Università
Ca'Foscari
Venezia

Master's Degree programme – Second Cycle (*D.M. 270/2004*)

in Economics and Finance

Final Thesis

The Laplace Transform in option pricing

–

Ca' Foscari
Dorsoduro 3246
30123 Venezia

Supervisor

Ch. Prof. Antonella Basso

Graduand

Roberta Ruggeri

847664

Academic Year

2014 / 2015

INDEX

INTRODUCTION **5**

CHAPTER 1 .... **7**

1.1 General Probability Theory **7**

1.2 Stochastic Calculus **10**

1.2.1 Stochastic or Itô Integrals *10*

1.2.2 Itô Formula *11*

1.3 Stochastic Differential Equation **14**

1.3.1 Linear Stochastic Differential Equation (SDE) *15*

1.3.2 Geometric Brownian Motion *15*

1.3.3 Itô Operator *17*

1.4 The Black-Scholes-Merton Model **17**

1.4.1 The BSM equation *18*

1.4.2 Considerations *20*

CHAPTER 2 .... **23**

2.1 Basic Principles **23**

2.2 Convergence **26**

2.3 Basic Properties **28**

2.4 Laplace Transform Calculus **30**

2.5 The Inverse Laplace Transform **34**

2.6 Numerical Inversion of the Laplace Transform **39**

2.6.1 The Gaver-Stehfest Algorithm 41

2.6.2 The Euler Algorithm 43

2.6.3 The Talbot algorithm 45

CHAPTER 3 .... **47**

3.1 Solving a PDE using the Laplace Transform **48**

3.1.1 Second Order Linear PDE's 48

3.1.2 The Heat Equation 49

3.1.3 The Black-Scholes-Merton equation 53

3.2 Option Pricing ... **55**
3.2.1 European option ... 55
3.2.2 Double Barrier Options ... 57
3.2.3 Lookback Options ... 69
3.2.4 Asian Option ... 72
CHAPTER 4 ... **74**
4.1 A comparison among Gaver-Stefhest, Euler and Talbot inversion algorithm ... **77**
4.1.1 Inversion of $F_{01}(s) = s\log(s)$ ... 79
4.1.2 Inversion of $F_{02}(s) = \log\left(1+\frac{1}{s}\right)$ ... 82
4.1.3 Inversion of $F_{03}(s) = \frac{e^{-\frac{1}{4s}}}{\sqrt{s^3}}$ ... 85
4.1.4 Inversion of $F_{04}(s) = \arctan\left(\frac{1}{s}\right)$ ... 87
4.1.5 Inversion of $F_{05}(s) = -\log(s^2+1)$ ... 91
4.1.7 Considerations ... 94
4.2 Lookback Options: results ... **95**
4.2.1 Lookback options: comparison among different pricing methods ... 96
4.2.2 Numerical inversion for different values of σ ... 99
CHAPTER 5 ... **102**
5.1 The core of the problem ... **103**
5.2 Bessel process ... **105**
5.3 The law of $A_t^{(v)}$ at an independent exponential time ... **107**
5.4 Price of an arithmetic Asian option ... **109**
5.5 Numerical inversion of the Laplace transform of the arithmetic Asian option price ... **112**
5.5.1 The Euler algorithm proposed by Abate and Whitt (1995) ... 113
5.5.2 Asian option: numerical analysis ... 114
CONCLUSIONS ... **120**
APPENDIX A ... **123**
A.1 Probability theory: functions and distributions ... **123**
A.2 Proof of $\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right) = \frac{e^{-\frac{a^2}{4t}}}{\sqrt{\pi t}}$ ... **124**
A.3 Cauchy's Theorem ... **125**

A.4 Cauchy's Integral Formula **126**

A.5 Residues Theorem **126**

APPENDIX B **129**

B.1 Second Linear Differential Equation **129**

B.2 Initial-value and Boundary-value problem **131**

APPENDIX C **134**

C.1 The Arbitrage Pricing Theory under the Martingale Approach **134**

C.2 The BSM under the martingale approach **140**

MATLAB CODES **143**

D.1 Inversion algorithms **143**

Gaver-Stefhest inversion algorithm 143

Euler inversion algorithm 144

Talbot inversion algorithm 145

Comparison 1 146

Comparison 2 147

D.2 Floating lookback options **148**

Cox-Rubistein-Ross binomial tree 148

Monte Carlo simulation 149

Laplace Transform approach 149

Closed-form formula 150

D.3 Arithmetic Asian call options **151**

Monte Carlo Simulation 151

Laplace Transform approach 152

Complex functions 154

BIBLIOGRAPHY **156**

INTRODUCTION

The aim of this thesis is the study of the application of the Laplace Transform to option pricing.
The thesis is inspired by recent papers from which it takes the idea to study the application of a mathematical tool such as the Laplace Transform to a financial evaluation problem such as the pricing of the options.
The Laplace Transform is a mathematical tool used to transform differential problems into algebraic ones of simplest resolution. In our case, we apply the transform to the well-known Black-Scholes-Merton partial differential equation whose result is the price of an option. The solution of this differential equation can be obtained inverting the Laplace Transform. Unfortunately, for some of the options studied, such as Asian options, it is difficult to find an analytical expression for the inverse Laplace Transform. Hence the necessity of the numerical inversion. We have chosen three different inversion algorithms (Gaver-Stefhest, Euler and Talbot), which have been implemented in MATLAB. In addition, we have compared the new approach, based on the Laplace Transform with other two financial methods, the Monte Carlo simulation and the Cox-Ross-Rubistein model.
In the end, looking at our results and the literature we investigate if the Laplace Transform approach is an efficient and valid pricing instrument.

In the first chapter, we start presenting the Black-Scholes-Merton model and its assumptions. The most important result of this first section is the partial differential equation that can be used to price every type of option under the hypothesis of the Black-Scholes-Merton model.

In Chapter 2, we present the theory of the Laplace Transform as it allows to pass from a differential problem to an algebraic one, which is an easier task to solve. Then, as far as the numerical inversion of the Laplace Transform concerns, we begin presenting the analytical method, based on the Residue Theorem, that provides an exact solution but that is really hard to be applied. For this reason, we discuss three methods of numerical inversion: the Gaver-Stefhest algorithm, the Euler algorithm and the Talbot algorithm.

In Chapter 3 we apply the Laplace Transform to the partial differential equation of European, Barrier and Lookback options. For the European and Barrier options it is possible to invert the Laplace Transform exploiting the analytical methods and reaching an analytical formula for the price of both. Instead, for Lookback options we have to use the numerical inversion algorithms to invert the transform.

In Chapter 4 we analyze the results obtained by developing MATLAB codes for the three numerical inversion algorithms and for the computation of the price of a lookback option using the Laplace Transform approach, Monte Carlo simulation and Cox-Rubistein binomial tree.

In Chapter 5 we present the arithmetic Asian option pricing problem. We begin describing the theory and the assumptions needed to solve the pricing problem using the Laplace Transform. Then we invert the found transform using the Euler inversion algorithm and we go on making numerical tests of the presented model. Finally, we compare the price given by the numerical inversion with the price computed using Monte Carlo simulation and the Cox-Rubistein binomial tree.

CHAPTER 1

OPTION PRICING IN CONTINUOS TIME

1.1 General Probability Theory

The final objective is the presentation of methods for resolving the partial differential equation found by Black, Scholes and Merton to price options. In order to simplify the resolution of the well-known equation it will be used the Laplace Transform, a mathematical tool widely applied in physics to solve problems involving partial differential equations (i.e heat equation).

Before entering into the heart of the problem, it is interesting to analyze the procedure that leads to the Black-Scholes-Merton formula.

For doing that it is fundamental to have some knowledge about stochastic calculus, as the equation is computed under the assumption of an economic continuous time model.

However, the stochastic calculus itself is based on several principles of probability theory which will be explained in this section.

Definition 1.1: σ-algebra

Let Ω be a nonempty set, and let $\mathcal{F}$ be a collection of subset of Ω.

We can say that $\mathcal{F}$ is a σ-algebra if:

I. The empty set, $\emptyset$, belongs to $\mathcal{F}$,

II. Whatever set $A \epsilon \mathcal{U},$ also its complement $A^C \epsilon \mathcal{F}$

III. Whenever a sequence of sets $A_1, A_2, \ldots$ belongs to $\mathcal{F}$, their union -_$\bigcup_{n=1}^{\infty} A_n$ also belongs to $\mathcal{F}$.

Definition 1.2: Probability Space

Let Ω be a non empty set, and $\mathcal{F}$ be a σ-algebra of subsets of Ω. A probability measure $\mathbb{P}$ is a function that, to every set $A \epsilon \mathcal{F}$, assigns a number in [0,1], called $\mathbb{P}(A)$. In addition, it is required that:

I. $\mathbb{P}(\Omega) = 1$

II. Whenever $A_1, A_2, \ldots$ is a sequence of disjoint sets in $\mathcal{F}$, then

$$\mathbb{P}\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum_{n=1}^{\infty} \mathbb{P}(A_n)$$

Finally, the triple $(\Omega, \mathcal{F}, \mathbb{P})$ is defined as a probability space.

Definition 1.3: Borel σ-algebra

The σ-algebra obtained by beginning with closed intervals and adding everything else necessary on order to have a σ-algebra:

$$[a, b] = \bigcup_{n=1}^{\infty} \left[a + \frac{1}{n}, b + \frac{1}{n}\right]$$

is called Borel σ-algebra of subset of $[0,1] = \mathcal{B}[0,1]$.

Definition 1.4: Random Variable

If $X: \Omega \to \mathbb{R}$ and $(\Omega, \mathcal{F}, \mathbb{P})$ is the probability space and $(\mathbb{R}, \mathcal{B})$ is a Borel σ-algebra, then X is a random variable if for every $B \epsilon\, \mathcal{B}$, $\mathcal{F}$ such that X is $\mathcal{F}$-measurable.

Definition 1.5: $\mathcal{F}$-measurable

Let X be a random variable defined on a non-empty space Ω and $\mathcal{F}$ a σ-algebra of subset of Ω. If every set in $\sigma(X)$ is also in $\mathcal{F}$, it is said that X is $\mathcal{F}$-measurable.

Definition 1.6: A "Finer" σ-algebra

If $\mathcal{U}_1, \mathcal{U}_2$ are σ-algebras on the same nonempty set Ω, then $\mathcal{U}_1$ is defined to be finer than $\mathcal{U}_2$ when $\mathcal{U}_1 \supseteq \mathcal{U}_2$.

Definition 1.7: Filtration

A filtration $\{\mathcal{F}(t)\}_{t\geq 0}$ on the probability space $(\Omega, \mathcal{F}, \mathbb{P})$ is an indexed family of σ-algebras on Ω such that

$$\forall s \leq t : \ \mathcal{F}(s) \leq \mathcal{F}(t) \ or \ A\epsilon\ \mathcal{F}(s) \Rightarrow A\epsilon\mathcal{F}(t)$$

Definition 1.8: Adapted process

A stochastic process is adapted to a filtration $\{\mathcal{F}(t)\}_{t\geq 0}$ on some probability space, and a random process X on the same space. The process X is adapted to the filtration $\{\mathcal{F}(t)\}_{t\geq 0}$ if $\forall t \geq 0 : X_t \epsilon \mathcal{F}_t$.

Definition 1.9: Stochastic Process

A stochastic process is a collection of random variables $\{X(t)\}_{t\geq 0}$ parameterized by time and defined on the same probability space $(\Omega, \mathcal{F}, \mathbb{P})$.

Definition 1.10: Wiener Process or Standard Brownian Motion

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space. If $\forall \omega \epsilon \Omega$ there exist a continuous function $W(t)$ which:

I. at 0, $W(0) = 0$
II. depends only on ω

Then $W(t)$ is a Brownian Motion if $\forall\ 0 = t_0 < t_1 < \cdots < t_m$ its increments

$$W(t_1) = W(t_1) - W(t_0), W(t_2) - W(t_1), \dots , W(t_m) - W(t_{m-1})$$

are:

III. independent
IV. normally distributed with mean 0 and variance $t_{i+1} - t_i$.

1.2 Stochastic Calculus

Stochastic calculus defines a method to calculate integral of processes that changes continuously in time such as stochastic processes.

Initially, it was widely used in physics to study diffusion processes which follows a Brownian Motion but in the last 30 years it has been applied also in financial mathematics and economics to model the evolution in time of stock prices which are usually assumed to follow a Brownian Motion.

In this chapter, it is presented a complete introduction to stochastic calculus but without any attempt at maximal detail. By the way it will be enough to reach our goal: the building of the Black-Scholes-Merton equation.

1.2.1 Stochastic or Itô Integrals

First of all, it is provided a definition for Itô integrals

$$\int_0^t \Delta(s) dW(s) \text{ where } \Delta(s) \epsilon £^2[0,t]$$

Secondly, to guarantee the existence of these integrals it has been imposed the following integrability conditions:

I. The process $\Delta(s)$ belongs to the class $£^2[a,b]$ which means that:
   - $\int_a^b E[\Delta^2(s)]ds < \infty$
   - The process $\Delta(s)$ is adapted to the $\mathcal{F}_t^W$-filtration[1].

II. The process $\Delta(s)$ holds to $£^2$ if $g \epsilon £^2[0,t]$, for all $t > 0$.

Unfortunately stochastic integrals cannot be solved using the ordinary procedure because the Brownian motion paths are not differentiable with respect to time.

The method to find out the solution of the integrals is divided in three steps:

I. <u>Define the integral for a constant process.</u>

$$\Delta(s) = \bar{\Delta}$$

$$\Rightarrow \int_0^t \bar{\Delta}\, dW(s) = \bar{\Delta} \int_0^t dW(s) = \bar{\Delta} \cdot [W(t) - W(0)]$$

[1] $\mathcal{F}_t^W$ is the σ-algebra generated by the random variable W over the interval $[0,t]$

II. Define the integral for a simple process.

$$\Delta(s) = \sum_{j=0}^{n-1} \Delta(t_j) \cdot \mathcal{X}_{(t_j, t_{j+1})}(t)$$ [2]

$$\Rightarrow \int_0^t \Delta(s) dW(s) = \sum_{j=0}^{n-1} \Delta(t_j)[W(t_{j+1}) - W(t_j)]$$

III. Define the integral for a general process.

In this circumstances $\Delta(s)$ is a stochastic process that can jump and is continuously varying. The evaluation of the integrand is build dividing the time interval $[0, t]$ in n parts, then for each of these sub interval, indicated with $[t_j, t_{j+1})$ for $j = 1, \dots n-1$, $\Delta(s)$ is approximated with a simple process $\Delta(t_j)$. Higher is n better is the estimation, so for $n \longrightarrow \infty$ the sequence of $\Delta_n(s)$ converges to $\Delta(s)$. But for every $\Delta_n(s)$ the integral has been defined in the previous point so it is immediate to furnish the following formula:

$$\boldsymbol{I} = \int_{\boldsymbol{0}}^{\boldsymbol{t}} \Delta(\boldsymbol{s}) \boldsymbol{dW}(\boldsymbol{s}) = \lim_{\boldsymbol{n} \longrightarrow \infty} \int_{\boldsymbol{o}}^{\boldsymbol{t}} \Delta_{\boldsymbol{n}}(\boldsymbol{s}) \boldsymbol{dW}(\boldsymbol{t})$$

1.2.2 Itô Formula

The Itô formula is an identity used in stochastic calculus to find the differential of a time dependent function of a stochastic process.

Firstly, it should be considered the Taylor's expansion formula for a k-times differentiable $f(x)$, $f: \mathbb{R} \rightarrow \mathbb{R}$:

$$f(x+h) \cong f(x) + f'(x)h + \frac{1}{2} f''(x)h^2 + \cdots + \frac{f^{(k)}(x)}{k!} h^k$$

If h is equal to dx and $f(x)$ is subtract from both parts:

$$f(x+dx) - f(x) \cong f'(x)dx + \frac{1}{2} f''(x)(dx)^2 + \cdots + \frac{f^{(k)}(x)}{k!} (dx)^k$$

---

[2] $\mathcal{X}_{(t_j, t_{j+1})}(t) = \begin{cases} 1 & if\ x \epsilon (t_j, t_{j+1}) \\ 0 & otherwise \end{cases}$

But it can be seen that the first part of the equation is nothing more than the usual definition of $df(x)$.

For linear equation a linear approximation is sufficient on the other hand, stochastic differential equations require at least the second order for a good approximation:

$$df(x) = f'(x)dx + \frac{1}{2}f''(x)(dx)^2$$

Now we are able to find the differential of $f(W(t))$, where $f(\cdot)$ is a differentiable function and $W(t)$ is a Brownian motion, which has a continuous trajectory but it is not differentiable in every point. The absence of smoothness makes the quadratic variation of the process different from zero[3].

$$df\big(t, W(t)\big) = f'\big(W(t)\big)dW(t) + \frac{1}{2}f''\big(W(t)\big)\big(dW(t)\big)^2$$

$$but\ \big(dW(t)\big)^2 = dt$$

$$\Rightarrow df\big(t, W(t)\big) = f'\big(W(t)\big)dW(t) + \frac{1}{2}f''\big(W(t)\big)dt$$

This is the Itô formula for a Wiener process in differential form but we are interested in the integrated representation so:

$$\boldsymbol{f\big(W(t) - W(0)\big) = \int_o^t f'\big(W(s)\big)dW(s) + \frac{1}{2}\int_0^t f''(W(s))ds}$$

In the next step, we want to define the Ito formula for Itô processes, which are more general processes then the Brownian Motion and take in consideration almost all stochastic processes with exception of the ones with jumps.

Definition 1.11: Itô process

A stochastic process $X(t)$ is named Itô process if its stochastic differential is:

$$dX(t) = \Delta(t)dW(t) + \Theta(t)dt$$

Where $\Delta(t)$ and $\Theta(t)$ are stochastic process adapted to a $\mathcal{F}_t^X$-filtration.

---

[3] Defining the quadratic variation of a Brownian Motion S_n it is possible to demonstrate that S_n tends to the deterministic limit t.

Definition 1.12: Itô Formula for Itô processes

Now, assume that the process $X(t)$ is an Itô process and let f be a $C^{1,2}$-function.

Define the process Z by $Z(t) = f(t, X(t))$, then Z has a stochastic differential given by:

$$^{4}\boldsymbol{df\big(t, X(t)\big) = \left\{f_t\big(t, X(t)\big) + f_x\big(t, X(t)\big)\Theta(t) + \frac{1}{2} f_{xx}(t, W(t))\Delta^2(t)\right\} dt + \Delta(t) f_x\big(t, X(t)\big) dW(t)}$$

Finally, it is useful to define also the Itô formula for a multidimensional process. Assuming that:

- $X = (X_1, \dots, X_n)^T$ where $dX_i(t) = \Delta_i(t)dt + \sum_{j=1}^{n} \Theta_{\mathrm{ij}}(t) dW_j(t)$
- $W_1, \dots, W_d$ are independent Brownian Motions.
- $\Delta_i(t)$ is the i-th element of the n column vector $\Delta(t)$
- $\Theta_{\mathrm{ij}}(t)$ is the ij-th element of the $n \times d$ matrix $\Theta(t)$, called diffusion matrix.
- $Z(t) = f(t, X(t))$, where $f: R_+ \times R^n \to R$ holds to the class $C^{1,2}$.

Then the stochastic differential is defined as (the term $\big(t, X(t)\big)$ is omitted to simplify the formula):

$$df = \left\{ f_t + \sum_{i=1}^{n} f_{x_i} \Delta_i(t) + \frac{1}{2} \sum_{i,j=1}^{n} C_{ij} f_{x_i x_j} \right\} dt + \sum_{i=1}^{n} f_{x_i} \Theta_{\mathrm{i}}(t) dW(t)$$

Where:

- $\Theta_{\mathrm{i}}(t)$ is the i-th row of the matrix $\Theta(t)$
- $\mathrm{C} = \Theta(t)\Theta(t)^{\mathrm{T}}$

[4] A full formal proof is given in SHREVE S.E. (2004), *Stochastic Calculus for Finance II. Continuous Time - Models,* United States, Springer. ISBN 0-387-40101-6

1.3 Stochastic Differential Equation

Stochastic differential equations (SDE) are differential equation composed by one or more stochastic process, and they results in a solution which is itself a stochastic process.

The general form of a SDE is:

$$\begin{cases} dX(t) = \mu\big(t, X(t)\big)dt + \sigma\big(t, X(t)\big)dW(t) \\ X_0 = x_0 \end{cases}$$ [5]

The question is whether there exists a stochastic process $X(t)$ which satisfies the SDE, or in other words if it is possible to find a process $X(t)$ satisfying the integral equation:

$$X(t) = x_0 + \int_o^t \mu\big(s, X(s)\big)ds \int_0^t \sigma\big(s, X(s)\big)dW(s) \quad \forall t \geq 0$$

The answer is given by the below proposition**.**

Proposition 1.1: Uniqueness of SDE's solution

Assume $\exists k \geq 0 \ \wedge \forall t \ \geq 0 \wedge \forall x, y \epsilon : \mathbb{R}$

$$|\mu(t,x) - \mu(t,y)| \leq k|x-y|$$
$$|\sigma(t,x) - \sigma(t,y)| \leq k|x-y|$$
$$|\sigma(t,x)| + |\mu(t,x)| \leq k(1-|x|)$$

Then a solution for the SDE exists and it has the following properties:

I. $\mathcal{F}_t^W$-filtration adapted
II. Continuous trajectory
III. It is a Markov process

[5] Notice that instead of Δ and Θ is used respectively μ and σ.

1.3.1 Linear Stochastic Differential Equation (SDE)

It is the easiest case of SDE, where μ and σ are constant, and it is represented as:

$$\begin{cases} dX(t) = \mu dt + \sigma dW(t) \\ X_0 = x_0 \end{cases}$$

The solution of this system is achievable making a parallelism with linear ODE and exploiting what it is known about them.

