frac{H_{i+1}}{2} \frac{(t_k - T_i^r)^2}{T_{i+1}^r - T_i^r} & \text{if } t_k \in \left[T_i^r, T_{i+1}^r\right). \end{cases}$$

Hence, the adjoint variable for some $r_j^\star$ on F_k is

$$\bar{r}_{j,F_k}^\star = \begin{cases} t_k & \text{if } t_k \leq T_1^r \text{ and } j = 1 \\ 0 & \text{if } t_k \leq T_1^r \text{ and } j \neq 1 \\ \mathbf{1}_{[j=i]} T_i^r + \frac{\partial H_i}{\partial r_j^\star} \Gamma_i(t_k) + \frac{\partial H_{i+1}}{\partial r_j^\star} \frac{(t_k - T_i^r)^2}{2\left(T_{i+1}^r - T_i^r\right)} & \text{if } t_k \in \left[T_i^r, T_{i+1}^r\right), \end{cases}$$

were the partial derivatives $\partial H_i / \partial r_j^\star$ are defined by (5.8). It follows that

$$\frac{\partial p(0,t_k)}{\partial r_j^\star} = \frac{\partial p(0,t_k)}{\partial F_k} \frac{\partial F_k}{\partial r_j^\star} = -e^{-F_k} \bar{r}_{j,F_k}^\star = -\bar{r}_{j,F_k}^\star p(0,t_k).$$

Remark. Note that since the discount factors are evaluated at $t = 0$, they are independent of the volatility σ, hence

$$\frac{\partial p(0,t_k)}{\partial \sigma^{(j)}} = 0, \quad \forall\, j,k.$$

5.6.2 Accrual factor and forward swap rate

In order to calculate the price of a swaption, the accrual factor and forward swap rate is needed. Recall the definition of the accrual factor in Definition 4.10,

$$S_n^N(t_k) = \sum_{i=n+1}^{N} (T_i - T_{i-1})\, p(t_k, T_i),$$

where the sum is over the payments during the lifetime of the swap contract. Hence, if the swap contract is written on the 3-month interest rate with tenor 2 years, the accrual factor will be a sum over the $4 * 2$ number of payments. Furthermore, the adjoint of $r_j^\star$ of S_n^N is

$$\bar{r}_{j,S_n^N}^\star = \sum_{i=n+1}^{N} (T_i - T_{i-1}) \frac{\partial p(0,T_i)}{\partial r_j^\star} = \sum_{i=n+1}^{N} (T_{i-1} - T_i)\, \bar{r}_{j,F_i}^\star p(0,T_i).$$

The definition of the forward swap rate, Defnintion 4.11,

$$R_n^N(t_k) = \frac{p(t_k,T_n) - p(t_k,T_N)}{S_n^N(t_k)},$$

again evaluated at $t_k = 0$, yields the adjoint variables as

$$\begin{aligned}
\bar{r}^{\star}_{j,R^N_n} &= \frac{\partial R^N_n(0)}{\partial r^{\star}_j} = \\
&= \frac{-\bar{r}^{\star}_{j,F_n} p(0,T_n)}{S^N_n(0)} - \frac{-\bar{r}^{\star}_{j,F_N} p(0,T_N)}{S^N_n(0)} - \bar{r}^{\star}_{j,S^N_n} \frac{p(0,T_n) - p(0,T_N)}{S^N_n(0)^2} \\
&= \frac{1}{S^N_n(0)} \left(\bar{r}^{\star}_{j,F_N} p(0,T_N) - \bar{r}^{\star}_{j,F_n} p(0,T_n) - \bar{r}^{\star}_{j,S^N_n} R^N_n(0) \right).
\end{aligned}$$

With the above results we return to (5.15) and find its adjoint variables. This can be done by constructing a thorough forward pass of $\mathbf{PSN}^N_n(0;K)$, which is then stepped back through in order to create the adjoint variables. First of all, since σ is piecewise constant, the integrals in (5.16) can be rewritten as

$$\begin{aligned}
\int_{t_n}^{t_N} \sigma(s)^2 ds =& \sigma^{(T_-(t_n))^2} \int_{t_n}^{T_{T_+(t_n)}} ds + \sum_{i=T_+(t_n)}^{T_-(t_N)-1} \sigma^{(i)^2} \int_{T^s_i}^{T^s_{i+1}} ds \\
&+ \sigma^{(T_-(t_N))^2} \int_{T^s_{T_-(t_N)}}^{t_N} ds \\
=& \sigma^{(T_-(t_n))^2} \left(T^s_{T_+(t_n)} - t_n \right) + \sum_{i=T_+(t_n)}^{T_-(t_N)-1} \sigma^{(i)^2} \left(T^s_{i+1} - T^s_i \right) \\
&+ \sigma^{(T_-(t_N))^2} \left(t_N - T^s_{T_-(t_N)} \right),
\end{aligned}$$

where $\sigma^{(k)}$ denotes one of the constant σs, and T^s_i denotes the time of maturity of the ith swaption. By setting

$$\int_{t_n}^{t_N} \sigma(s)^2 ds = \Gamma^N_n,$$

equations (5.16) become

$$\begin{aligned}
d_1 &= \frac{\ln\left(\frac{R^N_n(0)}{K}\right) + \frac{1}{2}\Gamma^N_n}{\sqrt{\Gamma^N_n}}, \\
d_2 &= d_1 - \sqrt{\Gamma^N_n}.
\end{aligned}$$

Now let $R^N_n(0)/K = v_1$ and $\ln(v_1) = v_2$, such that

$$d_1 = \frac{v_2 - \Gamma^N_n/2}{\sqrt{\Gamma^N_n}}, \qquad d_2 = d_1 - \sqrt{\Gamma^N_n}.$$

Furthermore, recall that

$$N[x] = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-1}{2}s^2} ds, \tag{5.17}$$

which means that

$$\frac{\partial N[x]}{\partial x} = \frac{e^{-x^2/2}}{\sqrt{2\pi}}.$$

Using (5.17), (5.15) can be rewritten as

$$\mathbf{PSN}_n^N(0;K) = \frac{S_n^N(0)}{\sqrt{2\pi}} \left(R_n^N(0) \int_{-\infty}^{d_1} e^{-x^2/2} dx - K \int_{-\infty}^{d_2} e^{-x^2/2} dx \right).$$

Continuing the forward pass and creating temporary variables yields

$$\begin{aligned}
\mathbf{PSN}_n^N(0;K) &= \frac{S_n^N(0)}{\sqrt{2\pi}} \left(R_n^N(0) \int_{-\infty}^{d_1} e^{-x^2/2} dx - K \int_{-\infty}^{d_2} e^{-x^2/2} dx \right) \\
&= \underbrace{\frac{S_n^N(0)}{\sqrt{2\pi}}}_{v_3} \left(R_n^N(0) \underbrace{\int_{-\infty}^{d_1} e^{-x^2/2} dx}_{v_4} - K \underbrace{\int_{-\infty}^{d_2} e^{-x^2/2} dx}_{v_5} \right) \\
&= v_3 \left(\underbrace{R_n^N(0) v_4 - K v_5}_{v_6} \right) \\
&= v_3 v_6 = v_7
\end{aligned}$$