Linear ODE	Linear SDE
$\frac{dx}{dt} = a(t)x(t) + u(t)$	$\frac{dX(t)}{dt} = \mu(t)X(t) + \frac{dW(t)}{dt}$
Solutions for linear ODE	Solutions for linear SDE
$x(t) = e^{a(t)}x_0 + \int_0^t e^{a(t-s)}u(s)ds$	$X(t) = e^{\mu(t)}X_0 + \sigma \int_0^t e^{a(t-s)}dW(s)$

Table 1.1: Parallelism between linear ODE and linear SDE

This correspondence is workable in consequence of a linear structure which implies that the second order in the SDE's Itô formula doesn't appear at all.
It can also be computed a more general rule considering multidimensional processes.[6]

1.3.2 Geometric Brownian Motion

This process is of primary importance for option pricing as it is assumed that the price of the underlying stock is exactly a Geometric Brownian Motion (GBM).
The GBM is represented as:

$$\begin{cases} dX(t) = \alpha X(t)dt + \sigma X(t)dW(t) \\ X_0 = x_0 \end{cases}$$

It can be immediately noticed that it is a generalization of a linear SDE.

[6] See BJÖRK T. (2009), *Arbitrage Theory in Continuous Time*. United States, Oxford University Press Inc. ISBN 978-0-19-957474-2

The resolution to the equation is furnished by applying the technique of separation of variables, which is also another method to find a solution for linear ODEs.

For ODEs the procedure consists in four steps:

- Find the separable representation of the normal form of the equation

$$x'(t) = f(t, x) = f_0(t)g(x(t))$$

- Write it in a more usable way

$$\frac{dx(t)}{dt} = f_0(t)g(x(t))$$

- Integrate both parts

$$\int \frac{dx(t)}{g(x(t))} = \int f_0(t)dt$$

Likewise for a linear SDE, we obtain:

$$\int \frac{dX(t)}{X(t)} = \int \left(\alpha dt + \sigma dW(t)\right) dt$$

It can be observed that $\mathrm{dln}\left(X(t)\right) = \frac{dX(t)}{X(t)}$ so applying the Itô formula to $\mathrm{dln}\left(X(t)\right)$, it can be reached the solution:

$$\ln(X(t)) = \ln(x_0) + \left(\alpha - \frac{1}{2}\sigma^2\right) t + \sigma W(t)$$

Elevating to the exponential:

$$X(t) = x_0 \cdot e^{\left(\alpha - \frac{1}{2}\sigma^2\right)t} \cdot e^{\sigma W(t)}$$

An interesting result is provided by its expected value:

$$E[X(t)] = x_o \cdot e^{\alpha t}$$

1.3.3 Itô Operator

The role of this operator will be clear later, in the discussion related to the Black-Scholes-Merton Model. Now, it is only provided the definition.

Defined an n-dimensional SDE: $dX(t) = \mu(t, X(t))dt + \sigma\big(t, X(t)\big)dW(t)$.

The partial differential operator $\mathcal{A}$, called Itô operator or infinitesimal operator of $X(t)$, is specified $\forall h(x)$ with $h \epsilon C^2(\mathbb{R}^n)$ by:

$$\mathcal{A}h(t,x) = \sum_{i=1}^{n} \mu_i(t,x) h_{x_i}(x) + \frac{1}{2}\sum_{i,j=1}^{n} C_{ij}(t,x) h_{x_i x_j}(x) \quad where\ C(t,x) = \sigma\sigma^T$$

1.4 The Black-Scholes-Merton Model[7]

The model is based on several assumption regarding its three components:

I. a stock with price $S(t)$,
II. a risk free asset with price $B(t)$,
III. a contingent claim[8] with price $\Pi(t, \mathcal{X})$.

The hypothesis concerning the first two elements are three:

I. The stock price follows a Geometric Brownian Motion.

$$dS(t) = \alpha(t, s(t))S(t)dt + \sigma(t, S(t))S(t)d\overline{W}(t)^{9}$$

II. Both are exchanged in a continuous time model.
III. It is possible to sell or buy any amount of the stock and the derivative (i.e short selling is allowed).

[7] The difference between the Black and Scholes Model and the Black-Scholes-Merton Model is that the second considers also the stocks which pay dividends.
[8] A **contingent claim** with maturity T is any stochastic variable $\mathcal{X} \epsilon \mathcal{F}_T^S$. It is a simple claim if it's of the form $\mathcal{X} = \Phi(S(T))$ where **Φ** is the **contract function.**
[9] The symbol $\overline{\ }$ on W indicates that the model is built considering risk neutral probabilities instead of the real ones.

Furthermore, there are other four presumptions on the market where all the three constituents are traded:

I. <u>There are no arbitrage opportunities in the market</u>.
The market is arbitrage free if there no exists arbitrage possibilities. Arbitrage is possible when a self-financial portfolio, h, can be built and its value process V^h satisfies the following conditions:
   - $V^h(0) = 0$,
   - $P(V^h(T) \geq 0) = 1$,
   - $P(V^h(T) > 0) > 0$.

II. The market is frictionless (there no exists transaction or tax costs)

III. The rate of return on the risk free asset, $r(t)$, is constant.

IV. The borrowing and lending interest rate is the same.

The fundamental equation of the model, obtained determining the initial capital required to perfectly hedge a short position on the derivative, is the so called Black-Scholes-Merton (BSM) equation.
The used dynamic hedging strategy led to a partial differential equation which rules the price of the derivative.
The solution of the equation is given by the BSM formula. Let's see in details.

1.4.1 The BSM equation

A consequence of the assumption already explained is that $\Pi(t, \mathcal{X})$ is such that there are no arbitrage possibilities on the market. So the procedure to obtain the BSM equation it is basically a demonstration of the existence of a self-financing portfolio, h, so that $dV^h(t) = \alpha(t)V^h(t)dt \; where \; \alpha(t) = r(t)$. Here the procedure structured in three steps.

a. <u>Study the evolution of a discounted portfolio</u>

Let us assume that an agent at time t has a portfolio valued $X(t)$ formed by an investment in the money market account, paying the risk free rate r, and in a stock modeled by a Geometric Brownian Motion so that:

$$dS(t) = \alpha S(t)dt + \sigma S(t)d\overline{W}(t)$$

In addition consider that the investor holds $\Delta(t)$ shares of stock where $\Delta(t)$ is a stochastic process adapted to a filtration associated with $\overline{W}(t), t \geq 0$. The remaining part is invested in the money market account.

Now we are able to write the equation for the differential $dX(t)$:

$$dX(t) = \Delta(t)dS(t) + r(X(t) - \Delta(t)S(t))dt$$

Substituting $dS(t)$, it is obtained:

$$dX(t) = \underline{rX(t)dt} + \underline{\Delta(t)(\alpha - r)S(t)dt} + \underline{\Delta(t)\sigma S(t)d\overline{W}(t)}$$

Where each of the three parts has a financial significance:

I. The first indicates an average underlying rate of return r from the portfolio.
II. The second points out the risk premium given by the investment in the stock.
III. The third shows the stock volatility in proportion to the held quantity.

Moreover the stock price and the portfolio are considered discounted in t, under the hypothesis of a compound interest regime. Thus, it is interesting to define the differential for this two new variables:

- $d\left(e^{-rt}S(t)\right) - It\hat{o}\ Formula \rightarrow (\alpha - r)e^{-rt}S(t)dt + \sigma e^{-rt}S(t)dW(t)$
- $d\left(e^{-rt}X(t)\right) \ - It\hat{o}\ Formula \rightarrow \ \Delta(t)d\left(e^{-rt}S(t)\right)$

The most important result is given by the last equation which indicates that changes in discounted portfolio are given only by the variation of the discounted stock price.

b. **Study the evolution of derivative**

Defining $\Pi(t, \mathcal{X}) = F(t, S(t))$, where F is a smooth function, as the value at t of a contingent claim where the stock price is valued $x = S(t)$; the objective is to get what F must look like so that the market is arbitrage free.

First, it is calculated the differential for $F(t, x)$:

$$\left[F_t + \alpha F_x + \frac{1}{2}\sigma^2 S^2(t) F_{xx}\right] dt + \sigma S(t) F_x dW(t)$$ [10]

---

[10] [11]In the formula $\left(t, S(t)\right)$ has been omitted to simplify it.

And then for the price discounted $d(e^{-rt}F)$:

$$e^{-rt}\left[\left(-rF + F_t + \alpha S(t)F_x + \frac{1}{2}\sigma^2 S^2(t)F_{xx}\right)dt + \sigma S(t)F_x dW(t)\right]$$

c. <u>Equate the results obtained in a. and b.</u>

The only pricing function of the form $\Pi(t,\mathcal{X}) = F(t,x)$, consistent with the absence of arbitrage, is when F is the solution of the following boundary value problem:

$$\begin{cases} \left(de^{-rt}X(t)\right) = d(e^{-rt}F(t,x) \\ X(0) = F(0,S(0)) \end{cases}$$

Additionally, if the portfolio at t has to be risk free, the following system should be satisfied:

$$^{11}\begin{cases} \Delta(t) = F_x \quad \forall t \in [0,T) \qquad (delta\ hedging\ rule) \\ F_x \cdot S(t) \cdot (\alpha - r) = -rF + F_t + \alpha S(t)F_x + \frac{1}{2}\sigma^2 S^2(t)F_{xx} \end{cases}$$

The resolution give us the BSM partial differential equation:

$$\boldsymbol{F_t(t,x) + rS(t)F_x(t,x) + \frac{1}{2}\sigma^2 S^2(t)F_{xx}(t,x) = rF(t,x)} \qquad (1.1a)$$

And the terminal condition that the PDE has to satisfy:

$$\boldsymbol{F\left(T,S(t)\right) = \Phi(x)} \qquad (1.1b)$$

1.4.2 Considerations

- In an arbitrage free market F has to satisfy the BSM equation.
- The only relevant aspect of the underlying price process, when it comes to contingent claims (or derivatives), is the volatility.
- A PDE is a differential equation which contains unknown multivariate function and their partial derivative.
- Using the Feynman-Kač stochastic representation formula it can be shown that:

$$\boldsymbol{F(t,x) = e^{-r(T-t)}E^{Q}_{t,x}}{}^{12}\boldsymbol{\left[\Phi\left(S(T)\right)\right]} \qquad (1.2)$$

[12] The Q indicates that risk neutral probabilities are used.

Demonstration

Assume that F is a solution to:

$$F_t(t,x) + rS(t)F_x(t,x) + \frac{1}{2}\sigma^2 S^2(t)F_{xx}(t,x) = rF(t,x)$$

$$F\big(T, S(t)\big) = \Phi(\mathrm{x})$$

And that $e^{-rs}\sigma\big(s, X(s)\big)F_x\big(s, X(s)\big)$is in £^{2} where S is defined in a time interval $[t,T]$ as the solution to the SDE:

$$dS(s) = \alpha\big(s, S(s)\big)ds + \sigma\big(s, S(s)\big)dW(s)$$

$$S(t) = x$$

The Itô operator $\mathcal{A}$ for X is given by:

$$\mathcal{A}F(t,x) = \alpha^{13}S(t)F_x + \frac{1}{2}\sigma^2 S^2(t)F_{xx}$$

The boundary condition become:

$$F_t + \mathcal{A}F(t,x) - rF = 0$$

Applying now integrating the formula $de^{-rT}F\big(s, X(s)\big)$:

$$e^{-rt}\left[\left(-rF + F_t + \alpha S(t)F_x + \frac{1}{2}\sigma^2 S^2(t)F_{xx}\right)dt + \sigma S(t)F_x dW(t)\right]$$

$$e^{-rT}F\big(T, S(T)\big) - e^{-rt}F(t,x)$$

$$= \int_t^T e^{-rs}(\mathcal{A}F(t,x) - rF + F_t)dt + \int_t^T e^{-rs}\sigma S(s)F_x dW(s)$$

For definition $\mathcal{A}F(t,x) - rF + F_t = 0$

$$e^{-rT}F\big(T, S(T)\big) - e^{-rt}F(t,x) = 0 + \int_t^T e^{-rs}\sigma S(s)F_x dW(s)$$

Taking conditional expectations in both sides, the Itô integral is zero so:

$$E_{t,x}\big[e^{-rT}F\big(T, S(T)\big) - e^{-rt}F(t,x)\big] = 0$$

$$E_{t,x}\big[e^{-rT}F\big(T, S(T)\big)\big] = E_{t,x}[e^{-rt}F(t,x)]$$

$$e^{-rt}F(t,x) = e^{-rT}E_{t,x}\big[F\big(T, S(T)\big)\big]$$

Finally:

$$F(t,x) = e^{-r(T-t)}E^{Q}_{t,x}\big[\Phi\big(\mathrm{S}(\mathrm{T})\big)\big] \qquad \blacksquare$$

[13] Remember that the market is arbitrage free only if $\alpha = r$

This means that the price of a contingent claim is computed by taking the expected value of the final payment and the discounting it.

- Analytical solution of the PDE can be found for certain type of simple contingent claim such as European and American Options but for others, for example Asian Option, it is almost impossible to reach.

The next step is to use the Laplace transform to calculate an analytical solution for some off the most used options.

CHAPTER 2

THE LAPLACE TRANSFORM

2.1 Basic Principles

The interest on the Laplace Transform $(\mathcal{L})$ is given by its characteristics that allow to convert a differential problem, such as the option pricing, into an algebraic one of easier resolution. The obtained algebraic solution, generally represented in form of complex roots, is again transmuted into the solution of the original problem.

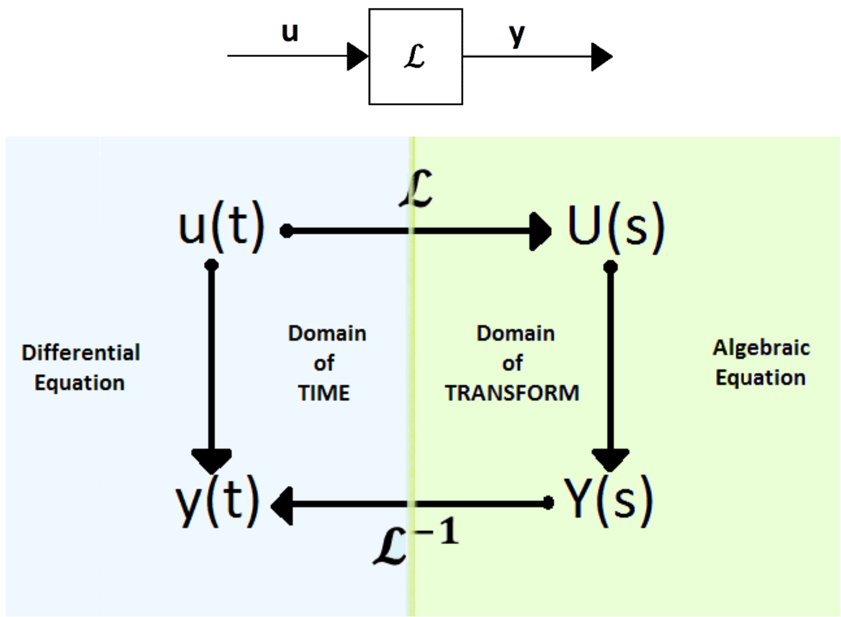

Figure 2.1: The Laplace Transform procedure

The shift between a differential issue and an algebraic one is called “transformation” or “transform” that is the action of an operator that turns the differential equation into an algebraic equation, functional variables into numerical variables, operations of derivation and integration into algebraic operations.

Once the solution of the transformed problem is found, it is fundamental to move to the solution of the original problem through an inverse operation, the so called inverse transform.

Having presented the general scheme to be followed for the application of a transform, let's see the specific case of the Laplace transform.

First of all, as usual, we will introduce the basic definitions.

Definition 2.1: Laplace Transform

Let f be a function dependent on the time $t,$ where:

- $t \in \mathbb{R}$
- $f(0) = 0 \; \forall \, t < 0$

and a parameter $s \in \mathbb{C},\; s = \alpha + i\omega$. The Laplace transform of the function $f(t)$ is defined as[14]:

$$F(s) = \mathcal{L}\big(f(t)\big) = \int_0^\infty e^{-st} f(t)dt \qquad (2.1)$$

$$= \lim_{\tau \to \infty} \int_0^\tau e^{-st} f(t)dt$$

[14] The operator $\mathcal{L}$ is built in order to be adapted to the enounced characteristics. It is notable that:

- It is a generalized integral so that the correspondence between $f(t)$ and the new imagine function $F(s)$ takes into account all the values that $f(t)$ can assume.
- The function e^{-st}, and in particular the two variables s and t , are introduced so the transform gives the "right" weights to every single value of the original function. In fact if the transformation were only a simple integration of the original function the new function would be a value that keeps in mind only "broadly" the value of $f(t)$ in $[0, \infty)$.
- The function weight in the formula is given by e^{-st}. It was chosen the exponential function to assure that the improper integral converges in most of the cases. It is possible to notice that:

  $$e^{-st} = e^{-\alpha t} e^{i\omega t}$$

  where $e^{-\alpha t}$ shoots down the probabilities that the integral doesn't exist as, at least for some value of α, it assures to weaken the initial function so that the generalized integral can exist also in case of a "bad" behavior of the initial function for $\tau \to \infty$.
  This guarantees that the number of functions for which the Laplace Transform exists is bigger than expected.

In spite of the precaution taken choosing e^{-st} as a weighting function for $f(t)$ in the definition of the Laplace Transform to ensure its convergence for a wider range of function, the existence of the transform of a general $f(t)$ is not guarantee, at least not on all $\mathbb{C}$.

This is the case, for example, of the function $f(t) = e^{at}$, which is really important in the study of problems regarding dynamic regimes. In this case:

$$\begin{aligned}\mathcal{L}(e^{at}) &= \int_0^\infty e^{-st}e^{at}\,dt \\ &= \int_0^\infty e^{-t(s-a)}\,dt \\ &= \left.\frac{e^{-t(s-a)}}{-(s-a)}\right|_0^\infty \\ &= \frac{1}{s-a}\end{aligned}$$

So, the integral:

- Exists and is finite for $Re(s) > a$[15]
- Doesn't exist for $Re(s) = a$
- Is unlimited for $Re(s) < a$

It is immediate to notice that the integral converges for all the value of $s = \alpha + i\omega$ with $\alpha > a$, in this case the transformation exists but it is not defined on all $\mathbb{C}$.

Actually, the situation is not as uncomfortable as it may appear since it is possible to determine some sufficient conditions of existence which will be explained hereunder.

Notice that all the theorems, lemmas, propositions and their demonstrations are taken principally from Churchill and Brown (2009)[6], Spiegel (1965)[7], and Cohen (2007)[5].

[15] The symbol $Re(x)$ means for all the real values of x.

2.2 Convergence

Definition 2.2: Piecewise Continuous Function

A function *f* is said to be **piecewise continuous** on an interval $[a, b]$ if the interval can be subdivided into a finite number of intervals in each of which the function is continuous and has finite right and left hand limits.

In the Laplace Transform definition, in particular see the formula 2.1, it is chosen $s > 0$ or $Re(s) > 0$, thus the integral will converge as long as $f(t)$ does not grow too rapidly. So an appropriate rate of growth for the function $f(t)$ can be defined.

Definition 2.3: Exponential Order

A function f has exponential order α if there exist constants $M > 0$ and α such that for some $t_0 \geq 0$:

$$|f(t)| \leq Me^{\gamma t}, \qquad \forall t \geq t_0$$

Intuitively, functions of exponential order can't grow in absolute value more rapidly than the fixed threshold as t increases. In practice, this is not a restriction because γ and M can be selected as large as desired.
Bounded functions, such as $\sin(at)$ or $\cos(at)$, are of exponential order.

Given these two definition, we are able to identify a large class of functions which possess a Laplace Transform.

Theorem 2.1: Existence of the Laplace Transform

If f is piecewise continuous on the interval $[0, \infty)$ and of exponential order α, then the Laplace Transform $\mathcal{L}(f)$ exists for $Re(s) > \alpha$ and converges absolutely.

Proof (found in Spiegel (1965)[7])

What it is known from definitions is that:

$$|f(t)| \leq M_1 e^{\alpha t}, \qquad t \geq t_0 \text{ for some real } \alpha$$

As f is piecewise continuous on $[0, t_0]$, it is also bounded, say:

$$|f(t)| \leq M_2, \qquad 0 < t < t_0$$

Since $e^{\alpha t}$ has a positive minimum on $[0, t_0]$, a constant M can be chosen sufficiently large so that

$$|f(t)| \leq Me^{\alpha t}, \qquad t > 0$$

Hence:

$$\int_0^\tau |e^{-st} f(t)| dt \leq M \int_0^\tau e^{-(x-\alpha)t} dt$$

where:

$$M \int_0^\tau e^{-(x-\alpha)t} dt = \left. \frac{Me^{-(x-\alpha)t}}{-(x-\alpha)} \right|_0^\tau$$

$$= \frac{M}{x-\alpha} - \frac{Me^{-(x-\alpha)t}}{x-\alpha}$$

Now, letting $\tau \to \infty$ and remembering that by assumption there exists a real number: $Re(s) = x > \alpha$, it can be said that:

$$\int_0^\infty |e^{-st} f(t)| dt \leq \frac{M}{x-\alpha}$$

It is immediate to see that the Laplace integral converges for $Re(s) > \alpha$.

It must be emphasized that the stated conditions are sufficient to guarantee the existence of the Laplace Transform but they are not necessary. If the conditions are not satisfied, however, the L.T. may or may not exist. It is the case, for example, of $f(t) = t^{-1/2}$.

2.3 Basic Properties

In the following list of theorems we assume that all functions satisfy the conditions for the Laplace Transform existence.

Theorem 2.2: Linearity property

Let c_1 and c_2 be constants; then:

$$\mathcal{L}(c_1 f_1 + c_2 f_2) = c_1 \mathcal{L}(f_1) + c_2 \mathcal{L}(f_2)$$

Theorem 2.3: First Translation or Complex Translation property

Let a be a constant; then:

$$\mathcal{L}\left(e^{at} f(t)\right) \; = \; \mathcal{L}\left(f(t)\right)(s-a)$$

Multiply f by e^{at} replaces s by $(s-a)$.