The corresponding backward pass is then

$$\begin{aligned}
\bar{v}_6 &= v_3, \\
\bar{v}_3 &= v_6, \\
\bar{v}_5 &= \bar{v}_6(-K) = -K v_3, \\
\bar{v}_4 &= \bar{v}_6 R_n^N(0) = R_n^N(0) v_3, \\
\bar{d}_2 &= \bar{v}_5 \frac{\partial v_5}{\partial d_2} = -K v_3 e^{-d_2^2/2}, \\
\bar{d}_1 &= \bar{v}_4 e^{-d_1^2/2} + \bar{d}_2 = \frac{S_n^N(0)}{\sqrt{2\pi}} \left(R_n^N(0) e^{-d_1^2/2} - K e^{-d_2^2/2} \right), \\
\bar{\Gamma}_n^N &= \bar{d}_1 \left(\frac{-1}{2{\Gamma_n^N}^{3/2}} - \frac{1}{4\sqrt{\Gamma_n^N}} \right) - \frac{\bar{d}_2}{2\sqrt{\Gamma_n^N}}, \\
\bar{v}_2 &= \frac{\bar{d}_1}{\sqrt{\Gamma_n^N}}, \\
\bar{v}_1 &= \frac{\bar{v}_2}{v_1}.
\end{aligned}$$

And finally

$$\begin{aligned}
\bar{S}_n^N(0) &= \frac{\bar{v}_3}{\sqrt{2\pi}} = \frac{v_6}{\sqrt{2\pi}} = \frac{\mathbf{PSN}_n^N(0;K)}{S_n^N(0)}, \\
\bar{R}_n^N(0) &= \bar{v}_6 v_4 + \frac{\bar{v}_1}{K} = \frac{S_n^N(0)}{\sqrt{2\pi}} \int_{-\infty}^{d_1} e^{-x^2/2} dx + \frac{\bar{d}_1}{R_n^N(0)\sqrt{\Gamma_n^N}}, \\
\bar{\sigma}^{(T_-(t_n))} &= 2\bar{\Gamma}_n^N \sigma^{(T_-(t_n))} \left(T^s_{T_+(t_n)} - t_n \right), \\
\bar{\sigma}^{(i)} &= 2\bar{\Gamma}_n^N \sigma^{(i)} \left(T^s_{i+1} - T^s_i \right), \quad i \in [T_+(t_n), T_-(t_N) - 1], \\
\bar{\sigma}^{(T_-(t_N))} &= 2\bar{\Gamma}_n^N \sigma^{(T_-(t_n))} \left(t_N - T^s_{T_-(t_N)} \right).
\end{aligned} \tag{5.18}$$

This means that the derivative of a swaption price, with respect to some observed interest rate $r_j^\star$ is

$$\bar{r}_j^\star = \bar{r}^\star_{j,S_n^N} \bar{S}_n^N + \bar{r}^\star_{j,R_n^N} \bar{R}_n^N.$$

5.7 Newton-Raphson

Recall that the objective by using Algorithmic Differentiation, is to find the volatilities $\sigma^{(i)}$ such that

$$\mathbf{PSN}_{n_i}^{N_i}(0;K_i) = C_i^\star,$$

for all observed swaption prices $C_i^\star$ with strike rate K, maturity t_{n_i} and tenor $t_{N_i} - t_{n_i}$. In order to solve for the implied volatilites, consider the Newton-Raphson method

Definition 5.1. Newton-Raphson
The point x where the function $f(x) = 0$ can be found by iterating

$$x_{n+1} = x_n - \frac{f(x)}{f'(x)}, \quad f'(x) = \frac{df(x)}{dx},$$

until

$$|x_{n+1} - x_n| \leq \epsilon,$$

for some cut-off value ϵ.

Proof. The tangent to a function $f(x)$ at x_0 has according to the point-slope form, the representation

$$y = f'(x_0)(x - x_0) + f(x_0).$$

At the point where $f(x) = 0$, this becomes

$$\begin{aligned}
0 &= f'(x_0)(x - x_0) + f(x_0), \\
x - x_0 &= -\frac{f(x_0)}{f'(x_0)}, \\
x &= x_0 - \frac{f(x_0)}{f'(x_0)},
\end{aligned} \tag{5.19}$$

which can be iterated until $f(x) = 0$ is reached. □

Using Definition 5.1 with

$$f(\sigma^{(i)}) = C_i^\star - \mathbf{PSN}_{n_i}^{N_i}(0; K_i),$$

it is possible to find the volatility $\sigma^{(i)}$ where $f(\sigma^{(i)}) = 0$, which gives

$$C_i^\star = \mathbf{PSN}_{n_i}^{N_i}(0; K_i)|_{\sigma=\sigma_{imp}}.$$

Since $C_i^\star$ is an observed constant, it follows that

$$\frac{\partial f(\sigma^{(i)})}{\partial \sigma^{(i)}} = -\frac{\partial \mathbf{PSN}_{n_i}^{N_i}(0; K_i)}{\partial \sigma^{(i)}} = -\bar{\sigma}^{(i)},$$

where $\bar{\sigma}^{(i)}$ is defined by (5.18). Definition 5.1 then implies that the correct *(or close enough)* implied volatility $\sigma_{imp}^{(i)}$ is found by iterating

$$\sigma_{k+1}^{(i)} = \sigma_k^{(i)} - \frac{C_i^\star - \mathbf{PSN}_{n_i}^{N_i}(0; K_i)}{-\bar{\sigma}^{(i)}}, \tag{5.20}$$

until

$$\left| \frac{C_i^\star - \mathbf{PSN}_{n_i}^{N_i}(0; K_i)}{-\bar{\sigma}^{(i)}} \right| < \epsilon, \tag{5.21}$$

for some cut-off value ϵ.

Furthermore, there are two factors that affect the price of a swaption:

- **The implied volatility** $\sigma_{imp}^{(i)}$, generated by the actors on the market can change, resulting in a new price $C_i^\star + \Delta C_i^\star$. These changes are picked up by (5.18), since

$$C_i^\star = \mathbf{PSN}_{n_i}^{N_i}(0; K_i) \Rightarrow \frac{\partial C_i^\star}{\partial \sigma^{(i)}} = \frac{\partial \mathbf{PSN}_{n_i}^{N_i}(0; K_i)}{\partial \sigma^{(i)}} = \bar{\sigma}^{(i)}.$$

- **The quoted rates** $r_i^\star$ can change, resulting in a different term structure, and subsequently a different swap value and swaption price. However, if the term structure is changed, and the swaption price stays the same, there has been a change in implied volatility. Theses scenarios require a new pass through the Newton-Raphson solver in order to obtain the altered implied volatilities. These scenarios are not covered further in this thesis, hence we assume that if the term structure is changed, the swaption prices change accordingly as to eliminate arbitrage.

5.8 Numerical Results

Numerical values to test against were provided by ICAP, one of the largest interest rate derivative brokers in the world, making their quoted prices more reliable than those from a less liquid marketplace. The data is a snapshot of the market values on the 29th of June 2017. The first objective in the calibration is to find the instantaneous forward rate discussed in Section 5.3. In Figure 5.2, the instantaneous short rate is plotted along with the quoted rates in red. The

quoted rates have been added with 0.75%, as to represent the borrow-lending gap at Sweden's Central bank, the Riksbank, giving the vector

$$\mathbf{r}^\star = \begin{bmatrix} 0.29 & 0.495 & 0.717 & 0.938 & 1.145 & 1.338 & 1.515 & 1.675 & 1.815 & 1.943 \end{bmatrix},$$

with corresponding maturities

$$\mathbf{T}^r = \begin{bmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \end{bmatrix}.$$

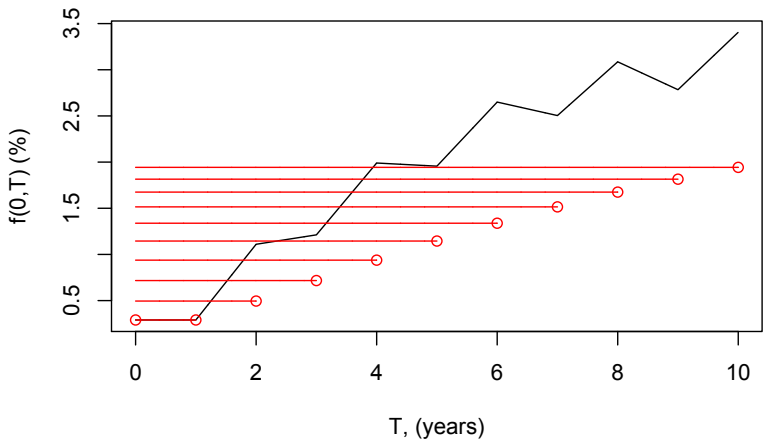

Figure 5.2: Graph of $f(0,T)$ for various T. The red lines represent the quoted rates, ending with a circle at maturity for that rate.

A question might arise as to why the quoted rates seem so much lower than the instantaneous short rate. This can be explained by the fact that the instantaneous short rate is fitted against the integrals

$$\int_0^{T_i^r} r_i^\star du = \int_0^{T_i^r} f(0,u)du, \quad \forall i.$$

Therefore, the shape of Figure 5.2, is a result of $f(0,T)$ "compensating" for being lower than r_i during most of the interval from 0 to T_i^r. The integrated curve $\int_0^T f(0,u)du$ is plotted in Figure 5.3, where the red circles represent the integrated observed values $r_i^\star T_i^r$.

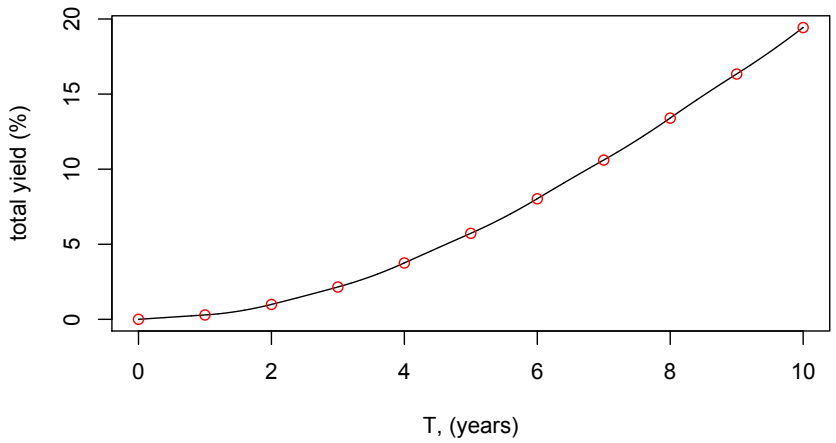

Figure 5.3: Graph of $\int_0^T f(0,u)du$ for various T. The red circles are represent the yield of the observed rates, i.e. $\int_0^{T_i^r} r_i^\star du = r_i^\star T_i^r$.

The curve fit is a lot more clear in Figure 5.2, and thus we can be confident that the observed term structure is reproduced by $f(0,T)$.

To obtain the implied volatilities, a bootstrapping method was applied whereby the first implied volatility $\sigma_{imp}^{(1)}$ is solved against the first swaption maturity T_1^s. Note that the maturity in this sense is referring to the maturity of the underlying swap contract, and not the maturity of the option. The maturity of the swaptions were constructed as T_{i-1}^s for swaption i, giving the tenor $T_i^s - T_{i-1^s}$ of the underlying swap contract. When $\sigma_{imp}^{(1)}$ is found it is "locked" and treated as a constant while solving $\sigma_{imp}^{(2)}$ using the swaption with maturity at T_2^s. Then $\sigma_{imp}^{(2)}$ is "locked" while solving for $\sigma_{imp}^{(3)}$, and so on. Furthermore, with the implied volatilities provided in the ICAP data, a set of *at-the-money* swaptions were created, meaning that the strike rate K is equal to the forward swap rate R_n^N. These swaptions were constructed on swap contracts written on the quarterly interest rate, hence there are 4 payments per year during the tenor of the swap. The implied volatilities are

$$\sigma = \begin{bmatrix} 0.347 & 0.389 & 0.404 & 0.450 & 0.410 & 0.376 \end{bmatrix},$$

with corresponding maturities

$$T^s = \begin{bmatrix} 0.5 & 1.0 & 3.0 & 5.0 & 8.0 & 10 \end{bmatrix},$$

where the time in T^s is denoted in years.

With the maturities, tenors, strike rates and prices of the swaptions corresponding to the implied volatility computed, the volatilities were reset into default values of 0.2. Then the Newton-Raphson solver iterated each volatility

$\sigma^{(i)}$ until $\left|C_i^\star - \mathbf{PSN}_{n_i}^{N_i}(0; K_i)\right| < \epsilon$ for $\epsilon = 10^{-10}$. The resulting estimates are

$$\sigma = \begin{bmatrix} 0.3469759 \\ 0.3889807 \\ 0.4039994 \\ 0.4499999 \\ 0.4100003 \\ 0.3779617 \end{bmatrix},$$

which are close to the real implied volatilities provided.

With the instantaneous forward rate as in Figure 5.2, and the *solved* implied volatilities, a Monte Carlo simulation was performed in order to obtain the distribution of $r(t)$. With mean-reverting parameter $a = 0.5$, the result can be seen in Figure 5.4, where 5 simulated paths and their antithetic counterparts (giving 10 paths in total) are plotted.

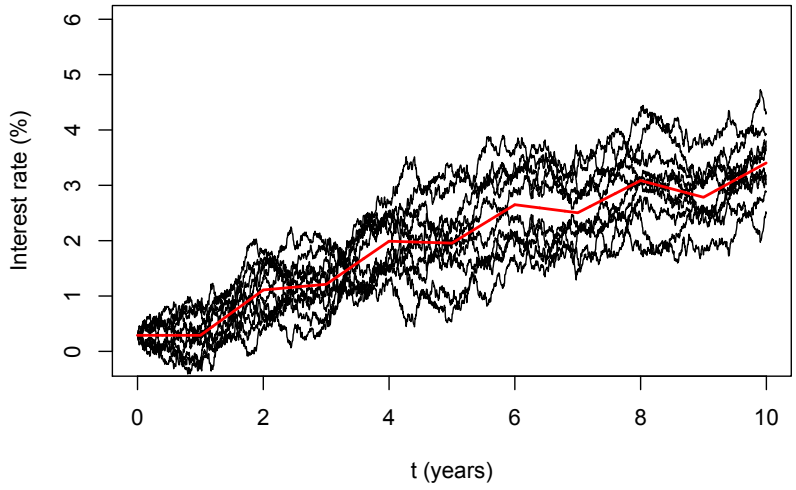

Figure 5.4: Graph of $f(0,T)$ in red, along with 10 Monte-Carlo paths for the short rate $r(t)$.