Theorem 2.4: Second Translation or Time Translation property

Let a be a constant, $a \geq 0,$ and $u_a(t)$ a heavy-side function [3]; then:

$$\mathcal{L}\left(u_a(t) f(t-a)\right) = e^{-as} \mathcal{L}\left(f(t)\right)$$

Theorem 2.5: Change of Scale property

Let a be a constant; then:

$$\mathcal{L}\left(af(t)\right) \; = \; \frac{1}{a} \mathcal{L}\left(f(t)\right)\left(\frac{s}{a}\right)$$

Multiply f by a replaces s by (s/a)

Theorem 2.6: Derivative property

We start giving the property for the first derivative:

$$\mathcal{L}\left(f'(t)\right) = s\mathcal{L}\left(f(t)\right) - f(0^+)$$

And then we extend the formula for the n-th derivative:

$$\mathcal{L}\left(f^{(n)}(t)\right) = s^n \mathcal{L}\left(f(t)\right) - s^{n-1} f(0^+) - f^{(n-1)}(0^+).$$

This property is often used in finding Laplace Transform without integration.

Example 2.1

If $f(t) = 1$ then $f'(t) = 0$ and $f(0) = 1$.

Then, for the derivative property,

$$\mathcal{L}(0) = 0 \text{ and } \mathcal{L}(0) = 0 = s\mathcal{L}(1) - 1 \text{ or } \mathcal{L}(1) = \frac{1}{s}$$

Theorem 2.7: Integration property

The integration property can be expressed as follow:

$$\mathcal{L}(\int_0^t f(\tau)d\tau) = \frac{1}{s}\mathcal{L}(f(t))$$

Theorem 2.8: Multiplication by t^n

The Laplace transform of a function $f(t)$ multiplied for t^n is given by:

$$\mathcal{L}\big(t^n f(t)\big) = (-1)^n \frac{d^n}{ds^n}\mathcal{L}\big(f(t)\big)$$

Theorem 2.9: Division by t

The Laplace transform of a function $f(t)$ divided by t is equal to:

$$\mathcal{L}\left(\frac{f(t)}{t}\right) = \int_0^\infty \mathcal{L}\big(f(u)\big)du$$

Theorem 2.10: Periodic funtion

If $f(t)$ is periodic with period T; then:

$$\mathcal{L}\big(f(t)\big) = \frac{1}{1 - e^{-sT}} \int_0^T e^{-st} f(t)dt$$

Theorem 2.11: Behavior of $\mathcal{L}\big(f(t)\big)$ as $s \to \infty$

The Laplace transform of $f(t)$ for $s \to \infty$ is equal to 0. We can write it also as:

$$\lim_{s\to\infty} \mathcal{L}\big(f(t)\big) = 0$$

Theorem 2.12: Initial-value

If $\mathcal{L}\big(f(t)\big) = F(s)$ and the following two limits exists, then

$$\lim_{t\to 0} f(t) = \lim_{s\to\infty} sF(s)$$

Theorem 2.13: Final value

If $\mathcal{L}\big(f(t)\big) = F(s)$ and the following two limits exists, then

$$\lim_{t\to \infty} f(t) = \lim_{s\to 0} sF(s)$$

Theorem 2.14: Convolution property

$$\mathcal{L}[(f \cdot g)(t)] = \mathcal{L}(f(t)) \cdot \mathcal{L}(g(t))$$

2.4 Laplace Transform Calculus

The resolution of a differential problem via the Laplace Transform doesn't require necessarily the computation of the definition integral. In fact, this operation is usually not effortless and, in general, when it is asked the estimation of the transform of a function this is "built" by resorting to the transform properties and to the known transform of some special function.

Hereinafter it is reported a minimal table (Table 2.1), which can be easily calculated applying the definition, and another table (Table 2.2), which is an extension of the former calculated using Table 2.1 itself and the transform integral properties.

The extended Table 2.2 is quoted for the purpose of explanation only; for a substantial review see Spiegel (1965) [7].

This combination is known as the "Laplace Transform Calculus".

$\mathcal{L}(t^n) = \frac{n!}{s^{n+1}}$
$\mathcal{L}(e^{at}) = \frac{1}{s-a}$
$\mathcal{L}(\cos(bt)) = \frac{s}{s^2+b^2}$
$\mathcal{L}(\sin(bt)) = \frac{b}{s^2+b^2}$

Table 2.1: A basic Laplace integral table[16]

$\mathcal{L}\big(H(t-a)\big) = \frac{e^{-as}}{s} \quad a \geq 0$
$\mathcal{L}\big(\delta(t-a)\big) = e^{-as}$
$\mathcal{L}\left(floor\left(\frac{t}{a}\right)\right) = \frac{e^{-as}}{s(1-e^{-as})}$
$\mathcal{L}\left(sqw\left(\frac{t}{a}\right)\right) = \frac{1}{s}\tanh\left(\frac{as}{2}\right)$
$\mathcal{L}\left(atrw\left(\frac{t}{a}\right)\right) = \frac{1}{s^2}\tanh\left(\frac{as}{2}\right)$
$\mathcal{L}(t^{\alpha}) = \frac{\Gamma(1+\alpha)}{s^{\alpha+1}}$
$\mathcal{L}\left(t^{-1/2}\right) = \sqrt{\frac{\pi}{s}}$

Table 2.2: An extension of the previous table for Laplace integral[16]

[16] The tables are taken from Schiff (1999) [3].

Let’s now see how it is possible to use Laplace calculus (and so the previous tables) in a differential problem.

Example 2.2

Solve the following differential equation:

$$f'(t) + 2t - 5 = 0 \qquad f(0) = 1$$

Isolating the “differential” term and multiplying both sides for e^{-st} and integrating between 0 and ∞, we obtain:

$$\int_0^\infty f'(t)e^{-st}dt = \int_0^\infty (5 - 2t)e^{-st}dt$$

Or rather,

$$\mathcal{L}\big(f'(t)\big) = \mathcal{L}(5 - 2t)$$

Applying Theorem 2.6 to the left side and Theorem 2.2 to the right and using Table 2.1, we have:

$$s \cdot \mathcal{L}\big(f(t)\big) - f(0) = \frac{5}{s} - \frac{2}{s^2}$$

$$\mathcal{L}\big(f(t)\big) = \frac{1}{s} + \frac{1}{s^2} + \frac{1}{s^3}$$

Using Table 2.1 and Theorem 2.2 again, we find:

$$\mathcal{L}\big(f(t)\big) = \mathcal{L}(1) + 5\mathcal{L}(t) - \mathcal{L}(t^2) = \mathcal{L}(1 + 5t - t^2)$$

Finally, here is the solution:

$$f(t) = 1 + 5t - t^2$$

Remark

In the last step, doing the reconversion from the domain of the transform to the domain of time, it has been implicitly assumed that $f(t)$ of its given $F(s)$ is unique.

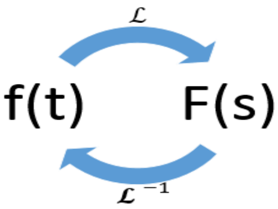

Figure 2.2: Instant reconversion

The indicated transition, which can be interpreted as the application of an inverse operator $\mathcal{L}^{-1}$ of $\mathcal{L}$, seems unnatural since it is undeniable that the operator $\mathcal{L}$, as integral, is not injective and cannot distinguish between function in the domain with the same imagine in the codomain. Graphically:

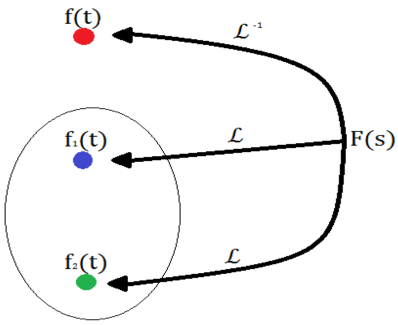

Figure 2.3: Laplace Transform function is not injective

It is similar the liaison between the symbols of improper integral as inverse operator of the derivation procedure.

This is why the writing $\mathcal{L}^{-1}\big(F(s)\big) = f(t)$ means: "$f(t)$ is one of the possible transformable function such that $\mathcal{L}\big(f(t)\big) = F(s)$".

2.5 The Inverse Laplace Transform

In this section it is analyzed the problem of inverting the transform using analytical methods; more information can be found in Churchill and Brown (2009) [6].

First of all we give a precise definition of what the inverse of the Laplace Transform is.

Then we enunciate the Lerch's Theorem which guarantees that the solution found using the Laplace Calculus method (see par.2.3, example 1) is unique.

After that it is stated the theorem for the Complex Inversion formula which gives the possibility to come back in the domain of time in all the cases that cannot be solved drawing upon the most used formulae of the transform.

However, the gained formula is a complex integral which is not always solvable. So, in the last part it is explained a procedure that can be used to solve this integral only under precise conditions.

Definition 2.4: The Inverse Laplace Transform

If the Laplace Transform of a function $f(t)$ is $F(s)$, i.e. if $\mathcal{L}\big(f(t)\big) = F(s)$, then $f(t)$ is called an inverse Laplace Transform of $F(s)$. We will use the following representation:

$$f(t) = \mathcal{L}^{-1}(F(s))$$

Theorem 2.15: Lerch's theorem

Suppose that f and g are continuous functions.

If $\mathcal{L}\big(\,f(t)\big) = \ \mathcal{L}\big(g(t)\big)$ then $f(t) = g(t)$.

Proof.

Let us start from the hypothesis and assume that:

$$\mathcal{L}(\ f(t)\ ) - \mathcal{L}(\ g(t)\ ) = 0$$

Then for Theorem 2.2 it is possible to say:

$$\mathcal{L}(\ f(t) - g(t)) = 0$$

Finally, if the theorem is true: $f(t) - g(t) = 0$.

To simplify the demonstration, $f(t) - g(t)$ is substituted with $h(t)$.

It has to be verified that $h(t) = 0$ and this is done using the following lemma.

Lemma 1

Suppose that:

- a function $h(u)$ is continuous in the interval $[0,1]$,
- $\int_0^1 h(u)u^n du = 0$

Then:

$$h(u) = 0$$

Proof

Any continuous function on $[0,1]$ is the uniform limit of polynomials[17]. It means that for any small number ε it can be found a polynomial P_ε such that

$$|h(u) - P_\varepsilon(u)| < \varepsilon$$

From the hypothesis, it is known that

$$\int_0^1 h(u)P_\varepsilon(u)du = 0$$

Taking $\varepsilon \to 0$ implies

$$\int_0^1 h(u)h(u)du = 0$$

and because $h(u)^2 \geq 0$ we have that $h(u) = 0$. ■

Using the new notation, the assumption becomes:

$$\mathcal{L}\big(h(t)\big) = \int_0^\infty h(t)e^{-st}dt = 0 \quad \forall s: Re(s) > \alpha$$

Now, choose a number s_0 which is positive, real and higher than α. (Notice that this conditions are made so to be sure that the integral converges).

Then establishing $s = s_0 + n + 2$, where $n \epsilon \mathbb{N}$, we get:

$$\mathcal{L}\big(h(t)\big) = \int_0^\infty h(t)e^{-nt}e^{-s_0 t}e^{-2t}dt = 0$$

---

[17] For the demonstration of this lemma other notions and theorems are necessary which are not reported here.

Making the change of variable $u = e^{-t}$, the result is

$$\mathcal{L}\big(h(t)\big) = \int_0^1 u^n\big(u^{s_0}h(-\ln(u))\big)dt = 0$$

Thanks to the result of Lemma 1, it can be said that $u^{s_0}h(-\ln(u))$=0. ■

Theorem 2.16: Complex inversion (Fourier-Mellin) formula

If $F(s) = \mathcal{L}\big(f(t)\big)$ then

$$\boldsymbol{f(t)} = \begin{cases} \boldsymbol{\dfrac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{ts}F(s)ds} & \boldsymbol{for\ t > 0} \\ \boldsymbol{0} & \boldsymbol{for\ t < 0} \end{cases} \quad (2.2)$$ [18]

Now, it is required to establish the validity of the complex inversion formula. This is done by using a theorem from the Complex Variable Theory known as Fourier's integral theorem. However, for a complete proof see Churchill and Brown (2009) [6].

Proof

We begin by considering the definition of Laplace Transform

$$\mathcal{L}\big(f(t)\big) = \int_0^\infty e^{-st}f(t)dt$$

Then it is possible to rewrite the integral in the Fourier-Mellin formula (2.2), as follows:

$$\frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty}\int_0^\infty e^{ts-su}f(u)duds$$

Defining $s = \gamma + iy$ and choosing it so that its real part is fixed higher than α, it is possible to state that the integral depends only on the variation of y. So,

$$ds = idy$$

[18] The condition $f(0) = 0\ \forall t < 0$ might seem too much restrictive but it is essential to guarantee the uniqueness of the transform. In addition, from a practical point of view, the restriction is acceptable as the time interval considered is between an initial instant always fixed equal to 0 and ah horizon T.

and the Fourier Mellin formula (2.2) can be expressed as follows

$$\frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty}\int_{0}^{\infty} e^{ts-su} f(u)duds = \frac{1}{2\pi i}\int_{-\infty}^{\infty}\int_{0}^{\infty} e^{(\gamma+iy)(t-u)}\ f(u)du\,idy$$

$$=\frac{1}{2\pi}\int_{-\infty}^{\infty}\int_{0}^{\infty} e^{\gamma t}\ e^{-\gamma u}e^{ity}e^{-iuy}f(u)du\,dy$$

$$=\frac{1}{2\pi}\ e^{\gamma t}\int_{-\infty}^{\infty} e^{ity}dy\int_{0}^{\infty}\ e^{-iuy}\{e^{-\gamma u}f(u)\}du$$

Using the Fourier’s integral theorem for which

$$g(t)=\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{iyt}dy\int_{-\infty}^{\infty}\ e^{-iyu}\ g(u)du$$

it is possible to see that

$$e^{\gamma t}\frac{1}{2\pi}\int_{-\infty}^{\infty} e^{ity}dy\int_{0}^{\infty}\ e^{-iuy}\{e^{-\gamma u}f(u)\}du = e^{\gamma t}\{e^{-\gamma t}f(t)\} = f(t)$$

Thus

$$\frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{ts}F(s)ds = f(t) \qquad \blacksquare$$

At that point the problem is to find a method to search for an analytical solution for this linear integral.

A possible way to solve it is to employ the *contour integral* which is defined as

$$\frac{\mathbf{1}}{\mathbf{2\pi i}}\oint_{C} \boldsymbol{e^{ts}\ F(s)ds} \qquad (2.3)$$

where C is the contour showed in the Figure 2.4 and it is called Bromwich contour. It is composed by:

- the segment AB
- the arch Γ $(BJKLA)$

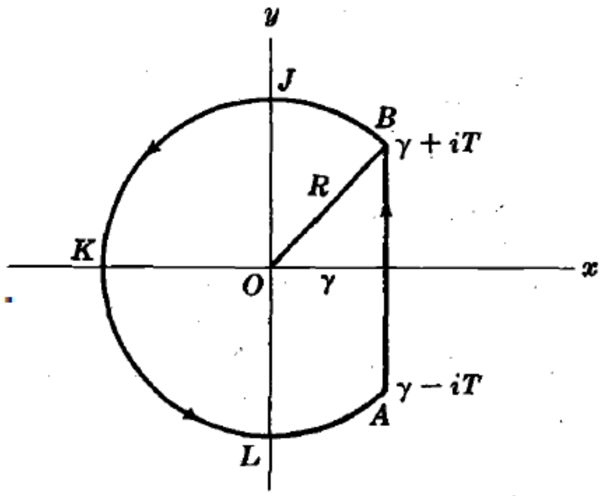

Figure 2.4: Bromwich Contour. (Spiegel, 1965) [7]

In view of all this, we have to proceed as it is reported hereafter:

1. Start from the Fourier-Miller formula

$$\frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{ts}F(s)ds$$

2. Define the contour integral

$$\frac{1}{2\pi i}\oint_C e^{ts}\, F(s)ds$$

3. Make $R \to \infty$ so that

$$\frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{ts}F(s)ds = \left\{\lim_{R\to\infty}\frac{1}{2\pi i}\oint_C e^{ts}\, F(s)ds - \frac{1}{2\pi i}\int_\Gamma e^{ts}F(s)ds\right\}$$

4. Control the following conditions:
   - All the poles[19] are at the left of the line $s = \gamma$
   - The integral on Γ goes to zero as $R \to \infty$[20]

---

[19] A pole is a type of singularity such that when the function goes to infinity it approaches a fixed value.

[20] The sufficient requisite for which the integral on Γ goes to zero as $R \to \infty$ is given by the following **theorem**.

If it is possible to find two constants $M > 0, k > 0$ such that on Γ

$$|F(s)| < \frac{M}{R^k} \qquad s = Re^{i\theta}$$

Then the integral around Γ of $e^{ts}F(s)$ approaches zero as $R \to \infty$.

Then the Residue Theorem of Cauchy can be applied and the solution is given by

$$f(t) = \sum residues\ of\ \ e^{st}\ F(s) at\ poles\ of\ F(s)$$

The last considerations were made without specifying all the details because it would have required a deep knowledge about the theory of complex variables and this falls outside the target of this dissertation. However for an insight of the Residue Theorem of Cauchy we remand to Appendix A, section A.3.

Nevertheless it is important to illustrate an analytic way to determine the antitransform, even though it can be so hard to manage that most of the times the Fourier-Mellin formula has to be integrated numerically.

2.6 Numerical Inversion of the Laplace Transform

Let's now consider the creation of an algorithm for the numerical inversion of the Laplace Transform. The main idea is to find out an evaluation of the antiderivative by estimating the integral exploiting some approximation algorithms.

In this specific case we have the Fourier-Mellin integral inscribed as a contour integral (Bromwich's contour):

$$f(t) = \frac{1}{2\pi i} \oint_C e^{ts}\, F(s) ds \qquad t > 0$$

Using the procedure shown by Wellekens (1970) [8], we make the change of variable $z = st$

$$f(t) = \frac{1}{2\pi i t} \oint_{C'} e^{z}\, F(z/t) dz \qquad t > 0$$

where C' is the same contour as a function of z.

Then we approximate the exponential function e^z with the rational function

$$e^z \ \approx \sum_{k=0}^{n} \frac{\omega_k}{(\alpha_k - z)}$$

where z is a complex variable and α_k and ω_k are complex numbers.

By substitution we obtain:

$$f(t) \approx f_n(t) = \frac{1}{2\pi i t} \oint_{C'} F(z/t) \sum_{k=0}^{n} \frac{\omega_k}{(\alpha_k - z)} dz$$

$$= \frac{1}{t} \sum_{k=0}^{n} \frac{1}{2\pi i} \oint_{C'} F(z/t) \frac{\omega_k}{(\alpha_k - z)} dz$$

Using the Cauchy Integral Formula (see Appendix A, section A.4), we get

$$f(t) \approx f_n(t) = \frac{1}{t} \sum_{k=0}^{n} \omega_k F(\alpha_k/t) \qquad t > 0$$

The result is an algorithm for the numerical inversion of the Laplace Transform where the function $f(t)$ is approximated by a finite linear combination of the value of the transform.

Clearly, there are advantages in having nodes, α_k, and weights, ω_k, in the representation of $f_n(t)$ which are independent from $F(s)$ and t. One is that the procedure can be efficiently applied to multiple transforms and multiple time points.

When the studied function is a real function, the following estimation can be used:

$$Re(f(t)) \approx Re(f_n(t)) = \frac{1}{t} \sum_{k=0}^{n} Re(\omega_k F(\alpha_k/t))$$

$$= \frac{1}{t} \sum_{k=0}^{n} \left[Re(\omega_k) Re\big(F(\alpha_k/t)\big) - Im(\omega_k)\, Im\big(F(\alpha_k/t)\big)\right]$$

Given $n, \omega_k$and α_k ,$f_n(t)$ is an approximation of $f(t)$ $\forall t \in (0, \infty)$. There can be numerical difficulties when t is too big or too small, and usually in these special cases we draw upon the initial and final value theorems, (1.12) and (2.13), for $t \to 0$ and $t \to \infty$.

For any specific t we can increase n in order to obtain a better accuracy.

The estimated formula obtained for $f(t)$, i.e

$$f(t) \approx f_n(t) = \frac{1}{t} \sum_{k=0}^{n} \omega_k F(\alpha_k/t) \qquad t > 0 \qquad (2.4)$$

can be considered as basic formula to build specific inversion algorithms., which, from now on, will be called "general framework",

In the next section we present three different ways to evaluate ω_k and α_k.

Note that the efficiency of the found algorithms in terms of required precision and produced significant digits, as showed in Abate and Valkó (2004) [9], is a function of n.

2.6.1 The Gaver-Stehfest Algorithm

In this algorithm the weights and the nodes are real numbers. This algorithm was derived by Gaver, see Gaver (1966) [10], and is known as "Gaver inversion formula". It can be written as follows

$$f(t) \approx f_n(t) = \frac{n\ln(2)}{t}\binom{2n}{n}\sum_{k=0}^{n} (-1)^k \binom{n}{k} F\big((n+k)\ln(2)/t\big)$$

The Gaver's approximation can be computed by a recursive algorithm, see always Gaver (1966) [10].

Unfortunately the convergence of $f_n(t)$ to $f(t)$ as $n \to \infty$ is slow so that it has to be accelerated. For example, $f_{1000}(t)$ yields an estimate of $f(t)$ with only two or three digits of accuracy.

To achieve a good approximation, a convergence acceleration algorithm is required for $f_n(t)$. In order to do that Stehfest, see Stehfest (1970) [12], proposed the Linear Salzer acceleration scheme, presented in Salzer (1956) [11], reaching the "*Gaver-Stehfest inversion formula*"

$$f_g(t,M) = \sum_{n=1}^{M} (-1)^{n+M} \left(\frac{n^M}{M!}\right)\binom{M}{n} f_n(t) \qquad \forall t > 0 \wedge M \in \mathbb{N}$$

where $f_n(t)$ is the "*Gaver inversion formula".*

Making some computation, we get

$$\boldsymbol{f_g(t,M) = \frac{\ln(2)}{t} \sum_{k=1}^{2M} \zeta_k \; F(k\ln(2)/t)} \tag{2.5}$$

where

$$\zeta_k = (-1)^{M+k} \sum_{j=\lfloor (k+1)/2 \rfloor^{21}}^{\min(k,M)} \frac{j^{M+1}}{M!} \binom{M}{j} \binom{2j}{j} \binom{j}{k-j}$$

The *Gaver-Stehfest inversion formula* is in the form of "*general framework*" with:

- $n = 2M$
- $\alpha_k = \ln(2)$
- $\omega_k = ln(2)\zeta_k$

Furthermore, note that:

$$\sum_{k=0}^{2M} \zeta_k \ = 0 \ \forall \, M \geq 1.$$

The Gaver-Stehfest algorithm has four interesting properties:

i. the approximations $f_n(t)$ are linear in $F(s)$ values;
ii. the algorithm requires that the values of $F(s)$ are real;
iii. the coefficients ζ_k can be easily computed;
iv. the Gaver-Stehfest approximations are exact for constant functions so if $f(t) = c$ then $f_n(t) = c \ \forall n \geq 1$.

This algorithm was studied in Davies & Martin (1979) [23], where it was numerically demonstrated that $f_n(t)$ converges very quickly to $f(t)$ for many initial functions $f(t)$ that are non-oscillating.

It is well known that this algorithm requires arithmetical high-precision for its implementation. This consideration is rather obvious, since the coefficients ζ_k are growing very rapidly and alternate in sign.

This is a delicate matter as the precision[22] is higher as M rises up but, on the other hand, the accuracy[23] decreases as M increases. The last fact is due to the inevitable propagation of the uncertainty that comes from the calculation itself. It seems like a dog chasing its tail.

To overcome this problem, Abate and Valko (2004) [9] made some experiments to analyze the precision as a function of the parameter M.

[21] With $\lfloor (k+1)/2 \rfloor$ being the greatest integer less than or equal to $(k+1)/2$

[22] The precision is intended as the number of significant figures.

[23] The accuracy is defined as the number of significant digits in the estimation.

They found that:

1. The required system precision (the number of precision decimal digits) is about 2.2 time M, indicated as $2.2M$.
2. About $0.90M$ significant digits are produced for $f(t)$ with good transform[24].

Thus the efficiency of the Gaver-Stehfest algorithm, measured by the ratio of significant digits produced over the precision required, is:

$$eff(G) = \frac{\text{significant digits produced}}{\text{precision required}} \approx \frac{0.90\ \mathrm{M}}{2.2M} \approx 0.4$$

For further information see Kunznetsov (2013) [13].

Now, let us summarize the procedure to use the GS-algorithm:

1. Specify the transform $F(s)$.
2. Set $M = \lceil 1.1j \rceil$, where j is the number of significant desired digits.
3. Set the system precision equal to $\lceil 2.2M \rceil$
4. Compute the weights $\zeta_k, 1 \leq k \leq 2.2M$.
5. Finally, calculate the approximant $f_g(t, M)$.

2.6.2 The Euler Algorithm

The Euler algorithm is an implementation of the Fourier-series method. The Euler summation is exploited to accelerate convergence of the final infinite series. How it is shown in Abate, Choudhury & Whitt (1999) [14], the Euler inversion formula is applied to calculate numerically $f(t)$ when f is a real function.

---

[24] Transforms are said to be “good” if all their singularities are on the negative real axis and the functions f are infinitely differentiable for all $t > 0$. If the transforms are not good, then the number of significant digits may not be so great and may not be proportional to M.

The approximants is:

$$\boldsymbol{f_e(t,M) = \frac{10^{M/3}}{t} \sum_{k=0}^{2M} \eta_k \; Re(F\,(\beta_k/t)\,)} \tag{2.6}$$

Where

$$\beta_k = \frac{M ln(10)}{3} + \pi i k, \qquad \eta_k = (-1)^k\, \zeta_k$$

With

1. $\zeta_0 = \frac{1}{2}, \quad \zeta_{2M} = \frac{1}{2^M}$
2. $\zeta_k = 1 \qquad \forall\, \mathrm{k} \in [1, M]$
3. $\zeta_{2M-k} = \zeta_{2M-k+1} + 2^{-M}\binom{M}{k} \qquad \forall\, \mathrm{k} \in (0, M)$

The *Euler inversion formula* is in the form of “*general framework*” with:

- $n = 2M,$
- $\alpha_k = \beta_k,$
- $\omega_k = 10^{M/3}\eta_k$ for $\beta_k = \frac{M ln(10)}{3} + \pi i k \,\wedge\, \eta_k = (-1)^k\, \zeta_k.$

Again it is employed M instead of n. Similarly to the Gaver-Stehfest algorithm, around half of the $2M + 1$ terms appearing in the sum in the Euler algorithm are directed to accelerate the convergence using the Euler summation instead of the Salzer scheme.
The nodes β_k are evenly spaced on the vertical line $s = M ln(10)/3t$ instead than on the real axis.

Comparatively to the Gaver-Stehfest algorithm, the weights are real, plus

$$\sum_{k=0}^{n} \eta_k = 0 \qquad for\ all\ even\ n \geq 2$$

However, now the nodes are complex but when f is real valued we work with

$$Re\big(\eta_k F\,(\beta_k/t)\big) = \eta_k Re(F\,(\beta_k/t))$$

The Euler algorithm tends to be more efficient than the Gaver-Stehfest algorithm. Indeed, given M, the required system precision is only about M but it produces about $0.6M$ significant digits for good transforms. Thus the efficiency of Euler algorithm, again measured by the ratio of the significant digits produced to the precision required, is

$$eff(E) = \frac{\text{significant digits produced}}{\text{precision required}} \approx \frac{0.60\ \mathrm{M}}{1.0M} \approx 0.6$$

which is about 3/2 times that of the Gaver-Stehfest algorithm.
For further information please refer to Sakurai (2004) [15].
Now, let us summarize the procedure to use the Euler-algorithm:

1. Specify the transform $F(s)$.
2. Set $M = \lceil 1.7j \rceil$, where j is the desired number of significant digits.
3. Set the system precision equal to M.
4. Compute the weights η_k and the nodes β_k for $1 \leq k \leq 2.2M$.
5. Finally, calculate the approximant $f_e(t, M)$.

2.6.3 The Talbot algorithm

The Talbot algorithm starts from the Fourier-Mellin integral and it is built by cleverly deforming the Bromwich contour. It also holds to the "general framework" formulae.

Abate & Valko (2004) [9] reported the Talbot inversion formula for real functions:

$$\boldsymbol{f_t(t, M) = \frac{2}{5t} \sum_{k=0}^{M-1} Re(\gamma_k F\,(\beta_k/t))} \qquad (2.7)$$

where

1. $\delta_0 = \frac{2M}{5}$; $\delta_k = \frac{2k\pi}{5}\left(\cot\left(\frac{k\pi}{M}\right) + i\right) \quad \forall\ \mathrm{k} \in (0, M)$
2. $\gamma_0 = \frac{1}{2}e^{\delta_0}$; $\gamma_k = \left[1 + i\left(\frac{k\pi}{M}\right)\left(1 + \left[\cot\left(\frac{k\pi}{M}\right)\right]^2\right) - i\cot\left(\frac{k\pi}{M}\right)\right]e^{\delta_k} \quad \forall\ \mathrm{k} \in (0, M)$

The *Talbot inversion formula* is in the form of "*general framework*" with:

- $n = M,$
- $\alpha_k = \delta_k$
- $\omega_k = \frac{2}{5}\gamma_k$

For this algorithm both the weights and the nodes are complex, the space between the nodes and the weights is uneven and do not sum to zero, however their sum is small. In particular:

$$Re\left(\sum_{k=0}^{M-1} \gamma_k\right) \approx 10^{-0.6M}$$

The Talbot algorithm is as efficient as the Euler algorithm. As a matter of fact, once fixed M, nearly $0.6M$ significant digits are produced for a good transform while about M digits of precision are required.

Thus the efficiency of the Talbot algorithm is

$$eff(T) = \frac{\text{significant digits produced}}{\text{precision required}} \approx \frac{0.60\text{ M}}{1.0M} \approx 0.6$$

just as for the Euler algorithm.
For further information please refer to Murli & Rizzardi (1990) [16].
Now, let us summarize the procedure to use the Talbot-algorithm:

1. Specify the transform $F(s)$.
2. Set $M = \lceil 1.7j \rceil$, where j is the desired number of significant digits.
3. Set the system precision equal to M.
4. Compute the weights γ_k and the nodes δ_k for $0 \le k \le M - 1$.
5. Finally, calculate the approximant $f_t(t, M)$.

CHAPTER 3

LAPLACE TRANSFORM APPLIED TO OPTION PRICING

The chapter starts giving the definition and the classification of linear second order PDEs.

The BSM equation is categorized as a parabolic or hyperbolic equation. The exact placing depends on the option payoff function. These particular types of second order PDE are solved applying the Laplace transform.

Before studying the BSM equation, it will be analyzed a particular type of parabolic PDEs known as diffusion equation, which is of fundamental importance as its solution can be easily found applying the Laplace transform.

As parabolic equations can be transformed into a heat equation, the general BSM equation will be “transformed” into a diffusion equation using several change of variables. However, also in this case, the final result depends on the option payoff function.

Therefore, it is fundamental to provide a single analysis for each of the options considered:

- European option
- Barrier option
- Look-back option
- Asian option

At the end of each part, the option price is obtained by exploiting the numerical inversion presented in chapter 2.

3.1 Solving a PDE using the Laplace Transform

First, it is given a mathematical introduction to the problem. The paragraph begins with the definition of linear second order PDE's and then it presents the heat equation and its resolution. At the end, it is given the general solution to the BSM equation by exploiting the results found previously.

3.1.1 Second Order Linear PDE's

The generic form of a second-order linear equation is:

$$a \cdot y_{xx} + b \cdot y_{xt} + c \cdot y_{tt} + d \cdot y_x + e \cdot y_t + f \cdot y + g = 0$$

where

- $y(x,t)$ is the unknown function;
- x, t the independent variables;
- a, b, c, d, e, f, g are continuous functions.

Furthermore, as conic sections and quadratic forms can be categorized into parabolic, hyperbolic and elliptic based on the discriminant $b^2 - 4ac$, the same can be done with the second order PDEs. So if:

- $b^2 - 4ac > 0$ it is a hyperbolic PDE. The equation has two distinct real roots.
- $b^2 - 4ac = 0$ it is a parabolic PDE and it can be transformed into a form analogous to the diffusion equation by a change of independent variables. The equation has two equal roots.
- $b^2 - 4ac < 0$ it is an elliptic PDE whose model equation is the wave equation. The equation has no real roots.

So it is possible to categorized the BSM equation

$$F_t(t,x) + rS(t)F_x(t,x) + \frac{1}{2}\sigma^2 S^2(t) F_{xx}(t,x) = rF(t,x)$$

as a linear parabolic PDE because $b = 0$ and $a = 0$.

3.1.2 The Heat Equation

The standard diffusion formula is defined as follow

$$\frac{\boldsymbol{du(x,\tau)}}{\boldsymbol{d\tau}} = \frac{\boldsymbol{d^2u(x,\tau)}}{\boldsymbol{dx^2}} \tag{3.1}$$

The heat equation is a prototypical example of PDE and it has the following properties:

- It is linear.

  If u_1 and u_2 are solutions of the equation (3.1), then $\forall c_1, c_2: c_1u_1 + c_2u_2$ is itself a solution.
- It is of second order.
- It is parabolic

  Its characteristics[25] are given by τ (time) constant and if a change is made to u at a particular point its effect is felt instantly everywhere else.
- In general, its solutions are analytic functions of x. This is again a consequence of the fact that information propagates with infinite speed along the characteristics τ constant.

In physics, this equation describes the distribution of the variation in temperature or "heat" in a given region over time.

To find a solution to the heat equation, initial and boundary conditions must be specified. The requisites change depending on the type of region considered.

A. Finite region

Let us consider the finite interval $-L < x < L$. The information to calculate $u(x,\tau)$ is enough when:

- It is specified the temperature at the two extremes

$$u(-L,\tau) = g_-(\tau) \qquad u(L,\tau) = g_+(\tau)$$

- It is specified the heat flux at the two extremes

$$-\frac{du(-L,\tau)}{dx} = h_-(\tau) \qquad \frac{du(L,\tau)}{dx} = h_+(\tau)$$

In addition, the initial temperature is $u(x,0) = u_0(x)$ for $x\epsilon(-L,L)$.

[25] The characteristics of a second order linear PDE can be imagined as curves along which information can propagate, or as curves across which discontinuities in the second derivatives of u can occur.

B. Infinite region

With infinite region, we intend every region included in the interval $[-\infty, +\infty]$. Intuitively it is possible to say that as long as the function $u(x,\tau)$ is not allowed to grow too rapidly the solution exists, is unique, and depends continuously on the initial data $u_0(x)$. The following considerations, found in Willmott (1995) [17], can be made:

- The initial condition is given by $u(x,0) = u_0(x)$ and $u_0(x)$ is sufficiently well-behaved[26].
- $\lim_{|x|\to\infty} u_0(x)e^{-ax^2} = 0 \quad \forall a > 0$
- $\lim_{|x|\to\infty} u(x,\tau)e^{-ax^2} = 0 \quad \forall a > 0 \wedge \tau > 0$

Furthermore, the equation can be represented in two ways:

- Forward representation, which is the one considered until now. The analysis of the evolution of the temperature starts from its initial values.
- Backward representation, which starts considering the behavior of the temperature from its final value and tries to determine the temperature from which the initial distribution could have evolved.

The solution can be found using the backward form of the equation and applying the Laplace transform to it. The procedure is given immediately after. First, it is useful to summarize the initial value problem and the boundary conditions:

$$\frac{du(x,\tau)}{d\tau} = \frac{d^2u(x,\tau)}{dx^2} \qquad with \ -\infty < x < +\infty$$

$$u(x,0) = u_0(x)$$

$$\lim_{|x|\to\infty} u_0(x)e^{-ax^2} = 0 \qquad \forall a > 0$$

$$\lim_{|x|\to\infty} u(x,\tau)e^{-ax^2} = 0 \quad \forall a > 0 \wedge \tau > 0$$

As $u(x,\tau)$ is bounded the Laplace transform exists.

By applying the transform

$$\mathcal{L}\big(u_\tau(x,\tau)\big) = \mathcal{L}\big(u_{xx}(x,\tau)\big)$$

[26] It indicates any function that has no worse than a finite number of jump discontinuities.

and substituting $\mathcal{L}\big(u(x,\tau)\big) = U(x,s)$ plus applying the Derivative property (Theorem 2.6), we get

$$s \cdot U(x,s) - u_0 = U_{xx}(x,s)$$

Rewriting the equation as a non-homogeneous, second-order linear equation with constant coefficients, we obtain:

$$U_{xx}(x,s) - s \cdot U(x,s) = -u_0$$

To find a solution to this equation, we use the Variation of parameters formula theorem (see Appendix B, section B.2).

Following the procedure suggested by Theorem B.4, the first thing to do is to solve the homogeneous equation:

$$U_{xx}(x,s) - s \cdot U(x,s) = 0.$$

Using the auxiliary equation, which is equal to $r^2 - s = 0$ and calculating its roots, a solution to that equation is given by

$$U_h(x,s) = c_1 e^{\sqrt{s}\cdot x} + c_2 e^{-\sqrt{s}\cdot x}$$

where $r_1 = \sqrt{s}, r_2 = -\sqrt{s}$.

Then it is calculated $W_{U_1 U_2}$defining $U_1 = e^{\sqrt{s}\cdot x}$ and $U_2 = e^{-\sqrt{s}\cdot x}$

$$W_{U_1 U_2} = U_1 U_2' - U_2 U_1'$$

$$\begin{aligned} W_{U_1 U_2} &= -\sqrt{s} e^{\sqrt{s}\cdot x} e^{-\sqrt{s}\cdot x} - \sqrt{s} e^{-\sqrt{s}\cdot x} e^{\sqrt{s}\cdot x} \\ &= -2\sqrt{s} \end{aligned}$$

So,

$$u_1(t) = \int_{-\infty}^{+\infty} \frac{-e^{-\sqrt{s}\cdot \xi} u_0}{2\sqrt{s}} d\xi$$

$$u_2(t) = \int_{-\infty}^{+\infty} \frac{e^{\sqrt{s}\cdot \xi} u_0}{2\sqrt{s}} d\xi$$

Finally

$$U_p(x,s) = -\frac{e^{\sqrt{s}\cdot x}}{2\sqrt{s}} \left(\int_{-\infty}^{+\infty} e^{-\sqrt{s}\cdot \xi} u_0(\xi) d\xi \right) + \frac{e^{-\sqrt{s}\cdot x}}{2\sqrt{s}} \left(\int_{-\infty}^{+\infty} e^{\sqrt{s}\cdot \xi} u_0(\xi) d\xi \right)$$

$$U_p(x,s) = \frac{1}{2\sqrt{s}}\left(\int_{-\infty}^{+\infty} e^{\sqrt{s}\xi}e^{-\sqrt{s}x}u_0(\xi)d\xi - \int_{-\infty}^{+\infty} e^{-\sqrt{s}\xi}e^{\sqrt{s}x}u_0(\xi)d\xi\right)$$

$$= \frac{1}{2\sqrt{s}}\left(\int_{-\infty}^{+\infty} e^{\sqrt{s}(\xi-x)}u_0(\xi)d\xi - \int_{-\infty}^{+\infty} e^{\sqrt{s}(\xi-x)}u_0(\xi)d\xi\right)$$

$$= \frac{1}{2\sqrt{s}}\int_{-\infty}^{+\infty} e^{-\sqrt{s}|x-\xi|}u_0(\xi)d\xi$$

Thus, to get $u(x,s)$ it has to be done the following operation

$$u(x,s) = \mathcal{L}^{-1}\left(U_p(x,s)\right)$$

$$= \mathcal{L}^{-1}\left(\frac{1}{2\sqrt{s}}\int_{-\infty}^{+\infty} e^{-\sqrt{s}|x-\xi|}u_0(\xi)d\xi\right)$$

Using the linear property (Theorem 2.2) what we need to find is

$$\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}|x-\xi|}}{2\sqrt{s}}\right)$$

If $|x-\xi|$ is set equal to a we get

$$\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right)$$

and the solution is already known (see Appendix A, section A.2)

$$\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right) = \frac{e^{-a^2/4t}}{\sqrt{\pi t}}$$ [27]

So

$$\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}|x-\xi|}}{2\sqrt{s}}\right) = \frac{e^{-(x-\xi)^2/4t}}{\sqrt{4\pi t}}$$

Finally we obtain the solution of the heat equation:

$$u(x,\tau) = \int_{-\infty}^{+\infty} \mathcal{L}^{-1}\left(\frac{1}{2\sqrt{\tau}}e^{-\sqrt{\tau}|x-\xi|}\right)u_0(\xi)d\xi$$

$$= \int_{-\infty}^{+\infty} \frac{\boldsymbol{e^{-(x-\xi)^2/4\tau}}}{\boldsymbol{\sqrt{4\pi\tau}}}\boldsymbol{u_0(\xi)d\xi} \tag{3.2}$$

[27] See appendix B

3.1.3 The Black-Scholes-Merton equation

In paragraph 1.1.1 the BSM equation has been categorized as a parabolic linear second order PDE.

In this section we start turning the BSM equation into a heat equation by applying the following substitutions:

- $S = K \cdot e^x$
- $t = T - \frac{\tau}{\frac{1}{2}\sigma^2}$
- $F = K \cdot v(x, \tau)$

Let us begin with the calculation of [4]

- $\boldsymbol{\frac{dv}{d\tau}} = \frac{d\frac{F}{K}}{d\tau} = \frac{1}{K} \cdot \frac{dF}{d\tau} \cdot \frac{dt}{dt} = \frac{1}{K} \cdot \frac{dF}{dt} \cdot \frac{dt}{d\tau} = \boldsymbol{\frac{1}{K} \cdot \frac{dF}{dt} \cdot \frac{2}{\sigma^2}}$
- $\boldsymbol{\frac{dv}{dx}} = \frac{d\frac{F}{K}}{dx} = \frac{1}{K} \cdot \frac{dF}{dx} \cdot \frac{dS}{dS} = \frac{1}{K} \cdot \frac{dF}{dS} \cdot \frac{dS}{dx} = \frac{1}{K} \cdot \frac{dF}{dS} \cdot Ke^x = \boldsymbol{\frac{dF}{dS}} e^x$
- $\boldsymbol{\frac{d^2v}{dx^2}} = \left(\frac{\frac{dF}{dS}e^x}{dx}\right) = \frac{dF}{dS}e^x + \frac{d^2F}{dSdx}e^x = \frac{dv}{dx} + \frac{d^2F}{dSdx}\frac{dS}{dS}e^x = \boldsymbol{\frac{dv}{dx}} + Ke^{2x}\boldsymbol{\frac{d^2F}{(dS)^2}}$

Thus

- $\frac{dF}{dt} = F_t = \frac{dv}{d\tau} \cdot \frac{K\sigma^2}{2}$
- $\frac{dF}{dS} = F_x = \frac{dv}{dx} \cdot \frac{1}{e^x}$
- $\frac{d^2F}{(dS)^2} = F_{xx} = \left(\frac{d^2v}{dx^2} - \frac{dv}{dx}\right) \cdot \frac{1}{Ke^{2x}}$

By applying the substitutions to the original formula, we obtain:

$$\frac{dv}{d\tau} \cdot \frac{K\sigma^2}{2} + rK \cdot e^x \frac{dv}{dx} \cdot \frac{1}{e^x} + \frac{\sigma^2}{2} K^2 e^{2x} \left(\frac{d^2v}{dx^2} - \frac{dv}{dx}\right) \cdot \frac{1}{Ke^{2x}} - rK \cdot v = 0$$

$$\frac{dv}{d\tau} \cdot \frac{K\sigma^2}{2} + rK\frac{dv}{dx} + \frac{\sigma^2}{2} K \left(\frac{d^2v}{dx^2} - \frac{dv}{dx}\right) - rK \cdot v = 0$$

Since the target is to obtain a heat equation, we isolate the term $\frac{dv}{d\tau}$ and simplify the terms in common, so that we obtain

$$\frac{dv}{d\tau} = \frac{d^2v}{dx^2} + \frac{dv}{dx}\left(r\frac{2}{\sigma^2} - 1\right) - r\frac{2}{\sigma^2}v$$

To simplify, we substitute $r\frac{2}{\sigma^2}$ with k obtaining the succeeding formula

$$\frac{dv}{d\tau} = \frac{d^2v}{dx^2} + \frac{dv}{dx}(k-1) - kv$$

The equation found is more similar to the diffusion equation but it is needed another substitution: $v(x,\tau) = e^{\alpha x+\beta\tau}u(x,\tau)$, where α and β are constants defined as follows:

- $\beta = -\frac{(k+1)^2}{4}$
- $\alpha = -\frac{(k-1)}{2}$

As before, the terms are transformed individually[28].