A version of Figure 5.3 together with integrated Monte Carlo paths,

$$\int_0^T r(u)du,$$

are plotted in Figure 5.5. Again, each path has an antithetic counterpart, making the distribution symmetric around the mean $f(0,T)$.

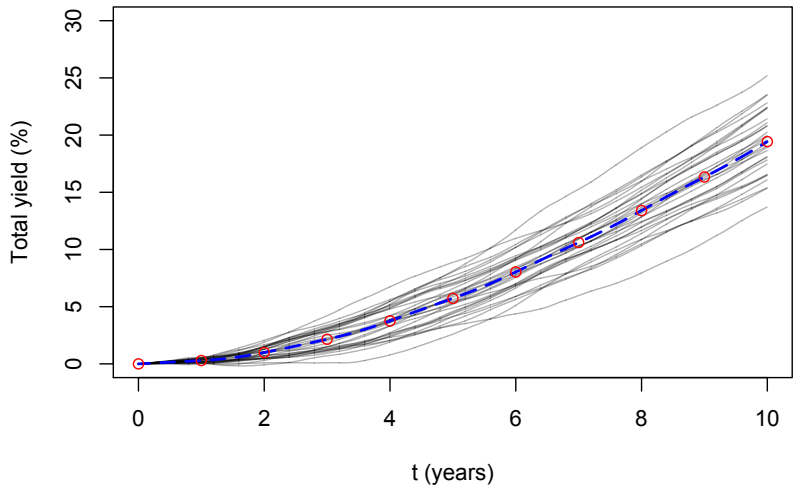

Figure 5.5: Graph of $\int_0^T f(0,u)du$ in red, along with 30 integrated Monte-Carlo paths for the short rate, $\int_0^T r(u)du$.

With the paths generated, the sensitivities of the short rate $r(t_k)$ at time t_k can be expressed in terms of the calibration parameters $r_i^\star$ and $C_i^\star$. Since the sensitivities of $\bar{r}^\star_{i,r_k}$ defined in (5.13) are independent of the random variables Z, the sensitivity will not be affected by the Monte Carlo simulation. For a random point t_k (in this case $k = 1337$) along the grid, the sensitivities of the short rate with respect to the calibration parameters $r_i^\star$ are

$$\begin{aligned}
\frac{\partial r_{1337}}{\partial r_0^\star} &= 0.936, & \frac{\partial r_{1337}}{\partial r_1^\star} &= -2.496,\\
\frac{\partial r_{1337}}{\partial r_2^\star} &= 3.744, & \frac{\partial r_{1337}}{\partial r_3^\star} &= -4.992,\\
\frac{\partial r_{1337}}{\partial r_4^\star} &= 6.240, & \frac{\partial r_{1337}}{\partial r_5^\star} &= 16.128,\\
\frac{\partial r_{1337}}{\partial r_6^\star} &= 4.816, & \frac{\partial r_{1337}}{\partial r_7^\star} &= 0,\\
\frac{\partial r_{1337}}{\partial r_8^\star} &= 0, & \frac{\partial r_{1337}}{\partial r_9^\star} &= 0.
\end{aligned}$$

This is due to the fact that $t_{1337} = 5.344$ in the discretized grid, hence $r_5^\star$ has the largest impact on the short rate at that point. These results can be replicated exactly using a finite difference method since $r(t)$ is linear with respect to $r_i^\star$, although via a linear function through both φ and $f(0,T)$.

The sensitivities of r_{1337} with respect to the volatilities σ, as defined in (5.14) are affected by the particular Monte Carlo path at hand. However, when using a antithetic sample, the random effects are canceled immediately, and $\bar{\sigma}^{(i)}_{r_k}$ converges towards $\bar{\sigma}^{(i)}_{\varphi_k} = \bar{\sigma}_\varphi[i,k]$. The resulting adjoints are therefore (even

with a sample of 2 antithetic paths)

$$\frac{\partial r_{1337}}{\partial \sigma^{(1)}} = 0.05018857, \qquad \frac{\partial r_{1337}}{\partial \sigma^{(2)}} = 0.07048655,$$
$$\frac{\partial r_{1337}}{\partial \sigma^{(3)}} = 0.49875497, \qquad \frac{\partial r_{1337}}{\partial \sigma^{(4)}} = 0.81266168,$$
$$\frac{\partial r_{1337}}{\partial \sigma^{(5)}} = 0.04095178, \qquad \frac{\partial r_{1337}}{\partial \sigma^{(6)}} = 0.$$

The maturities $\mathbf{T}^s$ imply that $\sigma^{(6)}$ is only applicable for $t_k > 8$, thus it has no effect on r_{1337}. These results are the analytical solutions to the partial derivatives, and the relationship between $r(t)$ and σ is not linear, therefore they are not *exactly* replicated by a finite difference approximation.

The change in a swaption price due to a change in implied volatility corresponds to moving the red line in Figure 5.6 up or down. Since $\mathbf{PSN}_n^N(0, R_n^N)$ is approximately linear in the region around $C_i^\star$, we approximate the derivative

$$\frac{d\sigma^{(i)}}{dC_i^\star} \approx \left(\frac{\partial C_i^\star}{\partial \sigma^{(i)}}\right) = \left(\frac{\partial \mathbf{PSN}_{n_i}^{N_i}(0; K_i)}{\partial \sigma^{(i)}}\right) = \frac{1}{\bar{\sigma}^{(i)}},$$

where $\bar{\sigma}^{(i)}$ is defined by (5.18). This is a very crude approximation indeed, but in the small region around $C_i^\star$ it is good enough due to the shape of the graph in Figure 5.6. With this approximation, the derivative $d\sigma^{(i)}/dC_i^\star$ is of the order 10^5. This might seem like a lot, but recall the small values of $C_i^\star$. Furthermore, if $C_i^\star$ increases by 10%, then σ_{imp} will only increase by 20%.

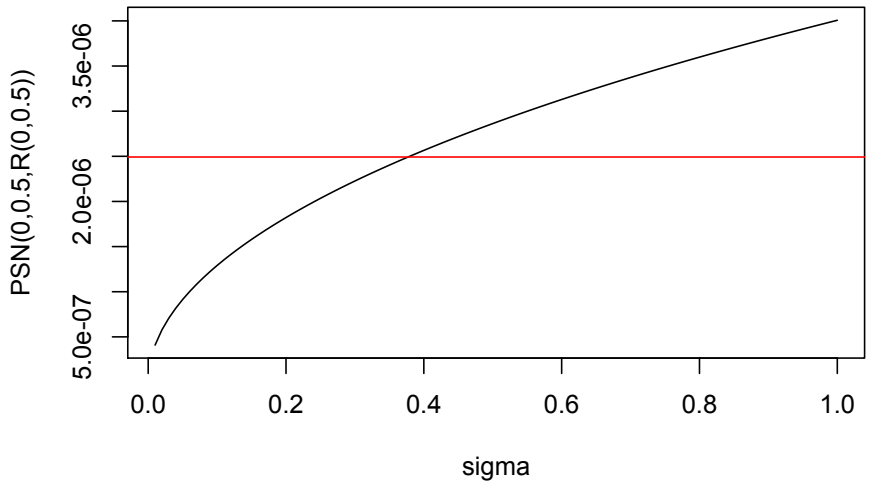

Figure 5.6: Different swaption prices $\mathbf{PSN}_n^N(0, R_n^N)$ for $n = 0$ and $N = 0.5$, for different volatilities $\sigma^{(1)}$. The red line corresponds to the price of the swaption using the implied volatility σ_{imp}.

Chapter 6

Conclusions

Chapter 2 introduced Adjoint Algorithmic Differentiation, and described the process assisted by examples. Section 2.5 confirmed the performance of AAD suggested by Griewank (see (11)) and Capriotti (see (5)), specifically that the runtime is in the region 3-5x the original runtime of the function. The results were compared to the forward Euler Finite Difference Method, which scales in runtime linearly with the number of inputs n.

In Chapter 3 AAD was implemented on a stochastic model, where the results could be compared to the analytical ones. By simulating the stochastic process using Monte Carlo with antithetic sampling, Section 3.2 could confirm that the numerical results using AAD converges toward the analytical result, thereby confirming that AAD works even in the stochastic setting.

In Chapter 4 the Hull-White model was introduced, and a method for fitting the model to an initial yield curve was presented. Then Section 4.3 introduced swaptions and described how their prices can be used to find the markets' implied volatility σ_{imp}. Finally all expressions were rewritten as a piecewise constant volatility version in Section 4.4.

Finally Chapter 5 combines Chapters 2-4, by applying Adjoint Algorithmic Differentiation in the calibration of the Hull-White model. In order to do that, the adjoint version of all input parameters are presented for the methods included in the calibration. The proposed instantaneous forward rate $f(0,T)$ is shown to have a nice fit to the initial yield curve generated by the market. Then the implied volatilities can be solved with very high accuracy using the Newton Raphson method. The sensitivities of the final short rate model with respect to the calibration parameters can then be computed without the need of a finite difference approximation. Furthermore, the sensitivities of the simulated short rate with respect to the calibration parameters could be computed without the need of a Monte Carlo simulation. Hence, the time consuming Monte Carlo simulation is only needed when the probability distribution of short rates $r(t)$ at t is desired.

Chapter 7

Bibliography

[1] Benhamou, E. Smart monte carlo: Various tricks using malliavin calculus. *Quantitative Finance 2*, 5 (January 2002), 329–336.

[2] Björk, T. *Arbitrage theory in continuous time.* Oxford university press, 2009.

[3] Brigo, D., and Mercurio, F. *Interest rate models-theory and practice: with smile, inflation and credit.* Springer Science & Business Media, 2007.

[4] Capriotti, L. Fast greeks by algorithmic differentiation. Avaliable at SSRN: https://ssrn.com/abstract=1619626.

[5] Capriotti, L., and Giles, M. Algorithmic differentiation: Adjoint greeks made easy. Avaliable at SSRN: https://ssrn.com/abstract=1801522.

[6] Cox, J. C., Ingersoll Jr, J. E., and Ross, S. A. A theory of the term structure of interest rates. *Econometrica: Journal of the Econometric Society* (1985), 385–407.

[7] Di Nunno, G., Øksendal, B., and Proske, F. *Malliavin Calculus for Lévy Processes with Applications to Finance*, 1 ed. Springer Berlin Heidelberg, 2009.

[8] Edsberg, L. *Introduction to computation and modeling for differential equations.* John Wiley & Sons, 2008.

[9] Glasserman, P. *Monte Carlo methods in financial engineering*, vol. 53. Springer Science & Business Media, 2013.

[10] Gomber, P., Arndt, B., Lutat, M., and Uhle, T. High-frequency trading. Available at SSRN: https://ssrn.com/abstract=1858626 .

[11] Greiwank, A., and Walther, A. *Evaluating Derivatives*. Society for Industrial and Applied Mathematics, 2008.

[12] Griewank, A., et al. On automatic differentiation. *Mathematical Programming: recent developments and applications 6*, 6 (1989), 83–107.

[13] Gurrieri, S., Nakabayashi, M., and Wong, T. Calibration methods of hull-white model. Available at SSRN: https://ssrn.com/abstract=1514192.

[14] Neumann, U. *The Art of Differentiating Computer Programs*. Society for Industrial and Applied Mathematics, 2012.

[15] Wengert, R. E. A simple automatic derivative evaluation program. *Communications of the ACM 7*, 8 (1964), 463–464.

[16] Wilmott, P. *Derivatives: The Theory and Practice of Financial Engineering*. Wiley, 1998.

Appendix A

Useful Proofs

Here are some Theorems, Propositions and Lemmas that are useful in the thesis.

Proposition A.1. ***Itô Process***
Suppose $\theta, \alpha \in \mathcal{L}^1$ and $\sigma \in \mathcal{L}^2$, and let r_0 be a random variable, measurable with respect to $\mathcal{F}_0$. Then there exists a unique Itô process r, such that

$$dr(t) = (\theta(t) + \alpha(t)r(t))\,dt + \sigma(t)dW(t), \quad r(0) = r_0. \tag{A.1}$$

It is given by

$$r(t) = e^{-\int_0^t \alpha(u)du}\left[r_0 + \int_0^t e^{\int_0^s \alpha(u)du}\theta(s)ds + \int_0^t e^{\int_0^s \alpha(u)du}\sigma(s)dW(s)\right],$$

or, given the filtration at $\mathcal{F}_\tau$

$$r(t) = e^{-\int_\tau^t \alpha(u)du}\left[r(\tau) + \int_\tau^t e^{\int_\tau^s \alpha(u)du}\theta(s)ds + \int_\tau^t e^{\int_\tau^s \alpha(u)du}\sigma(s)dW(s)\right].$$

Proof. Suppose $r(t)$ is a process that solves (A.1), we then have

$$\begin{aligned} d(e^{\int_s^t a d\nu} r(t)) =& e^{\int_s^t a d\nu} dr(t) + e^{\int_s^t a d\nu} r(t) a dt + e^{\int_s^t a d\nu} a \underbrace{dt dr(t)}_{=0} \\ =& e^{\int_s^t a d\nu}\left((\alpha(t) - ar(t))\,dt + \sigma(t)dW(t) + ar(t)dt\right) \\ =& e^{\int_s^t a d\nu}\left(\alpha(t)dt + \sigma(t)dW(t)\right). \end{aligned}$$