- $\frac{dv}{d\tau} = \beta e^{\alpha x+\beta\tau}u + e^{\alpha x+\beta\tau}\frac{du}{d\tau}$
- $\frac{dv}{dx} = \alpha e^{\alpha x+\beta\tau}u + e^{\alpha x+\beta\tau}\frac{du}{dx}$
- $\frac{d^2v}{dx^2} = \alpha^2 e^{\alpha x+\beta\tau}u + \alpha e^{\alpha x+\beta\tau}\frac{du}{dx} + \alpha e^{\alpha x+\beta\tau}\frac{du}{dx} + e^{\alpha x+\beta\tau}\frac{d^2u}{dx^2}$

Once all the computation are made, the final result is

$$\frac{du(x,\tau)}{d\tau} = \frac{d^2u(x,\tau)}{dx^2}$$

which is the heat equation (3.1)and of which it is known the general solution. At this point initial and boundary conditions have to be defined bearing in mind the substitutions applied. However, the requisites change for each type of option.

[28] In the formulae the terms between round brackets are omitted in the computation to simplify the writing.

3.2 Option Pricing

In this section, we calculate the price of different type of options.

3.2.1 European option

The BSM equation for a European call option and its initial and boundary conditions are respectively:

a) $F_t(t,x) + rS(t)F_x(t,x) + \frac{1}{2}\sigma^2 S^2(t)F_{xx}(t,x) = rF(t,x)$

b) $F(0,t) = 0$

c) $F(S,t) \sim S, S \to \infty$

Then its payoff function is specified as: $F(S,T) = \max(S-K,0)$.

In paragraph 3.1.3, it has been shown how the BSM equation can be transformed into a heat equation and the solution is given by formula (3.2).

Now, it is possible to determine $u_0(\xi)$ as the precise payoff function of the option as specified a few lines above.

Considering all the changes of variables applied, we know that:

- $S = K \cdot e^x$
- $F = K \cdot v(x,\tau) \wedge v(x,\tau) = \ \ where\ \tau = \frac{1}{2}\sigma^2(T-t)$.

Then,

$$F(S,T) = \max(K \cdot e^x - K, 0) = K \cdot e^{-\frac{k-1}{2}x - \frac{k+1}{4}\tau} u(x,\tau) max(e^x - 1, 0)$$

$$u(x,\tau) = \frac{K \cdot max(e^x - 1, 0)}{K} e^{+\frac{k-1}{2}x + \frac{k+1}{4}\tau}$$

It is possible to calculate $u_0(x)$

$$u_0(x) = u_0 = \max(e^{\frac{k+1}{2}x} - e^{\frac{k-1}{2}x}, 0)$$

so formula (3.2) becomes:

$$u(x,\tau) = \int_{-\infty}^{+\infty} \frac{e^{-(x-\xi)^2/4\tau}}{\sqrt{4\pi\tau}} \max(e^{\frac{k+1}{2}\xi} - e^{\frac{k-1}{2}\xi}, 0) d\xi$$

Making another change of variable

$$x' = \frac{(\xi - x)}{\sqrt{2\tau}}$$

we obtain

$$u_0(\xi) = u_0\left(x' \cdot \sqrt{2\tau} + x\right) = \max\left(e^{\frac{k+1}{2}(x' \cdot \sqrt{2\tau}+x)} - e^{\frac{k-1}{2}(x' \cdot \sqrt{2\tau}+x)}, 0\right)$$

Finally, we get

$$u(x,\tau) = \frac{1}{2\sqrt{\pi\tau}} \int_{-x/\sqrt{2\tau}}^{+\infty} \left[e^{\frac{k+1}{2}(x' \cdot \sqrt{2\tau}+x)} - e^{\frac{k-1}{2}(x' \cdot \sqrt{2\tau}+x)}\right] \sqrt{2\tau} e^{\frac{1}{2}x'^2} dx'$$

[29]

$$= \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{+\infty} e^{\frac{k+1}{2}(x' \cdot \sqrt{2\tau}+x)} e^{-\frac{1}{2}x'^2} dx'}_{I_1} - \underbrace{\frac{1}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{+\infty} e^{\frac{k+1}{2}(x' \cdot \sqrt{2\tau}+x)} e^{-\frac{1}{2}x'^2} dx'}_{I_2}$$

$$= \quad I_1 \quad - \quad I_2$$

To simplify the resolution, we analyze the two parts separately.

Looking at I_1, we can notice that adding and subtracting the quantity $e^{\frac{1}{4}(k+1)^2\tau}$, we obtain:

$$I_1 = \frac{e^{\frac{k+1}{2}x+\frac{1}{4}(k+1)^2\tau}}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}}^{+\infty} e^{-\frac{1}{2}\left[x'+\sqrt{\frac{\tau}{2}}(k+1)\right]^2} dx'$$

If we use once again the substitution method: $\rho = x' - \frac{1}{2}(k+1)\sqrt{2\tau}$, we find

$$I_1 = \frac{e^{\frac{k+1}{2}x+\frac{1}{4}(k+1)^2\tau}}{\sqrt{2\pi}} \int_{-x/\sqrt{2\tau}-\sqrt{\tau/2}(k+1)}^{+\infty} e^{-\frac{1}{2}\rho^2} d\rho$$

If we change the sign, this integral resembles the one of the cumulative distribution function of the standard normal distribution[30].

[29] The integrating function is defined on the set $(-\infty, +\infty)$ but as $\phi(x) = \max(quantity, 0)$ then the set need to be "restricted" to $[0, +\infty)$. So after the change of variable the new extremes are:

$$\xi = 0 \rightarrow x' = -\frac{x}{\sqrt{2\tau}} \quad and \quad \lim_{\xi\to\infty} \frac{(\xi - x)}{\sqrt{2\tau}} = +\infty$$

[30] The cumulative distribution function of the standard normal distribution is

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-t^2/2} dt$$

Therefore, it is possible to rewrite I_1 as follows:

$$I_1 = -e^{\frac{k+1}{2}x+\frac{1}{4}(k+1)^2\tau}\, N(d_1)$$

where

$$d_1 = \frac{x}{\sqrt{2\tau}} + \sqrt{\frac{\tau}{2}}(k+1)$$

The procedure to obtain I_2 is the same used to get I_1, the result differs only in the sign and instead of $k+1$ there is $k-1$.

Finally, we find the solution for $u(x,\tau)$:

$$u(x,\tau) = -e^{\frac{k+1}{2}x+\frac{1}{4}(k+1)^2\tau}\, N(d_1) + e^{\frac{k-1}{2}x+\frac{1}{4}(k-1)^2\tau}\, N(d_2)$$

Lastly, we have to compute F keeping in mind all the change of variables made:

$$F = K \cdot v(x,\tau) = K \cdot e^{-\frac{k-1}{2}x-\frac{k+1}{4}\tau}u(x,\tau)$$

Making all the calculations, we get the formula for a European call option.

$$\boldsymbol{F = C(S(T),t) = S \cdot N(d_1) - K \cdot e^{-r(T-t)}N(d_2)} \qquad (3.3a)$$

The price for a European put option is computed following the same procedure. Or, in an alternatively way, it is possible to exploit the put call parity for which

$$C - P = S - Ke^{-r(T-t)}$$

Anyway, the formula for its price is

$$P(S(T),t) = Ke^{-r(T-t)}N(d_2) - S \cdot N(d_1) \qquad (3.3b)$$

3.2.2 Double Barrier Options

They are a natural extension to the “single barrier option” seen in the previous paragraph.

We proceed in a different way, considering the Laplace transform of the density functions of hitting the upper or lower barrier instead of the transformation of the option price.

Then the Bromwich integral will be used to find the analytical expression for the density functions, so no numerical inversion is needed.

Finally, for each kind of barrier options the price is calculated by integrating the option payoff with respect to the density functions.

Transition density function[31]

As said before, the underlying asset is assumed to be a Geometric Brownian motion. In addition, the dynamics of logarithm of the asset price z, under the equivalent martingale measure, is described by: $dz = \mu dt + \sigma dW$, where μ and σ are constants.

Furthermore, we hypothesize that the lower barrier is at 0 and the upper barrier at l[32], called absorbing barriers.

Let us now consider the transition density function $p(t,x;s;y)$ for $t \leq s, 0 \leq x, y \leq l$ so that it satisfies the forward and backward equations.

We begin by considering the backward equation:

$$p_t + \mu p_x + \frac{1}{2}\sigma^2 p_{xx} = 0 \tag{3.4}$$

As we want that the process gets killed if it hits one of the two barriers and that at the end of the considered period it can takes only two values, 0 or $y - x$, the equation must satisfy the following boundaries conditions:

1. $p(t,0; \dots, \dots) = p(t,l; \dots, \dots) = 0$
2. $p(s,x;s,y) = \delta(y-x)$

If we consider the forward equation:

$$-p_s - \mu p_y + \frac{1}{2}\sigma^2 p_{yy} = 0$$

the boundary conditions change as the point of view does.

There, the solution of the two equations is represented in terms of Fourier series:

$$p(t,x;s,y) = e^{\frac{\mu}{\sigma^2}(y-x)} \frac{2}{l} \sum_{k=i}^{\infty} e^{z_k(s-t)} sin\left(k\pi \frac{x}{l}\right) sin\left(k\pi \frac{y}{l}\right)$$

where

$$z_k = -\frac{1}{2}\left(\frac{\mu^2}{\sigma^2} + \frac{k^2\pi^2\sigma^2}{l^2}\right)$$

[31] See Appendix B for more information.

[32] This specification does not lose of generality since we can always shift the process by a constant such that the lower barrier is placed at 0.

Barrier densities

Now, it is fundamental to calculate the density function of hitting the upper and the lower barrier since they are used to price all the options, which have a non-zero payoff as soon as one of the barriers is hit.

Defining:

- $g^+(t,x;s)$ as the probability density function to hit firstly the upper barrier at s before hitting the lower one.
- $g^-(t,x;s)$ as the probability density function to hit firstly the lower barrier at s before hitting the upper one

and supposing that the process z can hit one of the two barriers or none, it is possible to derive the coming identity:

$$\int_t^T g^+(t,x;s)ds + \int_t^T g^-(t,x;s)ds + \int_0^l p(t,x;T,y)dy = 1 \quad \forall\, T > t$$

Nevertheless, at this point it is necessary to calculate g^+ and g^- separately.

A. Derivation of g^+

The density g^+ must satisfy the backward equation and because μ and σ are constants the density depends only on $s-t$.

To make things easy we set $\tau = s - t$ so $g^+(t,x;s) = g^+(\tau,x)$ and the backward equation becomes:

$$-g_\tau^+ + \mu g_x^+ + \frac{1}{2}\sigma^2 g_{xx}^+ \tag{3.5}$$

Moreover, the “new” boundary conditions are

1. $g^+(\tau,l) = \delta(\tau)$
2. $g^+(0,l) = \delta(l-x)$
3. $g^+(\tau,0) = 0$

To transform this second order PDE into a second order ODE, it is applied the succeeding Laplace transform

$$\gamma^+(x;\nu) = \int_0^\infty e^{-\nu\tau}\, g^+(\tau,x)d\tau \quad \forall\, \nu \geq 0$$

Then we can rewrite the equation ($\mathbf{3.5}$) as follow:

$$-\nu\gamma^{+33} + \mu\gamma_x^+ + \frac{1}{2}\sigma^2\gamma_{xx}^+ = 0 \tag{3.6}$$

and the boundary conditions become:

1. $\gamma^+(0;\nu) = 0$
2. $\gamma^+(l;\nu) = \int_0^\infty e^{-\nu\tau}\delta(\tau) = 1$

We know how to solve this kind of equation. Using the procedure explained in Appendix B:

1. Solve the characteristic equation

$$\frac{1}{2}\sigma^2 r^2 + \mu r - \nu = 0$$

$$r_{1,2} = \frac{-\mu \pm \sqrt{\mu^2 + 2\sigma^2\nu}}{\sigma^2} \quad where \;\; \theta(\nu) = \frac{\sqrt{\mu^2 + 2\sigma^2\nu}}{\sigma^2}$$

Because the roots are complex numbers where

$$\alpha = -\frac{\mu}{\sigma^2} \qquad \beta = \theta(\nu)$$

the solution is

$$\gamma^+(x;\nu) = e^{-\frac{\mu}{\sigma^2}x}(c_1\sinh(\theta(\nu)x) + c_2\cosh(\theta(\nu)x)) \tag{3.7}$$

2. Solve the boundary conditions problem
   i. $\gamma^+(0;\nu) = 1\,(0 + c_2\cosh(0)) = c_2 = 0$
   ii. $\gamma^+(l;\nu) = e^{-\frac{\mu}{\sigma^2}l}c_1\sinh(\theta l) = 1 \;\; \rightarrow \;\; c_1 = \frac{e^{\frac{\mu}{\sigma^2}l}}{\sinh(\theta l)}$
3. Finally the solution is

$$\gamma^+(x;\nu) = e^{-\frac{\mu}{\sigma^2}x}\left(e^{\frac{\mu}{\sigma^2}l}\frac{\sinh(\theta(\nu)x)}{\sinh(\theta(\nu)l)}\right)$$

[33] The differentiation of the Laplace Transform gives:

$$\frac{d}{d\nu}\gamma^+(x;\nu) = \int_0^\infty \left[\frac{d}{d\nu}e^{-\nu\tau}\right]g^+(\tau,x)d\tau = -\nu\int_0^\infty e^{-\nu\tau}\,g^+(\tau,x)d\tau = -\nu\gamma^+(x;\nu)$$

At this point, we have to invert the transform, and this is done using the Bromwich integral. As showed in chapter 3, par. 2.5, we start from the Fourier-Mellin inversion formula:

$$g^{+}(t,x;s)=\frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{\tau z}\gamma^{+}(x;\nu)dz$$

We define the contour integral as follow:

$$\frac{1}{2\pi i}\oint_{C} e^{\tau z}\gamma^{+}(x;\nu)dz$$

From the theory, we know that the Cauchy Residue Theorem can determine the value of the contour integral (see Appendix A, A.5).

Therefore, we need to find the singularities of the function keeping in mind that they exists only if the term in the denominator of γ^{+} is equal to zero

$$\gamma^{+}(x;z)=e^{-\frac{\mu}{\sigma^2}x}\left(e^{\frac{\mu}{\sigma^2}l}\frac{\sinh(\theta(z)x)}{\sinh(\theta(z)l)}\right)$$

$$\sinh(\theta(z)l)=0$$

From the complex number theory, we know that

$$\sinh(z)=-isin(iz)$$

Then, in this case

$$-\mathrm{isinh}(i\theta(z_k)l)=0$$

This is true only when

$$i\theta(z_k)l=k\pi \;\; k\in\mathbb{N}$$

$$k\pi=i\frac{\sqrt{\mu^2+2\sigma^2 z_k}}{\sigma^2}$$

$$z_k=-\frac{1}{2}\left(\frac{\mu}{\sigma^2}+\frac{k^2\pi^2\sigma^2}{l^2}\right)$$

So, for the Cauchy Theorem

$$g^{+}(\tau,x)=Res(z_k)$$

$$Res(z_k)=\lim_{z\to z_k} e^{\tau z}e^{-\frac{\mu}{\sigma^2}x}\left(e^{\frac{\mu}{\sigma^2}l}\sinh(\theta(z)x)\frac{z-z_k}{\sinh(\theta(z)l)}\right)$$

Knowing that $\frac{d}{dx}\sinh(x) = \cosh(x)$ the limit can be rewritten as follow

$$\lim_{z \to z_k} e^{\tau z} e^{\frac{\mu}{\sigma^2}(l-x)} \left(\sinh(\theta(z)x) \frac{1}{\cosh(\theta(z)l) \frac{d\theta}{dz} l} \right)$$

which gives

$$e^{\tau z_k} e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} (-1)^k k\pi i \sinh\left(k\pi i \frac{x}{l}\right) = e^{\tau z_k} e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} k\pi \sin\left(k\pi \frac{l-x}{l}\right)$$

Thus, summing up all the residues gives the density function of hitting the upper barrier

$$g^+(t,x;s) = e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} e^{(s-t)z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right)$$

B. Derivation of g^-

We start by applying the Laplace transform to $g^+(\tau, x)$ and the backward equation and the solution of γ^- are equal to formulae (3.5) and(3.6).

What changes are the boundary conditions, that become:

1. $\gamma^-(0;v) = 1$
2. $\gamma^-(l;v) = 0$

Now we have to find the general solution to the backward equation of γ^- which means solving the boundary condition problem:

$$\gamma^-(x,v) = e^{-\frac{\mu}{\sigma^2}x}(c_1 \sinh(\theta(v)x) + c_2 \cosh(\theta(v)x))$$

➔ $\gamma^-(0,v) = c_2 = 1$

➔ $\gamma^-(l,v) = \cosh(\theta(v)l) + c_1 \sinh(\theta(v)l) = 0$ ➔ $c_1 = -\frac{\cosh(\theta(v)l)}{\sinh(\theta(v)l)}$

then

$$\gamma^-(x,v) = e^{-\frac{\mu}{\sigma^2}x} \left(\cosh(\theta(v)x) - \frac{\cosh(\theta(v)l)\sinh(\theta(v)x)}{\sinh(\theta(v)l)} \right)$$

it can be simplified by considering the identity for which

$$\sinh(a-b) = \sinh(a)\cosh(b) - \cosh(a)\sinh(b)$$

It is quiet immediate to see that in our case

$$\sinh(\theta(v)(l-x)) = \sinh(\theta(v)l)\cosh(\theta(v)x) - \cosh(\theta(v)l)\sinh(\theta(v)x)$$

where

$$\frac{\sinh(\theta(v)(l-x))}{\sinh(\theta(v)l)} = \cosh(\theta(v)x) - \frac{\cosh(\theta(v)l)\sinh(\theta(v)x)}{\sinh(\theta(v)l)}$$

Therefore, the general solution is

$$\gamma^-(x,v) = \; e^{-\frac{\mu}{\sigma^2}x}\left(\frac{\sinh(\theta(v)(l-x))}{\sinh(\theta(v)l)}\right)$$

It is important to see that there is a relationship between γ^- and γ^+:

$$\gamma^-(x,v) = e^{-2\frac{\mu}{\sigma^2}x}\,\gamma^+(l-x;v)$$

Hence, by substitution we get that

$$g^-(t,x;s) = e^{\frac{\mu}{\sigma^2}(l-x)}\frac{\sigma^2}{l^2}\sum_{k=1}^{\infty} e^{(s-t)z_k} k\pi \sin\left(k\pi\frac{x-l}{l}\right)$$

Once we have calculated the two barrier densities it is possible to price various type of double barrier options.

Pricing of different types of double barrier options

Before examining in deep the argument, it is fundamental to make a list of the economical assumption made.

In general, let r_F be the risk free interest rate , q the dividend yield, and σ the volatility exchange rate.

Here, the process z is defined as $log\frac{S(t)}{L}$ and $l = \log\left(\frac{U}{L}\right)$. Furthermore the drift term under the equivalent martingale measure is

$$\mu = r_F - q - \frac{1}{2}\sigma^2 \text{ [34]}$$

In the end, it is important to say that barrier options can be split into two big families:

- Knock-in options when the contract is exercised as the underlying asset hit the specified barrier prices.
- Knock-out options if the contract expires when the underlying asset hit the specified barrier prices.

Both groups can be divided into subgroups.

[34] Notice that the formula holds also if instead of r_F and q we have respectively the domestic interest rate and the foreign interest rate.

Now we analyze different types of knock-out options and only the general case of knock-in options.

Knock-out options

It is the simplest kind of double barrier options as at maturity they pay a constant amount.

These are the possible records:

- K_U if the upper barrier is the first to be hit,
- K_L if the lower barrier is the first to be hit,
- K if neither barrier is hit during the option life.

The value of this option is given by

$$V_{CPM}(t) = e^{-r_d(T-t)}\left(K_U p^+(T) + K_L p^-(T) + K\big(1 - p^+(T) - p^-(T)\big)\right) \quad (3.8)$$

where $p^+(T)$ and $p^-(T)$ are the probability of hit first the upper or the lower barrier before T, respectively.

To find the two probabilities we have to integrate over the barrier densities. To facilitate the writing we can use a single formula that indicates at the same time both the probabilities:

$$p^{\pm}(T) = \int_t^T g^{\pm}(t,x;s)ds = \int_t^{\infty} g^{\pm}(t,x;s)ds - \int_T^{\infty} g^{\pm}(t,x;s)ds$$

Remembering that

$$\gamma^{\pm}(x;\nu) = \int_0^{\infty} e^{-\nu\tau}\, g^{\pm}(\tau,x)d\tau \quad \forall\, \nu \geq 0$$

If $v = 0$, consequentely

$$\gamma^{\pm}(x;0) = \int_0^{\infty} g^{\pm}(t,x;s)d(s-t) \; = \int_t^{\infty} g^{\pm}(t,x;s)ds \quad \forall\, \nu \geq 0$$

Therefore,

$$\int_t^T g^{\pm}(t,x;s)ds = \gamma^{\pm}(x;0) - \int_T^{\infty} g^{\pm}(t,x;s)ds$$

Firstly, we analyze $p^+(T)$:

$$\gamma^+(x;0) = e^{\frac{\mu}{\sigma^2}(l-x)}\left(\frac{\sinh(\theta(0)x)}{\sinh(\theta(0)l)}\right)$$

but

$$\theta(0) = \frac{\sqrt{\mu^2 + 2\sigma^2 0}}{\sigma^2} = \frac{\mu}{\sigma^2}$$

So,

$$\gamma^+(x;0) = e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sinh\left(\frac{\mu}{\sigma^2} x\right)}{\sinh\left(\frac{\mu}{\sigma^2} l\right)}$$

and

$$\int_T^\infty g^+(t,x;s)ds = \int_T^\infty e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^\infty e^{(s-t)z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) ds$$ [35]

$$= e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^\infty k\pi \sin\left(k\pi \frac{l-x}{l}\right) \int_T^\infty e^{(s-t)z_k} ds$$

$$= e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^\infty \frac{e^{(T-t)z_k}}{z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right)$$

Finally the probability p^+ is

$$p^+(T) = e^{\frac{\mu}{\sigma^2}(l-x)} \left[\frac{\sinh\left(\frac{\mu}{\sigma^2} x\right)}{\sinh\left(\frac{\mu}{\sigma^2} l\right)} - \frac{\sigma^2}{l^2} \sum_{k=1}^\infty \frac{e^{(T-t)z_k}}{z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \right]$$

The probability p^- is equal to

$$p^-(T) = e^{\frac{\mu}{\sigma^2}(l-x)} \left[\frac{\sinh\left(\frac{\mu}{\sigma^2} (l-x)\right)}{\sinh\left(\frac{\mu}{\sigma^2} l\right)} - \frac{\sigma^2}{l^2} \sum_{k=1}^\infty \frac{e^{(T-t)z_k}}{z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \right]$$

It is obtained using the same procedure explained just before.

At this point, it is only needed to put the found formulae into equation (3.8) to find the value of the option.

These option barriers provide for a refund when one of the barrier is hit.

[35] Remember that for the fundamental theorem we have:

$$\int_a^b \sum_i f_i(x) = \sum_i \int_a^b f_i(x)$$

We first consider the case of hitting the upper barrier. The value of the option is given by the probability to hit the barrier multiply by the fixed reward, R_U:

$$V_{RAHU} = R_u \int_t^T e^{-r_d(s-t)} g^+(t,x;s)ds$$

Developing the formula

$$V_{RAHU}(t) = R_u \int_t^T e^{-r_d(s-t)} e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} e^{(s-t)z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) ds$$

$$= R_u e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \int_t^T e^{-(r_d - z_k)(s-t)}\, ds$$

To solve the integral, we only have to find the primitive for terms of the form $e^{(z_k - r_d)(s-t)}$. In addition, adopting the following notation, the computations are simplified:

$$r_d - z_k = \frac{1}{2}\left(\frac{\mu'^2}{\sigma^2} + \frac{k^2\pi^2\sigma^2}{l^2}\right) = -z'_k$$

where

$$\mu' = \sqrt{\mu^2 + 2\sigma^2 r_d}$$

In addition, we indicate with g'^+ as the barrier density with drift μ'.

Making the substitution:

$$V_{RAHU}(t) = R_u e^{\frac{\mu}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \int_t^T e^{z'_k(s-t)}\, ds$$

Multiplying and dividing for $e^{\frac{\mu\prime}{\sigma^2}(l-x)}$

$$V_{RAHU}(t) = R_u e^{\frac{\mu-\mu\prime}{\sigma^2}(l-x)} \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} e^{\frac{\mu\prime}{\sigma^2}(l-x)} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \int_t^T e^{z'_k(s-t)}\, ds$$

Reminding the integral property used before

$$V_{RAHU}(t) = R_u e^{\frac{\mu-\mu\prime}{\sigma^2}(l-x)} \int_t^T \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} e^{z'_k(s-t)} e^{\frac{\mu\prime}{\sigma^2}(l-x)} k\pi \sin\left(k\pi \frac{l-x}{l}\right) ds$$

$$= R_u e^{\frac{\mu-\mu'}{\sigma^2}(l-x)} \int_t^T g'^+(t,x;s)\, ds$$

$$V_{RAHU}(t) = R_u e^{\frac{\mu}{\sigma^2}(l-x)} \left[\frac{\sinh\left(\frac{\mu'}{\sigma^2}x\right)}{\sinh\left(\frac{\mu'}{\sigma^2}l\right)} - \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} \frac{e^{(T-t)z_k}}{z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \right]$$

Identically, we find the value of the amount R_l.

$$V_{RAHL}(t) = R_l e^{-\frac{\mu}{\sigma^2}(l-x)} \left[\frac{\sinh\left(\frac{\mu}{\sigma^2}(l-x)\right)}{\sinh\left(\frac{\mu}{\sigma^2}l\right)} - \frac{\sigma^2}{l^2} \sum_{k=1}^{\infty} \frac{e^{(T-t)z_k}}{z_k} k\pi \sin\left(k\pi \frac{l-x}{l}\right) \right]$$

<u>Double knock-out</u>

The payoff of a double knock-out call option, supposing neither barrier is hit during the life option, is equal to $\max\{S(T) - K, 0\}$. Therefore, we can express the value at t of a double knock-out option as follow:

$$V_{DKOC}(t) = e^{-r_d(T-t)} \int_0^l \max\{S(T) - K, 0\}\, p(t, x; T, y) dy$$

To simplify the calculations we are going to use the following notation

$$y = \log\left(\frac{S(T)}{L}\right) \rightarrow Le^y = S(T) \rightarrow y > \log\left(\frac{K}{L}\right) = d$$

Hypothesizing that $0 \le d \le l$, we obtain

$$V_{DKOC}(t) = e^{-r_d(T-t)} \int_d^l (Le^y - K) p(t, x; T, y) dy$$

$$= e^{-r_d(T-t)} \left[L \int_d^l e^y p(t, x; T, y) dy - K \int_d^l p(t, x; T, y) dy \right]$$

Analyzing separately the two integrals (omitting $(t, x; T, y)$ after p):

$$\int_d^l e^y p\, dy = \frac{2}{l} \sum_{k=i}^{\infty} e^{z_k(T-t)} e^{-\frac{\mu}{\sigma^2}x} sin\left(k\pi \frac{x}{l}\right) \int_d^l e^{y\left(1+\frac{\mu}{\sigma^2}\right)} \sin\left(k\pi \frac{y}{l}\right) dy \qquad (3.9)$$

$$\int_d^l p dy = \frac{2}{l} \sum_{k=i}^{\infty} e^{z_k(T-t)} e^{-\frac{\mu}{\sigma^2}x} sin\left(k\pi \frac{x}{l}\right) \int_d^l e^{y\frac{\mu}{\sigma^2}} \sin\left(k\pi \frac{y}{l}\right) dy \qquad (3.10)$$

Both integrals involve finding the primitive for terms of the form

$$e^{\alpha y} sin(\beta y)$$

The primitive is found using the integration by parts:

$$\int e^{\alpha y} sin(\beta y) dy = -e^{\alpha y}\frac{\cos(\beta y)}{\beta} + \int \alpha e^{\alpha y}\frac{\cos(\beta y)}{\beta} dy$$

$$= -e^{\alpha y}\frac{\cos(\beta y)}{\beta} + \frac{\alpha}{\beta}\int e^{\alpha y}\cos(\beta y)\, dy$$

where the integral $\int e^{\alpha y}\cos(\beta y)\, dy$ is also solved integrating by part

$$\int e^{\alpha y}\cos(\beta y)\, dy = e^{\alpha y}\frac{\sin(\beta y)}{\beta} - \int \alpha e^{\alpha y}\frac{\sin(\beta y)}{\beta} dy$$

so

$$\int e^{\alpha y} sin(\beta y) dy = e^{\alpha y}\frac{\cos(\beta y)}{\beta} + \frac{\alpha}{\beta}\left(e^{\alpha y}\frac{\sin(\beta y)}{\beta} - \frac{\alpha}{\beta}\int \alpha e^{\alpha y}\frac{\sin(\beta y)}{\beta} dy\right)$$

$$\left(1+\frac{\alpha^2}{\beta^2}\right)\int e^{\alpha y} sin(\beta y) dy = e^{\alpha y}\left(\frac{\alpha}{\beta}\sin(\beta y) - \frac{1}{\beta}\cos(\beta y)\right)$$

$$\int e^{\alpha y} sin(\beta y) dy = \frac{e^{\alpha y}\left(\frac{\alpha}{\beta}\sin(\beta y) - \frac{1}{\beta}\cos(\beta y)\right)}{\left(1+\frac{\alpha^2}{\beta^2}\right)}$$

$$= e^{\alpha y}\frac{\left(\frac{\alpha\sin(\beta y) - \beta\cos(\beta y)}{\beta^2}\right)}{\left(\frac{\alpha^2+\beta^2}{\beta^2}\right)}$$

$$= e^{\alpha y}\frac{(\alpha\sin(\beta y) - \beta\cos(\beta y))}{\alpha^2+\beta^2}$$

Exploiting this result, we can find a common representation for the two integral (3.9) and (3.10)

$$Q(\alpha, y) = \frac{2}{l} e^{\frac{\mu}{\sigma^2}(y-x)} e^{\alpha y}\sum_{k=i}^{\infty} e^{z_k(T-t)} sin\left(k\pi\frac{x}{l}\right)\frac{\left(\frac{\mu}{\sigma^2}+\alpha\right)\sin\left(k\pi\frac{y}{l}\right) - \frac{k\pi}{l}\cos\left(k\pi\frac{y}{l}\right)}{\left(\frac{\mu}{\sigma^2}+\alpha\right)^2 + \left(\frac{k\pi}{l}\right)^2}$$

By doing the appropriate substitution, the double knock-out call can be expressed as:

$$V_{DKOC}(t) = e^{-r_d(T-t)}\left[L\big(Q(1,l) - Q(1,d)\big) - K\big(Q(0,l) - Q(0,d)\big)\right]$$

To calculate the value of a put is the same.

Knock-in options

The contract gives the right to the owner to have a plain vanilla option when one of the fixed barrier is hit.

Let us suppose that the contract provide for:

- a call option if the upper barrier is hit first,
- a put option if the lower barrier is hit first.

The value of the barrier option is equal to

$$V_{DKI}(t) = \int_t^T e^{-r_d(T-t)} C(S;U) g^+(t,x;s) ds + \int_t^T e^{-r_d(T-t)} P(S;L) g^-(t,x;s) ds$$

This integral cannot be solved analytically but using a numerical integration, such as quadrature methods.

3.2.3 Lookback Options

In this section we want to introduce an alternative solution to the classical one proposed by the literature for pricing lookback options. In particular, we provide an alternative pricing method for floating strike lookback options based on the Laplace Transform.

All the paragraph is based on Kimura (2007) [27].

First, let us suppose that S_t is the price process and it is monitorized continuously. Then we can define the following two quantities:

- $m_t = \min_{0 \le u \le t} S_u$
- $M_t = \max_{0 \le u \le t} S_u$

Now, we can define the payoff for the two types of lookback options:

- fixed strike lookback
  Call:$\max(M_T - K, 0)$
  Put: $\max(K - m_T, 0)$
- floating strike lookback, whose payoff is equal to
  Call: $\max(S_T - m_T, 0)$
  Put: $\max(M_T - S_T, 0)$

It is possible to see that the payoff for the floating strike lookback are in any case positive or at least equal to 0. For this reason, their premium is very high and this makes them not appealing for investors. Conze and Viswanathan

(1991)[28] introduced a more general definition for floating lookback, called fractional lookback, whose payoff is defined as:

- Call: $\max(S_T - \alpha m_T, 0)$
- Put: $\max(\beta M_T - S_T)$

where $\alpha \geq 1$ and $\beta \in (0,1]$.

It is immediate to see that we can obtain the standard floating strike lookback setting $\alpha = \beta = 1$.

At this point, we give the analytic formula for pricing floating lookback options.[36] First, we give a different definition for m_t:

$$m_t = m \wedge \inf_{0 \leq u \leq t} S_u \quad t \geq 0$$

Furthermore, we indicate the value of a floating strike lookback call at $t \in [0,T]$ with $LB_{fl}(t,x,m)$.

Now, we can express the value of the option as the solution to the partial differential equation of the BSM model, reported in Chapter 1 formula (1.a) and (1.b).

$$LBc_{fl}(t,x,m) = xe^{-\delta(T-t)}\Phi(d_1^+) - \alpha m e^{-r(T-t)}\Phi(d_1^-) + \begin{cases} \dfrac{\alpha x}{\gamma}\left\{e^{-r(T-t)}\left(\dfrac{m}{x}\right)^{\gamma}\Phi(d_2^+) - e^{-\delta(T-t)}\alpha^{\gamma}\Phi(d_2^-)\right\} & r \neq \delta \\ \alpha x e^{-r(T-t)}\sigma\sqrt{T-t}\left(d_2^+\Phi(d_2^+) + \phi(d_2^+)\right) & r = \delta \end{cases} \tag{3.11}$$

where:

- Φ is the cumulative distribution function,
- ϕ is the probability distribution function,
- $\gamma = \frac{2(r-\delta)}{\sigma^2}$
- $d_1^{\pm} = \frac{1}{\sigma\sqrt{T-t}}\left\{log\left(\frac{x}{\alpha m}\right) + \left(r - \delta \pm \frac{1}{2\sigma^2}\right)(T-t)\right\}$
- $d_2^{\pm} = \frac{1}{\sigma\sqrt{T-t}}\left\{log\left(\frac{m}{\alpha x}\right) + \left(r - \delta \pm \frac{1}{2\sigma^2}\right)(T-t)\right\}$

[36] We present only the procedure for a call floating strike lookback option as the procedure for a put is identical.

However, the value given by $LB_{fl}(t,x,m)$ depends on the time only through the time $T-t$. For convenience, we introduce the time-reversed values as:

$$\widetilde{LB}c_{fl}(\tau,x,m) = \quad \tau \geq 0$$

The time-reversed price can be obtained applying the Laplace-Carson transformation[37]:

$$LBc_{fl}^{*}(\lambda,x,m) = \mathcal{LC}\left[LB_{fl}(T-\tau,x,m)\right]$$

The result is reported in the following theorem.

Theorem 3.1: Price of a Floating Lookback Call Option

$$LBc_{fl}^{*}(\lambda,x,m) = \begin{cases} \varphi_1(\lambda,x,m) & m \leq x < \alpha m \\ \varphi_2(\lambda,x,m) + \dfrac{\lambda x}{\lambda+\delta} - \dfrac{\lambda \alpha m}{\lambda + r} & x \geq \alpha m \end{cases} \tag{3.12}$$

where

- $\varphi_1(\lambda,x,m) = \frac{Sv_1}{v_1-v_2}\left(\frac{\lambda}{\lambda+\delta} + \frac{1-v_1}{v_1}\frac{\lambda}{\lambda+r}\right)\left\{\left(\frac{\alpha m}{x}\right)^{v_2} - \alpha^{v_2-v_1}\frac{v_2}{v_1}\left(\frac{\alpha m}{x}\right)^{v_1}\right\}$
- $\varphi_2(\lambda,x,m) = \frac{Sv_2}{v_1-v_2}\left\{\frac{\lambda}{\lambda+\delta} + \frac{1-v_2}{v_2}\frac{\lambda}{\lambda+r} - \alpha^{v_2-v_1}\left(\frac{\lambda}{\lambda+\delta} + \frac{1-v_1}{v_1}\frac{\lambda}{\lambda+r}\right)\right\}\left(\frac{\alpha m}{x}\right)^{v_1}$

and the parameters v_1, v_2 are two real root of the quadratic equation:

$$\frac{1}{2}\sigma^2 v^2 + \left(\delta - r - \frac{1}{2}\sigma^2\right)v - (\lambda+\delta) = 0$$

$$v_1, v_2 = \frac{1}{\sigma^2}\left\{-\left(\delta - r - \frac{1}{2}\sigma^2\right) \pm \sqrt{\left(\delta - r - \frac{1}{2}\sigma^2\right)^2 + 2\sigma^2(\lambda+\delta)}\right\}$$

For a demonstration of the theorem see Kimura (2007) [27].

Now, we only need to apply one of the inversion numerical method to find the solution.

---

[37] The Laplace-Carson it is a variation of the Laplace transform. It takes the name from Carson who devised the original formula to make the transform of the Heaviside step function a function whose Laplace transform is constantly equal to 1. For more details see Kimura (2007)[27].

For completeness, we give also the theorem for the price of a fractional floating lookback put option, whose payoff is defined as $\max(\beta M_T - S_T)$ where $\beta \in (0,1]$ and $M_t = M \vee \sup_{0\le u\le t} S_u\,, t \ge 0$.

Theorem 3.1: Price of a Floating Lookback Call Option

$$LBp_{fl}^*(\lambda, x, M) = \begin{cases} \varphi_1(\lambda, x, M) \ + \dfrac{\lambda x}{\lambda+\delta} - \dfrac{\lambda \beta M}{\lambda + r} & 0 \le x \le \beta M \\ \varphi_2(\lambda, x, M) & \beta M < x \le M \end{cases}$$

where

- $\varphi_1(\lambda, x, M) = \frac{Sv_1}{v_1-v_2}\left\{\frac{\lambda}{\lambda+\delta} + \frac{1-v_1}{v_1}\frac{\lambda}{\lambda+r} - \ \beta^{v_1-v_2}\left(\frac{\lambda}{\lambda+\delta} + \frac{1-v_2}{v_2}\frac{\lambda}{\lambda+r}\right)\right\}\left(\frac{\beta M}{x}\right)^{v_2}$
- $\varphi_2(\lambda, x, M) = \frac{Sv_2}{v_1-v_2}\left(\frac{\lambda}{\lambda+\delta} + \frac{1-v_2}{v_2}\frac{\lambda}{\lambda+r}\right)\left\{\left(\frac{\beta M}{x}\right)^{v_1} - \ \beta^{v_1-v_2}\frac{v_1}{v_2}\left(\frac{\beta M}{x}\right)^{v_2}\right\}$

and the parameters v_1, v_2 are two real root of the quadratic equation:

$$\frac{1}{2}\sigma^2 v^2 + \left(\delta - r - \frac{1}{2}\sigma^2\right) v - (\lambda + \delta) = 0$$

$$v_1, v_2 = \frac{1}{\sigma^2}\left\{-\left(\delta - r - \frac{1}{2}\sigma^2\right) \pm \sqrt{\left(\delta - r - \frac{1}{2}\sigma^2\right)^2 + 2\sigma^2(\lambda+\delta)}\right\}$$

For a demonstration of the theorem see Kimura (2007) [27].

3.2.4 Asian Option

There exists two types of Asian Options:

- Average price option
  The final price is the mean of the values taken by the underlying asset over a time interval $[0, T]$
- Average strike option
  The strike price is based on the average of the spot price over a time interval $[0, T]$.

They can also be distinguished by:

- Time to Maturity

- Sampling design (discrete or continuous)
- Averaging method used (arithmetic or geometric)
- Weight assigned to each price

In this paper only continuous arithmetic Asian option, with weights equal for each price and time period of the sampling design equal to the entire life of the option, will be considered. So, we will consider Asian option with the following payoff:

$$\max\left\{\frac{1}{T}\int_{i=1}^{T} S_i - K, 0\right\}. \tag{3.13}$$

In the next chapter, we are going to find the price for a continuous arithmetic Asian call option using the Laplace Transform and inverting the found formula using the Euler inversion algorithm.

Notice that we are going to consider only call Asian options[38] as exploiting the following formula, which is the well-known put-call parity, it is possible to get the price of a put Asian option:

$$\begin{aligned} &CAA_{call}(K, S_0, r, q, T) - CAA_{put}(K, S_0, r, q, T) \\ &\quad = \frac{1}{(r-q)T}\left(e^{-qT} - e^{-rT}\right)S_0 - e^{-rT}K \end{aligned} \tag{3.14}$$

Since the procedure to price Asian options needs the explanation of new concepts, we prefer to discuss the topics in a single chapter, see Chapter 5.

[38] From now on, we indicates the continuous arithmetic Asian option as Asian option.

CHAPTER 4

NUMERICAL ANALYSIS OF INVERSION ALGORITHMS

In Chapter 2 we have presented the theory of the Laplace Transform as it is an efficient method to convert differential problems, such as the option pricing, into algebraic ones of easier resolution.

After presenting the theory underlying the Laplace Transform, we discussed about its inversion, as we have to pass from the domain of the transform to the domain of the time to finally get the solution of the studied differential problem, see Figure 2.1.

The complex inversion formula, see theorem 2.16, provides for a complex integral that allows to come back in the domain of time; unfortunately this integral cannot always be solved using analytical methods, such as the one based on Bromwich integral and Residue Theorem, see Appendix A for further details about the mentioned thoerems. Therefore, we have presented three inversion algorithms (the Gaver-Stefhest, the Euler and the Talbot algorithm) that provide for an approximation of the integral by finite linear combinations of the transformed values.

The three inversion methods have been presented in a general framework, see formula 2.4, that allows us to get automatically the final result setting only two parameters: the time t and the coefficient M.

What we are going to do in the first part of this chapter is to test the accuracy and the efficiency of these three inversion algorithms.

In the literature, most of the references of the mentioned problem are mostly concerned about studying the problem from the point of view of the pure mathematician and therefore they are scarcely interested in the practical implications, which may be those useful for finance. The main references are Josso and Larsen (2012) [29], Abate and Whitt (2006) [30] and McClure (2012) [31].

In particular, what the quoted authors are coming to calculate are the values of the number of the terms to sum in each algorithm, indicated with M, which they define as "good guess". The authors themselves specify that the generality of these findings represents also their practical limit, but this does not mean that they cannot be use as starting points for specific cases. That is exactly what we are going to do in this chapter:

- we start by inverting a Laplace transform with the three inversion algorithms and analyzing their performance varying M and t. To study the behavior of the three inversion routines for different values of the time, we need to fix the value of the parameter M and it is in this moment that we will look at the literature for the suggested values of M;
- then, we are going to use the three inversion algorithms to invert the Laplace transform of a floating lookback call option price, see formula 3.12 and to see if they can provide for a good estimation (see below for its definition).

We start by studying two types of Laplace transform: the category of “good” Laplace transforms, whose all singularities lie on the nonnegative real axis (indicated also as transforms of class $\mathbb{F}$) and the category of transforms whose singularities lies out of the real axis, which are indicated as Laplace transform of class $\mathbb{G}$. The fundamental difference between these two types of Laplace transform is that for the first category the accuracy of the estimation, obtained using the inversion algorithms, depends only on the parameter M as it is independent from the value of the time; on the contrary, for the second category the accuracy depends also on the time. The aim is to consider the largest record of cases.

Talking about the accuracy, we want to specify immediately that we consider four digits of accuracy to get a “good” estimation. The criteria has been chosen looking at the practical aspect of this thesis. As what we want, is to estimate the price of an option, we think that the chosen number of digits are sufficient to get a good approximation.

Furthermore, we want to highlight the fact that for the inversion of Laplace transforms of class $\mathbb{G}$ we will consider only low values for $t, (t \leq 10)$. Again, the choice has been conditioned by the specific problem we are working on in this thesis: the pricing of the options, options whose lifetime is not usually higher than one year.