Integrating both sides from s to t yields

$$e^{\int_s^t a d\nu} r(t) - \underbrace{e^{\int_s^s a d\nu}}_{=1} r(s) = \int_s^t e^{\int_s^u a d\nu}\alpha(u)du + \int_s^t e^{\int_s^u a d\nu}\sigma(u)dW(u),$$

which implies

$$r(t) = e^{-\int_s^t a d\nu}\left(r(s) + \int_s^t e^{\int_s^u a d\nu}\alpha(u)du + \int_s^t e^{\int_s^u a d\nu}\sigma(u)dW(u)\right).$$

Furthermore, since the above process satisfies

$$e^{\int_s^t a d\nu}\left(\alpha(t)dt + \sigma(t)dW(t)\right) = d(e^{\int_s^t a d\nu} r(t)) = e^{\int_s^t a d\nu} dr(t) + e^{\int_s^t a d\nu} r(t) a dt,$$

we have

$$\alpha(t)dt + \sigma(t)dW(t) = dr(t) + r(t)adt,$$

which implies

$$dr(t) = (\alpha(t) - ar(t))\, dt + \sigma(t)dW(t),$$

and so the process $r(t)$ solves (A.1). □

Proposition A.2. ***Integration by parts***
If $X(t)$ and $Y(t)$ are one-dimensional Itô processes with

$$\begin{aligned} dX(t) &= a(t)dt + b(t)dW(t), \\ dY(t) &= \alpha(t)dt, \end{aligned}$$

then

$$\int_t^T X(s)dY(s) = X(T)Y(T) - X(t)Y(t) - \int_t^T Y(s)dX(s).$$

Proof. By Itô's lemma for a product. $X(t)Y(t)$ is an Itô process with

$$d(X(t)Y(t)) = X(t)dY(t) + Y(t)dX(t),$$

and integrating both sides of the above expression yields

$$X(T)Y(T) - X(t)Y(t) = \int_t^T d(X(s)Y(s)) = \int_t^T X(s)dY(s) + \int_t^T Y(s)dX(s).$$

□

Proposition A.3. ***Interchanging the order of integration***
Suppose $\alpha \in \mathcal{L}^1$ and $\beta \in \mathcal{L}^2$. Then:

$$\int_\tau^t \int_\tau^s \alpha(s)\beta(u)dW(u)ds = \int_\tau^t \int_u^t \alpha(s)\beta(u)dsdW(u). \tag{A.2}$$

Proof. Define process X and Y by:

$$\begin{aligned} dX(t) = \beta(t)dW(t), \qquad & X(\tau) = 0 \quad \Rightarrow X(t) = \int_\tau^t \beta(u)dW(u), \\ dY(t) = \alpha(t)dt, \qquad & Y(\tau) = 0 \quad \Rightarrow Y(t) = \int_\tau^t \alpha(s)ds. \end{aligned}$$

Then, by applying Proposition A.2 to (A.2) it becomes

$$\int_\tau^t \int_\tau^s \alpha(s)\beta(u)dW(u)ds = \int_\tau^t \alpha(s) \underbrace{\int_\tau^s b(u)dW(u)}_{=X(s)} ds = \int_\tau^t \alpha(s)X(s)ds$$

$$= \int_\tau^t X(s) \underbrace{\alpha(s)ds}_{=dY(s)} = \{\text{Proposition A.2}\}$$

$$= X(t)Y(t) - \underbrace{X(\tau)Y(\tau)}_{=0} - \int_\tau^t Y(s)dX(s)$$

$$= \int_\tau^t dX(u) \int_\tau^t dY(s) - \int_\tau^t \left(\int_\tau^s dY(u) \right) dX(s)$$

$$= \int_\tau^t \left(\int_\tau^t dY(s) \right) dX(u) - \int_\tau^t \left(\int_\tau^u dY(s) \right) dX(u)$$

$$= \int_\tau^t \left(\int_\tau^t dY(s) - \int_\tau^u dY(s) \right) dX(u)$$

$$= \int_\tau^t \left(\int_u^t dY(s) \right) dX(u) = \int_\tau^t \int_u^t \alpha(s)\beta(u)dsdW(u).$$

□

Definition A.4. Stochastic exponential
If $\alpha \in \mathcal{L}^1$ and $\beta \in \mathcal{L}^2$ are processes of dimension 1×1 and $1 \times K$, respectively, we define the one-dimensional process $\eta[\alpha, \beta]$ by

$$\eta[\alpha, \beta](t) = exp\left[\int_0^t \left(\alpha - \frac{1}{2}\beta\beta^T \right) ds + \int_0^t \beta dW \right].$$

A process of this form is called a *stochastic exponential*.

Proposition A.5. ***Positive Itô processes***
If X is a one-dimensional positive Itô process, then there exists $\alpha \in \mathcal{L}^1$ and $\beta \in \mathcal{L}^2$ such that

$$X(t) = X(0)\eta[\alpha, \beta](t),$$

where $\eta[\alpha, \beta](t)$ is defined as in Definition A.4.

Proof. Since X is positive, $ln(X(t))$ is a well-defined Itô process. Write its differential as

$$dln(x(t)) = \gamma dt + \beta dW.$$

Now set

$$\gamma = \alpha + \frac{1}{2}\beta\beta^T.$$

Then we get

$$ln(X(t)) = ln(X(0)) + \int_0^t \left(\alpha + \frac{1}{2}\beta\beta^T \right) ds + \int_0^t \beta dW.$$

And taking the exponential of this gives us

$$X(t) = X(0)\eta[\alpha, \beta](t).$$

□

Remark. An interesting follow up to Proposition A.5 is if $\eta[\alpha,\beta](t)$ is written using $Y(t) = ln(X(t))$

$$\eta[\alpha,\beta](t) = \frac{X(t)}{X(0)} = \frac{exp[ln(X(t))]}{X(0)} = \frac{e^{Y(t)}}{X(0)}. \tag{A.3}$$

By applying Itô's lemma on $f(x) = e^x$, noting that $\frac{\partial f(x)}{\partial x} = f(x)$ and $\frac{\partial f(x)}{\partial t} = 0$ it becomes,

$$\begin{aligned} df(Y(t)) &= \left(\frac{\partial f(Y)}{\partial t} + \gamma\frac{\partial f(Y)}{\partial Y} + \frac{1}{2}\beta\beta^T\frac{\partial^2 f(Y)}{\partial Y^2}\right)dt + \beta\frac{\partial f(Y)}{\partial Y}dW \\ &= \left(0 + \gamma e^{Y(t)} + \frac{1}{2}\beta\beta^T e^{Y(t)}\right)dt + \beta e^{Y(t)}dW \\ &= \left(\left(\alpha - \frac{1}{2}\beta\beta^T\right)X(t) + \frac{1}{2}\beta\beta^T X(t)\right)dt + \beta X(t)dW \\ &= X(t)(\alpha dt + \beta dW) = \{\text{Using Proposition A.5}\} \\ &= X(0)\eta[\alpha,\beta](t)(\alpha dt + \beta dW). \end{aligned} \tag{A.4}$$

Combining (A.3) and (A.4) yields

$$d(\eta[\alpha,\beta](t)) = d\left(\frac{f(Y(t))}{X(0)}\right) = \eta[\alpha,\beta](t)(\alpha dt + \beta dW). \tag{A.5}$$