However, we have thought that a brief and not detailed analysis for higher values of t ($t = 1{:}250)$ is needed to make our work complete.

To do what said so far, we have taken five different Laplace Transforms whose inverse is known and we have compared the real value of the inverse, given by the known inverse formula, with the approximations obtained by the three inversion algorithms.

In the last part of the chapter, we have used the result obtained in chapter 3 for lookback options in order to compare the estimation of its price gained by the three inversion algorithms and the price provided by the closed-form valuation formula calculated by Conze and Viswanathan (1991) [39] . Furthermore, we have compared the two mentioned methods with the approximations provided by Cox-Ross-Rubistein binomial tree model and Monte Carlo simulation, which are two of the most used estimation methods in the literature.

The reason why we have chosen lookback options is given by the complexity of the application of the analytical inversion method to their Laplace Transform found in Chapter 3, see formula 3.11.

In fact, in the previous chapter we have applied in different ways the Laplace transform to price several types of options. Fortunately, we have been able to invert the Laplace transforms of European and Barrier options exploiting the Residue theorem so it was possible to provide an analytical formula for the pricing of both options.

[39] For more details about the closed-form valutation formula of lookback options see Conze and Viswanathan (1991).

All the empirical results has been calculated using MATLAB codes in MATLAB R2015b for academic use on a SONY VAIO with a processor Intel® Core™ i5-2450M CPU @ 2.50GHz and 4.00 GB of RAM.

4.1 A comparison among Gaver-Stefhest, Euler and Talbot inversion algorithm

Our starting point is formula 2.4, reported in Chapter 2, which represents the general framework of the three inversion algorithms. For each inversion algorithm it has been defined a precise equation:

- for Gaver-Stefhest Inversion Algorithm see formula 2.5;
- for Euler Inversion Algorithm, see formula 2.6;
- for Talbot Inversion Algorithm, see formula 2.7.

From the theory, we know that, once we have decided at which instant t we want to make the inversion, for all the three algorithms the only parameter we have to set is M, which represents:

- half of the number of terms to sum in the estimation, in the GS algorithm,
- half of the number of terms to sum in the estimation, in the Euler algorithm,
- the number of terms to sum in the estimation, in the Talbot algorithm.

As we have mentioned, Abate and Valko (2004) [9] have identified two classes of different Laplace transforms: $\mathbb{F}$ and $\mathbb{G}$. The transforms that belong to the first class are usually called good Laplace transforms, for their definition see note 24 chapter 2, and they have the characteristic that the relative error produced by the estimation of the three algorithms are almost independent on the value of t[40]. The second type of transforms, are transforms whose relative error[41] is less accurate if we fix M and increase t.

Taking advantage of the properties of this particular Laplace transforms we want to demonstrate the effectiveness of the three different algorithms.

[40] For further details see Abate and Valko (2004) [9]

[41] We mean the relative error calculated by the difference between the estimation of the inverse of a Laplace Transform using the three inversion algorithms and the real value of the known inverse.

As first thing, we want to study the behavior of the tree algorithms for different values of M.

In order to do that, we have taken three good Laplace transform, whose inverse is known, and we have compared the inversion routine estimations and the real value of the inverse for different values of M.

Looking at our results and at the literature, we are able to fix the value of M to focus our study on the performance of the three algorithms analyzing the inversion of other two Laplace Transforms, this time holding to class $\mathbb{G}$, whose inverse is known.

THE STUDIED FUNCTIONS

Class	Laplace Transform	Inverse of the Laplace Transform
$\mathbb{F}$	$F_{01}(s) = s\log(s)$	$f_{01}(t) = \frac{1}{t^2}$
	$F_{02}(s) = \log\left(1 + \frac{1}{s}\right)$	$f_{02}(t) = \frac{1 - e^{-t}}{t}$
	$F_{03}(s) = \frac{e^{\left(-\frac{1}{4s}\right)}}{\sqrt{s^3}}$	$f_{03}(t) = \frac{2\sin(\sqrt{t})}{\sqrt{\pi}}$
$\mathbb{G}$	$F_{04} = \arctan\left(\frac{1}{s}\right)$	$f_{04}(t) = t^{-1}\sin(t)$
	$F_{05}(s) = -\log(s^2 + 1)$	$f_{05}(t) = \frac{2\cos(t)}{t}$

Table 4.1: The analyzed functions.[42]

[42] The function are taken from Abate and Valko (2004) [9]

[43] GS stays for Gaver-Stefhest.

4.1.1 Inversion of $F_{01}(s) = s\log(s)$

We start analyzing the problem of the inversion of a simple good Laplace Transform for different values of the coefficient M .

As already said, the relative error that comes from the inversion of good Laplace transforms using the inversion algorithms does not depend on t, so we have chosen an arbitrary value for t ($t = 10$).

To get the following results, we have used the *invLT_table* function in MATLAB.

$f_{01}(10) = 0.01$, (real value of the inverse)						
M	**GS[43] estimate**	**Relative[44] error**	**Euler estimate**	**Relative error**	**Talbot estimate**	**Relative error**
2	0,0123	2,33E-01	0,0094	6,49E-02	0,0153	5,34E-01
4	0,0269	1,69E+00	0,0099	6,34E-03	0,0110	1,03E-01
6	-0,0056	1,56E+00	**0,0100**	**5,13E-04**	0,0102	1,50E-02
8	0,0173	7,35E-01	**0,0100**	**3,75E-05**	0,01	1,54E-03
10	0,0077	2,34E-01	**0,0100**	**2,52E-06**	**0,0100**	**1,20E-04**
12	0,0106	5,63E-02	**0,0100**	**1,56E-07**	**0,0100**	**4,90E-06**
14	0,0099	1,09E-02	**0,0100**	**8,46E-09**	**0,0100**	**2,37E-07**
16	0,0100	1,78E-03	**0,0100**	**4,22E-09**	**0,0100**	**7,28E-08**
18	**0,0100**	**2,33E-04**	**0,0100**	**2,69E-09**	**0,0100**	**8,63E-09**
20	**0,0100**	**8,90E-05**	**0,0100**	**6,55E-09**	**0,0100**	**6,95E-10**
22	0,0100	2,50E-03	**0,0100**	**3,04E-07**	**0,0100**	**3,94E-11**
24	0,0103	2,59E-02	**0,0100**	**2,09E-06**	**0,0100**	**4,01E-12**
26	0,0106	5,60E-02	**0,0100**	**3,68E-06**	**0,0100**	**4,77E-11**
28	-2,25598	2,27E+02	**0,0100**	**1,25E-05**	**0,0100**	**-8,70E-12**
30	120,9022	1,21E+04	**0,0100**	**3,24E-04**	**0,0100**	**-2,91E-10**
32	-122,8257	1,23E+04	0,0100	1,82E-03	**0,0100**	**1,19E-09**
34	7,02E+04	7,02E+06	0,0100	1,23E-03	**0,0100**	**5,38E-09**
36	-6,32E+05	6,32E+07	0,0088	1,17E-01	**0,0100**	**-1,15E-08**
38	-2,50E+07	2,54E+09	0,0057	4,29E-01	**0,0100**	**2,53E-08**
40	-1,90E+08	1,93E+10	-0,0022	1,22E+00	**0,0100**	**-1,80E-08**
42	9,24E+09	9,24E+11	0,1077	9,77E+00	**0,0100**	**-7,21E-08**
44	-4,90E+11	4,88E+13	0,8613	8,51E+01	**0,0100**	**8,83E-07**
46	1,69E+13	1,69E+15	-2,272	2,28E+02	**0,0100**	**1,24E-06**

[43] GS stays for Gaver-Stefhest.

[44] Notice that the number of significant digits is given by $-\log_{10}\left(\frac{Relative\ error}{0.5}\right)$ which means that if the value of the relative error is lower than 5×10^{-m}, then the estimate has at least $(m - 1)$ significant digits of x

48	-5,30E+14	5,31E+16	1,044	1,03E+02	**0,0100**	**2,63E-07**
50	9,31E+15	9,31E+17	-40,9342	4,09E+03	**0,0100**	**-1,31E-05**
52	8,41E+16	8,41E+18	125,5925	1,26E+04	**0,0100**	**-1,60E-05**
54	-1,03E+19	1,03E+21	1,35E+03	1,35E+05	**0,0100**	**1,46E-06**
56	1,53E+20	1,53E+22	-1,24E+04	1,24E+06	**0,0100**	**1,10E-04**
58	-5,92E+20	5,92E+22	-2,78E+04	2,78E+06	**0,0100**	**-4,78E-04**
60	6,14E+22	6,14E+24	1,44E+05	1,44E+07	**0,0100**	**4,16E-04**
62	2,31E+24	2,31E+26	8,23E+05	8,23E+07	0,0100	1,31E-03
64	-7,77E+25	7,77E+27	4,16E+06	4,16E+08	**0,0100**	**2,55E-05**
66	3,67E+26	3,67E+28	-1,50E+07	1,50E+09	0,0100	4,72E-03
68	-5,62E+28	5,62E+30	1,31E+08	1,31E+10	0,0096	3,89E-02
70	1,01E+30	1,01E+32	7,89E+08	7,89E+10	0,0094	5,91E-02
72	-1,86E+31	1,86E+33	5,96E+08	5,96E+10	0,0094	6,14E-02
74	1,71E+32	1,71E+34	-1,20E+08	1,20E+10	0,0202	1,02E+00
76	-1,09E+31	1,09E+33	-2,63E+10	2,63E+12	0,0027	7,32E-01
78	-2,86E+35	2,86E+37	4,51E+11	4,51E+13	0,0348	2,48E+00
80	-2,33E+35	2,33E+37	1,06E+12	1,06E+14	-0,0053	1,53E+00
82	5,63E+37	5,63E+39	5,33E+12	5,33E+14	0,1892	1,79E+01
84	1,68E+39	1,68E+41	-1,83E+13	1,83E+15	0,1727	1,63E+01
86	7,79E+40	7,79E+42	1,44E+14	1,44E+16	-0,2651	2,75E+01
88	7,03E+41	7,03E+43	-2,57E+12	2,57E+14	2,9741	2,96E+02
90	9,77E+42	9,77E+44	-1,83E+15	1,83E+17	2,1201	2,11E+02
92	8,26E+44	8,26E+46	-3,90E+15	3,90E+17	11,0657	1,11E+03
94	-1,36E+45	1,36E+47	-7,92E+16	7,92E+18	14,1375	1,41E+03
96	-1,39E+47	1,39E+49	-4,04E+16	4,04E+18	-67,1322	6,71E+03
98	-6,82E+47	6,82E+49	8,74E+17	8,74E+19	442,0075	4,42E+04
100	3,51E+50	3,51E+52	1,68E+19	1,68E+21	622,1894	6,22E+04

Table 4.2: Estimations with the three algorithms of the inverse of the Laplace Transform F_{01} for different values of M

Analyzing Table 4.2, if we fix the threshold of at least four digits to define an approximation to be good, we can identify the following intervals of “goodness” for the three algorithms:

- $M = 18, 20$ for GS algorithm,
- $M = 6, \dots, 30$ for Euler algorithm,
- $M = 10, 12, \dots, 60, 64$ for Talbot algorithm.

It is immediate to see that Talbot and Euler provide better approximations than GS algorithm as they produce some estimations with more than 10 digits of accuracy.

	CPU execution time		
M	**Gaver-Stefhest**	**Euler**	**Talbot**
2	0,3350	0,6178	0,4150
4	0,4799	**0,6338**	0,3247
6	0,8251	**0,8785**	0,4204
8	1,3769	**0,9684**	**0,7155**
10	1,7652	**2,4007**	**0,5628**
12	2,3387	**1,2730**	**0,5456**
14	3,1269	**1,3617**	**0,6351**
16	**4,2337**	**1,9504**	**0,4532**
18	**2,8745**	**1,0793**	**0,4528**
20	**3,3351**	**1,0735**	**0,4729**
22	**3,9907**	**1,1737**	**0,5078**
24	4,7604	**1,2743**	**0,5345**
26	8,8410	**1,3543**	**0,6010**
28	6,1077	**1,4245**	**0,6068**
30	6,8122	**1,4397**	**0,6425**
32	9,5639	**1,3518**	**0,8087**
34	6,5154	**1,1478**	**0,4676**
36	6,2522	1,0842	**0,4664**
38	7,4620	1,2110	**0,5173**
40	6,3450	1,0009	**0,4159**
42	7,2182	1,1696	**0,4438**
44	7,2535	1,1683	**0,4475**
46	7,8685	1,1125	**0,4688**
48	8,5910	1,2053	**0,5788**
50	9,1977	1,2283	**0,5099**
52	9,9995	1,2246	**0,5136**
54	10,6116	1,2730	**0,5353**
56	11,3965	1,3412	**0,5550**
58	12,2130	1,3674	**0,5653**
60	13,2627	1,4319	**0,5883**
62	14,5846	1,4857	**0,6039**
64	15,4307	1,5353	**0,6228**
66	16,7883	1,5612	0,6494
68	17,0104	1,6445	0,6511
70	18,1590	1,6478	0,6946

Table 4.3: Execution time of the three inversion algorithms for different values of M, expressed in milliseconds.

It is now important to make a comparison among the CPU execution time of the three algorithms for different values of M.
Notice that the execution time has been expressed in millisecond and rounded at the fourth digit. In addition, the table has been cut for $M = 70$, as after that value all the three inversion algorithms seems to degenerate.
From table 4.3, it is immediate to see that, for the inversion of F_{01} , the Talbot algorithm is the fastest among the three inversion algorithms, taking more than three time less than the GS routine.
Another important consideration to be made is that in general the execution time increases as the value of M becomes higher. However, it is possible to see that the relationship within the CPU execution time and the value of M is not linear.
In the end, we can say that all the three algorithms provide a good estimation of the inverse of F_{01} and that the most efficient algorithm is the Talbot one.

4.1.2 Inversion of $F_{02}(s) = \log\left(1 + \frac{1}{s}\right)$

As we did for F_{02}, we first analyze the estimations generated by the three inversion algorithms for different values of M and then we are going to compare the execution time of the three inversion routines.
Remember that the Laplace transform F_{02} is a good Laplace transform, which means that the relative error is almost independent from t, so we can fix a value for t being sure that the analysis is accurate.

As noticed in the analysis of F_{01}, there exist an interval of M where the approximation is good[45]. For GS algorithm we can see, in Table 4.4, that the interval is between 14 and 18 , while for the Euler method it is between 8 and 32 and for Talbot algorithm it is between 8 and 66. As in paragraph 4.1.1, we made a comparison of the execution time implied by the three routines and we cut the table for values of M higher than 70.

[45] We consider an approximation to be good with at least four digits of accuracy.

$f_{02}(10) = 0,0999954600070238$ (real value of the inverse)						
M	**GS estimate**	**Relative error**	**Euler estimate**	**Relative error**	**Talbot estimate**	**Relative error**
2	0,087384	0,126124	0,106073	6,08E-02	0,100243	0,00248
4	0,103068	0,03073	0,100325	3,29E-03	0,100228	0,00233
6	0,101099	0,01104	0,100013	1,72E-04	0,099982	0,000136
8	0,09992	0,000751	**0,099996**	**8,15E-06**	**0,099993**	**2,03E-05**
10	0,099862	0,001334	**0,099995**	**3,12E-07**	**0,099995**	**1,57E-06**
12	0,099954	0,000411	**0,099995**	**5,26E-09**	**0,099995**	**9,05E-08**
14	**0,099992**	**3,96E-05**	**0,099995**	**6,33E-10**	**0,099995**	**3,39E-09**
16	**0,099997**	**2E-05**	**0,099995**	**9,25E-11**	**0,099995**	**1E-11**
18	**0,099995**	**3,16E-06**	**0,099995**	**2,58E-10**	**0,099995**	**1,4E-11**
20	0,099984	0,000114	**0,099995**	**2,77E-09**	**0,099995**	**4,1E-13**
22	0,100086	0,0009	**0,099995**	**7,44E-09**	**0,099995**	**1,7E-12**
24	0,095963	0,040322	**0,099995**	**1,99E-08**	**0,099995**	**3,09E-13**
26	0,039903	0,600948	**0,099995**	**8,71E-08**	**0,099995**	**2,76E-13**
28	-2,21347	23,13569	**0,099995**	**1,84E-07**	**0,099995**	**5,2E-12**
30	-58,1507	582,5336	**0,099995**	**6,26E-08**	**0,099995**	**2,3E-11**
32	780,8996	7808,35	**0,099996**	**5,54E-06**	**0,099995**	**6,8E-11**
34	5847,944	58481,1	0,099984	1,11E-04	**0,099995**	**1,9E-12**
36	266740,8	2667528	0,100026	3,03E-04	**0,099995**	**7,5E-10**
38	-2,6E+07	2,56E+08	0,099315	6,81E-03	**0,099995**	**6,26E-10**
40	4,15E+08	4,1E+09	0,096352	3,64E-02	**0,099995**	**2,68E-09**
42	-1,1E+10	1,12E+11	0,097945	2,05E-02	**0,099995**	**5,8E-09**
44	3,77E+11	3,8E+12	0,080538	1,95E-01	**0,099995**	**3,84E-08**
46	-7,2E+12	7,25E+13	0,119294	1,93E-01	**0,099995**	**3,78E-08**
48	7,16E+13	7,2E+14	-0,71719	8,17E+00	**0,099995**	**2,11E-08**
50	-1,5E+15	1,49E+16	2,373687	2,27E+01	**0,099996**	**5,2E-07**
52	-4,5E+16	4,46E+17	-13,287	1,34E+02	**0,099995**	**3E-07**
54	-1,9E+17	1,92E+18	-12,5359	1,26E+02	**0,099995**	**1,1E-07**
56	-8,7E+18	8,65E+19	729,5434	7,29E+03	**0,099995**	**4,14E-06**
58	9,82E+19	9,8E+20	-2,70E+03	2,70E+04	**0,099996**	**1E-05**
60	-8,9E+21	8,92E+22	-1,07E+04	1,07E+05	**0,099995**	**1,53E-06**
62	5,49E+23	5,5E+24	9,41E+03	9,41E+04	**0,099994**	**1,89E-05**
64	-1E+25	1,02E+26	-1,49E+05	1,49E+06	**0,099997**	**1,8E-05**
66	1,79E+26	1,8E+27	9,70E+05	9,70E+06	**0,099991**	**4,26E-05**
68	-2,9E+27	2,92E+28	9,75E+05	9,75E+06	0,099943	0,000528
70	2,19E+28	2,2E+29	2,50E+07	2,50E+08	0,099967	0,00028
72	-8,8E+29	8,8E+30	-3,93E+07	3,93E+08	0,099951	0,000444
74	7,19E+30	7,2E+31	-1,28E+08	1,28E+09	0,100862	0,00866

76	-8E+32	7,99E+33	-6,20E+08	6,20E+09	0,099596	0,003993
78	1,83E+32	1,8E+33	-1,48E+10	1,48E+11	0,101246	0,01251
80	-2,9E+35	2,85E+36	-7,53E+10	7,53E+11	0,097256	0,027394
82	-4,4E+35	4,38E+36	-4,32E+11	4,32E+12	0,106759	0,06764
84	3,37E+37	3,4E+38	-9,55E+11	9,55E+12	0,104282	0,04287
86	2,93E+39	2,9E+40	1,98E+12	1,98E+13	0,077172	0,228244
88	9,45E+40	9,5E+41	-1,01E+12	1,01E+13	0,253109	1,53121
90	-1,2E+41	1,2E+42	-8,83E+13	8,83E+14	0,306833	2,06847
92	-7,2E+42	7,19E+43	-4,54E+14	4,54E+15	0,782439	6,82475
94	3E+44	3E+45	-2,98E+14	2,98E+15	1,050411	9,50458
96	1,67E+45	1,7E+46	-1,04E+16	1,04E+17	-1,21698	13,17033
98	2,18E+47	2,2E+48	-8,52E+16	8,52E+17	14,52765	144,283
100	5,23E+47	5,2E+48	-8,02E+16	8,02E+17	24,96178	248,629

Table 4.4: Estimations with the three algorithms of the inverse of the Laplace Transform F_{02} for different values of M

	CPU execution time		
M	**Gaver-Stefhest**	**Euler**	**Talbot**
2	0,1934	0,3641	0,1921
4	0,2730	0,4869	0,2061
6	0,4746	0,5008	0,2233
8	0,7221	**0,5374**	**0,2537**
10	0,9709	**0,5903**	**0,2750**
12	1,3071	**0,7398**	**0,3231**
14	**1,7012**	**1,0879**	**0,3941**
16	**2,2862**	**0,9200**	**0,3822**
18	**2,6027**	**0,9216**	**0,4044**
20	3,0629	**0,9598**	**0,4257**
22	3,7452	**1,0620**	**0,4758**
24	3,8835	**0,7049**	**0,2886**
26	2,7374	**0,7065**	**0,3075**
28	3,0457	**0,7061**	**0,2931**
30	3,7345	**0,7730**	**0,3674**
32	3,9410	**0,8186**	**0,3338**
34	4,6820	0,9220	**0,4122**
36	4,8873	0,8650	**0,3600**
38	5,3343	0,9040	**0,3871**
40	6,0663	1,0862	**0,4368**

42	6,5248	1,0735	**0,4388**
44	7,0552	1,1371	**0,4532**
46	7,7350	1,1536	**0,4778**
48	8,4309	1,1359	**0,4996**
50	8,9268	1,1679	**0,4770**
52	9,8086	1,3014	**0,5333**
54	10,5442	1,3309	**0,5168**
56	11,2109	1,2981	**0,5255**
58	11,8518	1,4450	**0,5550**
60	13,2512	1,4311	**0,5604**
62	13,9446	1,4487	**0,5719**
64	14,9853	1,6289	**0,6412**
66	15,8572	1,5263	0,6408
68	16,4980	1,5797	0,6269
70	17,8778	1,6199	0,7406

Table 4.5: Execution time of the three inversion algorithms expressed in millisecond

The Talbot algorithm seems to be the best performer for the inversion of F_{02} as it has the lowest execution time. As in paragraph 4.1.1, we can notice that the execution time increases not linearly as M rises up.
In the end, we can say that the Talbot algorithm provides for the most accurate estimations for F_{02}, spending the lowest execution time.