Lemma A.6. ***Integral of Gaussian processes*** (Lemma 4.2.1 in (3))
Consider the following two processes:

$$\begin{aligned} dX(t) &= -aX(t)dt + \sigma(t)dW_1(t) \qquad X(0) = 0, \\ dY(t) &= -bY(t)dt + \eta(t)d_W 2(t) \qquad Y(0) = 0, \end{aligned}$$

where (W_1, W_2) is a two-dimensional Browninan motion with instantaneous correlation ρ as

$$dW_1(t)dW_2(t) = \rho dt,$$

and a, b are constants, and $\sigma(t), \eta(t)$ are deterministic functions. Then, for each t, T the random variable

$$I(t,T) = \int_t^T [X(u) + Y(u)]\,du, \tag{A.6}$$

conditional on the sigma field $\mathcal{F}_t$ is normally distributed with mean

$$M(t,T) = \frac{1 - e^{-a(T-t)}}{a}X(t) + \frac{1 - e^{-b(T-t)}}{b}Y(t), \tag{A.7}$$

and variance

$$\begin{aligned} V(t,T) = &\int_t^T \frac{\sigma(u)^2}{a^2}\left(1 - e^{-a(T-u)}\right)^2 du + \int_t^T \frac{\eta(u)^2}{b^2}\left(1 - e^{-b(T-u)}\right)^2 du \\ &+ \frac{2\rho}{ab}\int_t^T \sigma(u)\eta(u)\left(1 - e^{-a(T-u)}\right)\left(1 - e^{-b(T-u)}\right)du. \end{aligned} \tag{A.8}$$

Proof. By first splitting the integral in (A.6) into

$$I(t,T) = \int_t^T X(u)du + \int_t^T Y(u)du, \tag{A.9}$$

it is possible to work with each integral separately. By Proposition A.2 with $Z(t) = t$, such that $dZ(u) = du$, and $dX(t) = -aX(t)dt + \sigma(t)dW(t) = dx(t)$ the first integral in (A.9) can be written as

$$\begin{aligned}
\int_t^T X(u)du &= \int_t^T X(u)dZ(u) = X(T)Z(T) - X(t)Z(t) - \int_t^T Z(u)dX(u) \\
&= TX(T) - tX(t) - \int_t^T udX(u) \\
&= (TX(t) - TX(t)) + TX(T) - tX(t) - \int_t^T udX(u) \\
&= (T-t)X(t) + TX(T) - TX(t) - \int_t^T udX(u) \\
&= (T-t)X(t) + \int_t^T TdX(u) - \int_t^T udX(u) \\
&= (T-t)X(t) + \int_t^T (T-u)dX(u).
\end{aligned} \tag{A.10}$$

From the definition of $dX(t)$, the integral on the right-hand-side can be written as

$$\int_t^T (T-u)dX(u) = -a\int_t^T (T-u)X(u)du + \int_t^T (T-u)\sigma(u)dW_1(u). \tag{A.11}$$

By applying Proposition A.1 to $X(u)$, we can rewrite the first integral of the right-hand-side in (A.11) as

$$\begin{aligned}
&-a\int_t^T (T-u)X(u)du \\
=&-a\int_t^T (T-u)\left(X(t)e^{-a(u-t)} + \int_t^u e^{-a(u-s)}\sigma(s)dW_1(s)\right)du \\
=&-aX(t)\int_t^T (T-u)e^{-a(u-t)}du - a\int_t^T (T-u)\int_t^u e^{-a(u-s)}\sigma(s)dW_1(s)du.
\end{aligned} \tag{A.12}$$

These two integral can be calculated separately, the first one being:

$$\begin{aligned}
&-aX(t)\int_t^T (T-u)e^{-a(u-t)}du \\
=&-aX(t)\left(\underbrace{\left[(T-u)\frac{e^{-a(u-t)}}{-a}\right]_{u=t}^{u=T}}_{=(T-t)/a}+\int_t^T \frac{e^{-a(u-t)}}{-a}du\right) \\
=&-aX(t)\left(\frac{(T-t)}{a}+\left[\frac{e^{-a(u-t)}}{a^2}\right]_{u=t}^{u=T}\right) \\
=&X(t)\left(-(T-t)-\frac{e^{-a(T-t)}-1}{a}\right).
\end{aligned} \tag{A.13}$$

And the second integral in (A.12) becomes

$$\begin{aligned}
&-a\int_t^T (T-u)\int_t^u e^{-a(u-s)}\sigma(s)dW_1(s)du \\
=&-a\int_t^T\int_t^u (T-u)e^{-a(u-s)}\sigma(s)dW_1(s)du \\
=&\{\text{Using Proposition A.3 }\} \\
=&\int_t^T \underbrace{-a\int_s^T (T-u)e^{-a(u-s)}du}_{\text{Use Equation (A.13)}}\sigma(s)dW_1(s) \\
=&\int_t^T \left(-(T-s)-\frac{e^{-a(T-s)}-1}{a}\right)\sigma(s)dW_1(s) \\
=&-\int_t^T \sigma(s)\left((T-s)+\frac{e^{-a(T-s)}-1}{a}\right)dW_1(s).
\end{aligned} \tag{A.14}$$

Putting (A.13), and (A.14) back into (A.12) gives

$$\begin{aligned}
&-a\int_t^T (T-u)X(u)du \\
=&X(t)\left(-(T-t)-\frac{e^{-a(T-t)}-1}{a}\right)-\int_t^T \sigma(s)\left((T-s)+\frac{e^{-a(T-s)}-1}{a}\right)dW_1(s).
\end{aligned}$$

Which can be inserted back into (A.11) to get

$$\begin{aligned}
\int_t^T (T-u)dX(u) =&-X(t)(T-t)-\frac{e^{-a(T-t)}-1}{a}X(t)-\int_t^T \sigma(u)(T-u)dW_i(u) \\
&-\int_t^T \sigma(u)\frac{e^{-a(T-u)}-1}{a}dW_1(u)+\int_t^T \sigma(u)(T-u)dW_i(u) \\
=&-X(t)(T-t)-\frac{e^{-a(T-t)}-1}{a}X(t) \\
&-\int_t^T \sigma(u)\frac{e^{-a(T-u)}-1}{a}dW_1(u).
\end{aligned}$$

Which, in turn implies that (A.10) becomes

$$\begin{aligned}\int_t^T X(u)du =&(T-t)X(t) - X(t)(T-t) - \frac{e^{-a(T-t)}-1}{a}X(t) \\ &- \int_t^T \sigma(u)\frac{e^{-a(T-u)}-1}{a}dW_1(u) \\ =&\frac{1-e^{-a(T-t)}}{a}X(t) + \int_t^T \frac{\sigma(u)}{a}\left(1-e^{-a(T-u)}\right)dW_1(u).\end{aligned} \tag{A.15}$$

The steps above is similar for $Y(t)$ in Equation (A.9), giving the analogous expression for $Y(t)$ as

$$\int_t^T Y(u)du = \frac{1-e^{-b(T-t)}}{b}Y(t) + \int_t^T \frac{\eta(u)}{b}\left(1-e^{-b(T-u)}\right)dW_2(u). \tag{A.16}$$

Hence

$$\begin{aligned}\mathbb{E}\left[I(t,T)|\mathcal{F}_t\right] &= \mathbb{E}\left[\int_t^T X(u)du + \int_t^T Y(u)du\middle|\mathcal{F}_t\right] \\ &= \frac{1-e^{-a(T-t)}}{a}X(t) + \frac{1-e^{-b(T-t)}}{b}Y(t) \\ &= M(t,T),\end{aligned} \tag{A.17}$$

which proves (A.7).

As for the conditional variance in (A.6) the definition of variance yields

$$\begin{aligned}Var\left\{I(t,T)|\mathcal{F}_t\right\} = \mathbb{E}&\left[\left(\int_t^T x(u)du + \int_t^T y(u)du\right)^2\middle|\mathcal{F}_t\right] \\ &- \mathbb{E}\left[\int_t^T x(u)du + \int_t^T y(u)du\middle|\mathcal{F}_t\right]^2,\end{aligned} \tag{A.18}$$

where it follows from the calculations earlier that the second expectation is equal to Equation (A.7) squared, specifically

$$\begin{aligned}M(t,T)^2 = &\left(\frac{1-e^{-a(T-t)}}{a}x(t)\right)^2 + \left(\frac{1-e^{-b(T-t)}}{b}y(t)\right)^2 \\ &+ 2\left(\frac{1-e^{-a(T-t)}}{a}x(t)\right)\left(\frac{1-e^{-b(T-t)}}{b}y(t)\right).\end{aligned} \tag{A.19}$$

The first expectation needs a bit more work however,

$$\mathbb{E}\left[\left(\int_t^T x(u)du + \int_t^T y(u)du\right)^2\middle|\mathcal{F}_t\right] = \text{ Inserting (A.15) and (A.16)}$$

$$= \mathbb{E}\left[\left(\frac{1-e^{-a(T-t)}}{a}X(t) + \int_t^T \frac{\sigma(u)}{a}\left(1-e^{-a(T-u)}\right)dW_1(u)\right.\right.$$
$$\left.\left. + \frac{1-e^{-b(T-t)}}{b}Y(t) + \int_t^T \frac{\eta(u)}{b}\left(1-e^{-b(T-u)}\right)dW_2(u)\right)^2 \middle| \mathcal{F}_t\right].$$

For a less cumbersome notation, let

$$A(t) = \frac{1-e^{-a(T-t)}}{a}X(t) \qquad \alpha(t) = \int_t^T \frac{\sigma(u)}{a}\left(1-e^{-a(T-u)}\right)dW_1(u),$$
$$B(t) = \frac{1-e^{-b(T-t)}}{b}Y(t) \qquad \beta(t) = \int_t^T \frac{\eta(u)}{b}\left(1-e^{-b(T-u)}\right)dW_2(u).$$

Which gives

$$\begin{aligned}
&\mathbb{E}\left[A^2(t) + 2A(t)\alpha(t) + 2A(t)B(t) + 2A(t)\beta(t) + \alpha(t)^2\right.\\
&\quad \left. +2\alpha(t)B(t) + 2\alpha(t)\beta(t) + B(t)^2 + 2B(t)\beta(t) + \beta(t)^2 \middle| \mathcal{F}_t\right]\\
=&\mathbb{E}\left[A^2(t)\middle|\mathcal{F}_t\right] + 2\mathbb{E}\left[A(t)B(t)|\mathcal{F}_t\right] + \mathbb{E}\left[B^2(t)\middle|\mathcal{F}_t\right]\\
&+ \mathbb{E}\left[2A(t)\alpha(t) + 2A(t)\beta(t) + 2\alpha(t)B(t) + 2B(t)\beta(t)|\mathcal{F}_t\right]\\
&+ \mathbb{E}\left[\alpha(t)^2 + 2\alpha(t)\beta(t) + \beta(t)^2\middle|\mathcal{F}_t\right].
\end{aligned}$$

The first three expectations are deterministic, and equal to $M(t,T)^2$ seen in (A.19). The fourth expectation vanishes since

$$\begin{aligned}
&\mathbb{E}\left[2A(t)\alpha(t) + 2A(t)\beta(t) + 2\alpha(t)B(t) + 2B(t)\beta(t)|\mathcal{F}_t\right]\\
=&\mathbb{E}\left[2\alpha(t)\left(A(t)+B(t)\right) + 2\beta(t)\left(A(t)+B(t)\right)|\mathcal{F}_t\right]\\
=&2\underbrace{\mathbb{E}\left[\alpha(t)|\mathcal{F}_t\right]}_{=0}\mathbb{E}\left[A(t)+B(t)|\mathcal{F}_t\right] + 2\underbrace{\mathbb{E}\left[\beta(t)|\mathcal{F}_t\right]}_{=0}\mathbb{E}\left[\left(A(t)+B(t)\right)|\mathcal{F}_t\right]\\
=&\,0.
\end{aligned}$$

Thus (A.18) becomes

$$\begin{aligned}
Var\left\{I(t,T)|\mathcal{F}_t\right\} &= M(t,T)^2 + \mathbb{E}\left[\alpha(t)^2 + 2\alpha(t)\beta(t) + \beta(t)^2\middle|\mathcal{F}_t\right] - M(t,T)^2\\
&= \mathbb{E}\left[\alpha(t)^2 + 2\alpha(t)\beta(t) + \beta(t)^2\middle|\mathcal{F}_t\right]\\
&= \mathbb{E}\left[\alpha(t)^2\middle|\mathcal{F}_t\right] + 2\mathbb{E}\left[\alpha(t)\beta(t)|\mathcal{F}_t\right] + \mathbb{E}\left[\beta(t)^2\middle|\mathcal{F}_t\right].
\end{aligned}$$

Returning to the original notation, the expectations become

$$\mathbb{E}\left[\left(\int_t^T \frac{\sigma(u)}{a}(1-e^{-a(T-u)}dW_1(u)\right)^2\middle|\mathcal{F}_t\right] = \frac{1}{a^2}\int_t^T \sigma(u)^2\left(1-e^{-a(T-u)}\right)^2 du,$$

$$\mathbb{E}\left[\left(\int_t^T \frac{\eta(u)}{b}(1-e^{-b(T-u)}dW_2(u)\right)^2\middle|\mathcal{F}_t\right] = \frac{1}{b^2}\int_t^T \eta(u)^2\left(1-e^{-b(T-u)}\right)^2 du,$$

$$2\mathbb{E}\left[\int_t^T \frac{\sigma(u)}{a}(1-e^{-a(T-u)}dW_1(u)\int_t^T \frac{\eta(u)}{b}(1-e^{-b(T-u)}dW_2(u)\middle|\mathcal{F}_t\right]$$
$$= \frac{2\rho}{ab}\int_t^T \sigma(u)\eta(u)\left(1-e^{-a(T-u)}\right)\left(1-e^{-b(T-u)}\right)du,$$

which prove (A.8). □

TRITA -MAT-E 2017:57
ISRN -KTH/MAT/E--17/57--SE