4.1.3 Inversion of $F_{03}(s) = \frac{e^{\left(-\frac{1}{4s}\right)}}{\sqrt{s^3}}$

This is the last good Laplace transform whose inversion will be studied. The procedure used is the same applied to the two previous problem of inversion. We start setting the time equal to 10 and then we invert F_{03} using the three inversion algorithms and then we compare the obtained estimations with the real value of the inverse calculated using the known inversion function. In the end, we analyze the execution time implied by the three inversion routines.

$f_{03}(10) = -0,0233388660799441$ (real value of the inverse)						
M	**GS estimate**	**Relative error**	**Euler estimate**	**Relative error**	**Talbot estimate**	**Relative error**
2	-0,236293	9,12	1,570691	68,30	0,617106	27,4411
4	-0,373425	15,00	-1,37566	57,94	0,641287	28,4772
6	-0,169092	6,25	1,971875	85,49	0,658997	29,2360
8	-0,044801	0,9196	-3,802484	161,92	1,763462	76,5590
10	-0,023542	0,0087	8,019725	344,62	0,17847	8,6469
12	-0,023083	0,0110	-17,79613	761,51	-1,08052	45,2971
14	-0,023321	7,67E-04	-214,1625	9,18E+03	-3,53829	150,6049
16	**-0,023341**	**7,37E-05**	605,5823	2,59E+04	-2,45735	104,2902
18	**-0,023341**	**1,12E-04**	-1764,785	7,56E+04	1,170572	51,1555
20	-0,023318	9,08E-04	5213,609	2,23E+05	4,88199	210,1785
22	-0,023404	0,00281	-15360,16	6,58E+05	9,260362	397,7786
24	-0,020876	0,10552	44182,4	1,89E+06	2,055536	89,0735
26	0,032254	2,38200	-119764,6	5,13E+06	3,306703	142,6822
28	-6,780658	289,53074	-1654820	7,09E+07	-25,0474	1072,2052

Table 4.6: The three-inversion-algorithm estimations for different values of M.

This time we have stopped the table for $M \leq 28$ as the estimations of the three inversion routines begin to degenerate.

In this case, we can see that GS algorithm is the only one that is able to provide a good approximation of the good Laplace transform inverse.

From McClure (2012) [31], we know that the poor performance of the algorithms can be due to the use of a double precision system and not to a problem of the inversion methods themselves. However, we are not discussing high precision in this papers, for further details see Abate and Valko (2004) [9].

At this point, we think that an analysis of the execution time implied by the three algorithms is useless as only two approximations are good.

4.1.4 Inversion of $\boldsymbol{F_{04}(s) = \arctan\left(\frac{1}{s}\right)}$

This Laplace transform belongs to class $\mathbb{G}$, which means that its relative errors are not independent from t.

Knowing that if we fix M and we increase the value of t the accuracy of the inversion decreases we have to fix a value of M that in our opinion will provide for the best estimations. Looking at the result already obtained and at the literature, in particular see McClure (2012) [31], Josso and Larsen (2012) [29], we have decided to set the value of M equal to:

- $M_{GS} = 18$ for GS algorithm,
- $M_{Euler} = 32$ for Euler algorithm,
- $M_{Talbot} = 64$ for Talbot algorithm.

These values are nothing more than the highest values for which the three inversion routines are able to produce a good approximation.

Once we have fixed the value of M, we can see how the three algorithms behave for different values of t. Notice that the values of M chosen cannot be seen as definitive so if they do not allow the three inversion algorithms to provide any good estimates they have to be changed.

The first thing to do is the analysis of the behavior of the three inversion routines for $t = 0.5, 1, 1.5, \dots, 10$. If the approximations obtained are good, we do not change the values of M and we test the three inversion algorithms for higher values of t until the three routines are able to provide for a good estimation. In Table 4.7, we have summarized the obtained results.

As in the previous case it is immediate to see, in Table 4.7, that the Gaver-Stefhest algorithm performance is the worst as from $t \geq 2$ its estimations starts to go away from the real value, for example from $t = 9$ on the estimate is around half of the real value. This can be seen more clearly in the Figure 4.1.

Time (t)	$f_{04}(t)$ (real val.)	GS, $M_{GS} = 18$		Euler, $M_{Eul} = 32$		Talbot, $M_{Tal} = 64$	
		Est.	Rel. Err	Est.	Rel. Err	Est.	Rel. Err
0,05	0,95885	**0,95885**	**7,57E-07**	**0,95885**	**1,07E-07**	**0,95885**	**1,45E-06**
1,00	0,84147	**0,84147**	**2,15E-07**	**0,84147**	**3,76E-07**	**0,84147**	**1,20E-06**
1,50	0,66500	**0,66500**	**1,82E-06**	**0,66500**	**9,99E-07**	**0,66500**	**2,44E-06**
2,00	0,45465	**0,45456**	**2,02E-04**	**0,45465**	**4,39E-07**	**0,45465**	**9,69E-07**
2,50	0,23939	0,23961	9,26E-04	**0,23939**	**2,62E-06**	**0,23939**	**7,43E-06**
3,00	0,04704	0,04803	2,10E-02	**0,04704**	**2,18E-05**	**0,04704**	**2,32E-05**
3,50	-0,10022	-0,10066	4,33E-03	**-0,10022**	**3,68E-06**	**-0,10022**	**7,49E-06**
4,00	-0,18920	-0,19357	2,31E-02	**-0,18920**	**9,64E-07**	**-0,18920**	**4,73E-06**
4,50	-0,21723	-0,22224	2,31E-02	**-0,21723**	**4,20E-06**	**-0,21723**	**4,39E-06**
5,00	-0,19178	-0,18944	1,22E-02	**-0,19179**	**6,22E-06**	**-0,19178**	**8,80E-06**
5,50	-0,12828	-0,11464	1,06E-01	**-0,12828**	**7,22E-06**	**-0,12828**	**1,15E-05**
6,00	-0,04657	-0,02780	4,03E-01	**-0,04657**	**2,57E-06**	**-0,04657**	**2,84E-05**
6,50	0,03310	0,04388	3,26E-01	**0,03309**	**3,92E-05**	**0,03310**	**4,81E-05**
7,00	0,09386	0,08549	8,91E-02	**0,09386**	**1,81E-06**	**0,09386**	**1,02E-05**
7,50	0,12507	0,09581	2,34E-01	**0,12507**	**3,68E-06**	**0,12507**	**9,86E-06**
8,00	0,12367	0,08272	3,31E-01	**0,12367**	**2,15E-06**	**0,12367**	**7,07E-06**
8,50	0,09394	0,05726	3,91E-01	**0,09394**	**1,53E-06**	**0,09394**	**1,10E-05**
9,00	0,04579	0,02916	3,63E-01	**0,04579**	**1,21E-05**	**0,04579**	**2,46E-05**
9,50	-0,00791	0,00488	1,61670	**-0,00791**	**9,26E-05**	**-0,00791**	**1,47E-04**
10,00	-0,05440	-0,01254	0,76951	**-0,05440**	**1,27E-05**	**-0,05440**	**2,79E-05**

Table 4.7: Comparison between the estimations obtained with the three algorithms and the real value of the inverse at $t = 0.5, 1, 1.5, \dots, 10$, fixed M.

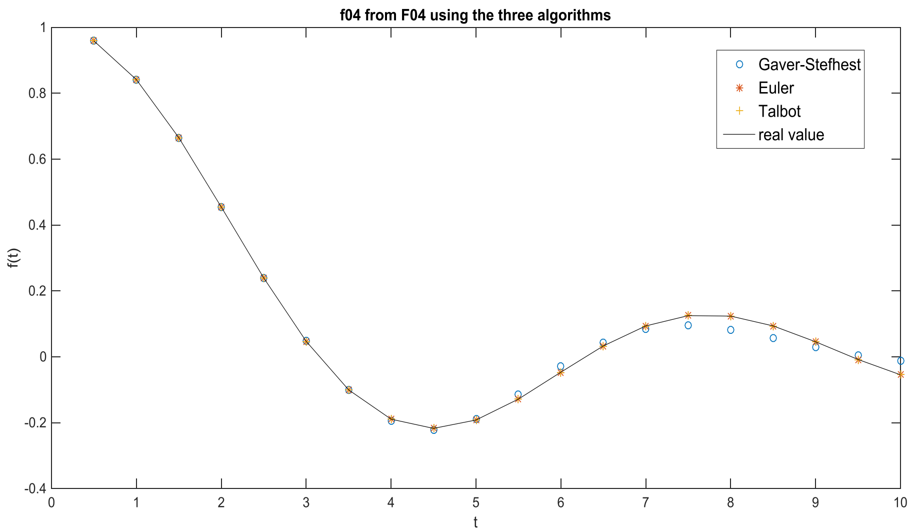

Figure 4.1: Comparison among the real value of the inversion and the approximations given by the three inversion algorithms of F_{04}, with $M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$ for $t = 1,2, \dots ,10$

As it is possible to notice from Figure 4.1 and Table 4.7, the Gaver-Stefhest method provides for the less accurate estimations while Euler and Talbot produce estimations with a mean of five digits of accuracy. Therefore, for higher values of t, we expect a good estimation from Euler and Talbot algorithm and an inaccurate approximation for Gaver-Stefhest.

Notice that we are going to present only a figure of all the estimations for $t = 1, 2, \dots, 250$ and only a synthetic table for the approximations and their relative errors and CPU execution time.

Time	$f_{04}(t)$ (real val.)	GS, $M_{GS} = 18$		Euler, $M_{Eul} = 32$		Talbot, $M_{Tal} = 64$	
		Est.	Rel. Err	Est.	Rel. Err	Est.	Rel. Err
50	-0,00525	0,00001	1,00	**-0,00525**	**3,21E-05**	-0,00135	0,74
100	-0,00506	0,00000	1,00	**-0,00506**	**1,27E-04**	-0,00068	0,87
150	-0,00477	0,00000	1,00	-0,06163	11,90	-0,00045	0,91
200	-0,00437	0,00000	1,00	0,00000	1,00	-0,00034	0,92
250	-0,00388	0,00000	1,00	0,00000	1,00	-0,00027	0,93

Table 4.8: Comparison among the estimations of the three inversion algorithms and the real value of the inverse for $t = 50{,}100{,}150{,}200{,}250$ of F_{05}.

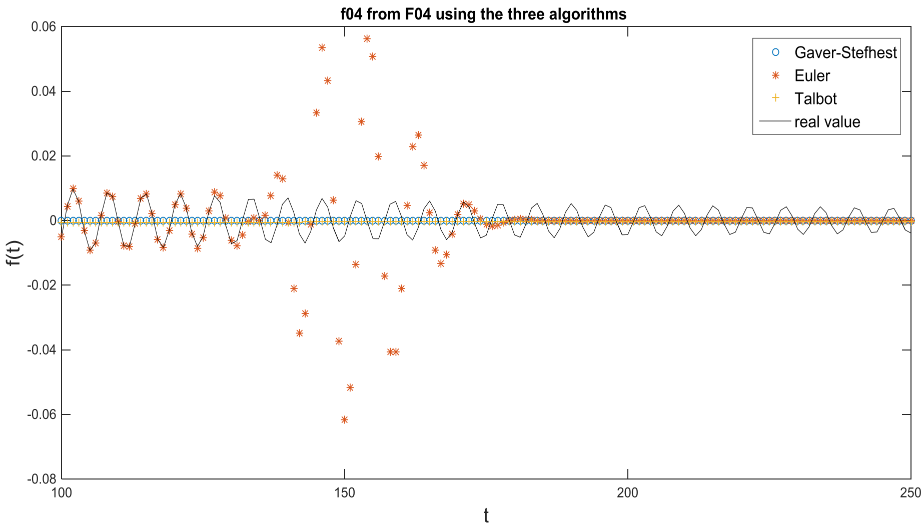

Figure 4.2: Comparison among the real value of the inversion and the approximations given by the three inversion algorithms of F_{04}, with $M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$ for $t = 100, 101, \dots, 250$.[46]

[46] In Figure 4.2, we have represented only the values of $t \geq 100$ to highlight the behavior of the Euler algorithm as from Table 4.8 we know that the Gaver-Stefhest and the Talbot algorithms cannot provide good estimations already from $t < 50$.

From Figure 4.2 it is possible to notice that the Euler algorithm provides for good estimations for high values of t, however around $t = 150$ it degenerates ending aligned with estimations of the other two algorithms. In addition, it possible to see in Table 4.8 that until $t = 50$ the Euler method still produces good approximations, it stops producing good estimates around $t = 100$
What it is not possible to see from that picture is when the Talbot algorithm stops to produce good estimations. However, analyzing the errors in Table 4.8, we can notice that the algorithm starts to do not provide for good approximation around $t = 50$.
As we expected, the Gaver-Stefhest routine is not able to provide good estimates for high values of t. We have tried to increase the value of the coefficient M but the accuracy of the results did not change and for values of $M > 22$ it starts to degenerates.
In the end, we can say that for the inversion of F_{04} Talbot and Euler algorithms produce good approximations for low values of t, however only Euler routine seems to provide good estimations for high values of t.
Lastly, we present a brief analysis of the CPU execution time.

	CPU execution time		
Time	**GS**	**Euler**	**Talbot**
50	2,28411	0,90314	0,62645
100	2,30710	1,04394	0,73647
150	2,31654	1,05954	0,76151
200	2,24101	1,05954	0,76602
250	3,25211	1,13097	0,73770

Table 4.9: Execution time of the three inversion algorithms for different values of t ($M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$) expressed in milliseconds.

From Table 4.9, we can see that for all the algorithms the time increases not linearly as t rises up (fixed M). In addition, we can notice that Talbot algorithm is always the fastest routine, followed by Euler and then by GS. Anyway, this does not change the fact that for F_{04} Euler is the only routine that is able to provide for better results for high values of t.

4.1.5 Inversion of $F_{05}(s) = -\log(s^2 + 1)$

The last transform we are going to analyze belongs to class $\mathbb{G}$, as F_{04}.

As we did for the previous Laplace transform, we first study the performance of the three algorithms for relatively low values of t, $t = 0.5, 1, \dots, 10$, and then for higher values. In the end, we briefly present the time implied by the three algorithms for different high values of t.

From Table 4.10 and Figure 4.3, it is possible to notice that Talbot algorithm is the only inversion routine that does not provide for a good approximation. We have tried to change the value of M but all the obtained estimations were worse than the ones reported in Table 4.5.

In addition, we can notice that Euler algorithm produces good approximations for all the considered values of t and GS method provides for good estimations until $t = 2$.

Therefore, we suppose that the only algorithm that will provide for good approximations for high values of t will be the Euler method.

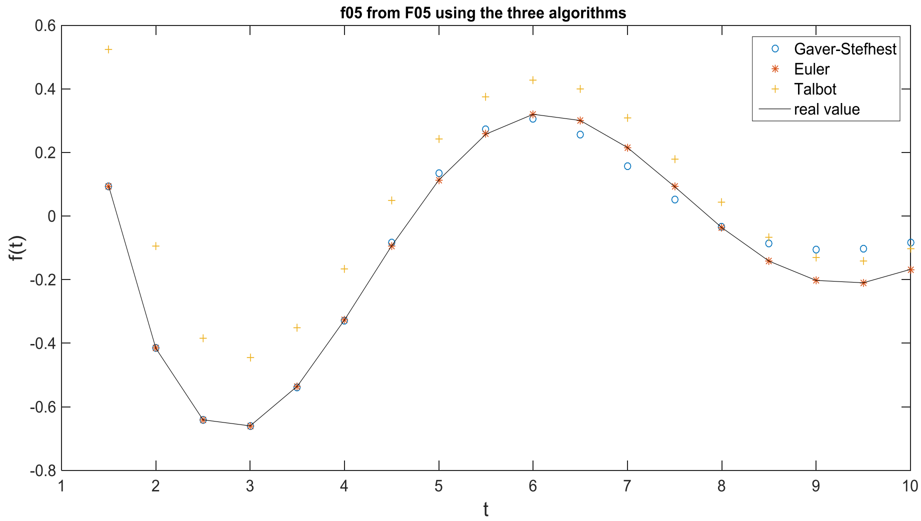

Figure 4.3: Comparison among the real value of the inversion and the approximations given by the three inversion algorithms of F_{05}, with $M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$ for $t = 1.5, 2, \dots, 10$[47]

[47] Notice that in Figure 4.3, we have not represented the estimations for $t = 0.05, 1.00$ because they would make the chart difficult to read as they have a different order of magnitude with respect to the other approximations

Time (t)	$f_{04}(t)$ (real val.)	GS, $M_{GS} = 18$		Euler, $M_{Eul} = 32$		Talbot, $M_{Tal} = 64$	
		Est.	Rel. Err	Est.	Rel. Err	Est.	Rel. Err
0,05	3,51033	**3,51040**	**1,92E-05**	3,50971	1,75E-04	4,79722	0,3666
1,00	1,08060	**1,08062**	**1,83E-05**	**1,08093**	**3,04E-04**	1,72399	0,5954
1,50	0,09432	**0,09433**	**1,30E-04**	0,09438	7,14E-04	0,52331	4,5484
2,00	-0,41615	**-0,41604**	**2,65E-04**	**-0,41613**	**4,33E-05**	-0,09442	0,7731
2,50	-0,64091	-0,64024	1,06E-03	**-0,64085**	**1,02E-04**	-0,38354	0,4016
3,00	-0,65999	-0,66097	1,48E-03	**-0,65996**	**4,67E-05**	-0,44549	0,3250
3,50	-0,53512	-0,53986	8,87E-03	**-0,53510**	**2,69E-05**	-0,35127	0,3436
4,00	-0,32682	-0,32888	6,29E-03	**-0,32684**	**6,22E-05**	-0,16595	0,4922
4,50	-0,09369	-0,08290	1,15E-01	**-0,09364**	**4,90E-04**	0,04933	1,5265
5,00	0,11346	0,13495	1,89E-01	**0,11346**	**3,21E-05**	0,24217	1,1343
5,50	0,25770	0,27144	5,33E-02	**0,25768**	**6,09E-05**	0,37470	0,4540
6,00	0,32006	0,30602	4,39E-02	**0,32004**	**4,69E-05**	0,42731	0,3351
6,50	0,30049	0,25507	1,51E-01	**0,30047**	**5,29E-05**	0,39950	0,3295
7,00	0,21540	0,15691	2,72E-01	**0,21541**	**3,54E-05**	0,30734	0,4268
7,50	0,09244	0,05142	4,44E-01	**0,09244**	**5,96E-06**	0,17824	0,9283
8,00	-0,03638	-0,03359	7,67E-02	**-0,03636**	**4,78E-04**	0,04407	2,2115
8,50	-0,14165	-0,08599	3,93E-01	**-0,14165**	**5,01E-06**	-0,06594	0,5345
9,00	-0,20247	-0,10632	4,75E-01	**-0,20249**	**6,34E-05**	-0,13097	0,3532
9,50	-0,20993	-0,10214	5,13E-01	**-0,20993**	**7,20E-06**	-0,14219	0,3227
10,00	-0,16781	-0,08314	5,05E-01	**-0,16782**	**2,25E-05**	-0,10346	0,3835

Table 4.10: Comparison between the estimations with the three algorithms and the real value of the inverse at $t = 0.5, 1, 1.5, \dots, 10$, fixed M.

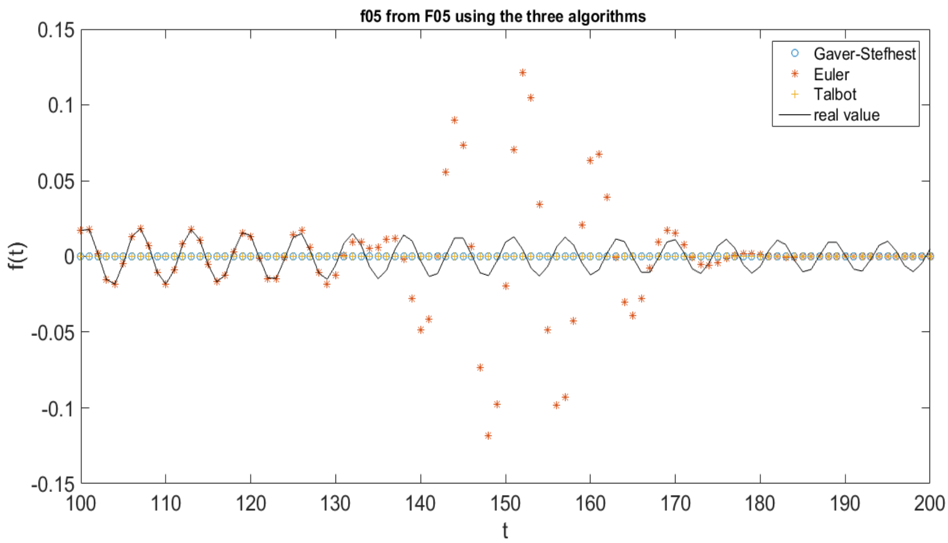

Figure 4.4: Comparison among the real value of the inversion and the approximations given by the three inversion algorithms of F_{05}, with $M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$ for $t = 100, 101 \dots, 200$[48]

[48] 48 In Figure 4.4, we have represented only the values of $t \geq 100$ to highlight the behavior of the Euler algorithm as from Table 4.11 we know that the Gaver-Stefhest and the Talbot algorithms cannot provide good estimations already from $t < 50$.

Time (t)	$f_{05}(t)$ (real val.)	GS		Euler		Talbot	
		Est.	Rel. Err	Est.	Rel. Err	Est.	Rel. Err
50	0,03860	0,00005	0,99872	**0,03860**	**5,50E-06**	0,00000	1,00001
100	0,01725	0,00000	1,00010	**0,01725**	**1,87E-05**	0,00000	1,00000
150	0,00932	0,00000	1,00004	-0,01967	3,10975	0,00000	0,99999
200	0,00487	0,00000	1,00001	0,00000	0,99995	0,00000	1,00001
250	0,00193	0,00000	1,00004	0,00000	0,99999	0,00000	1,00001

Table 4.11: Comparison among the estimations of the three inversion algorithms and the real value of the inverse for $t = 50{,}100{,}150{,}200{,}250$ of F_{05}.

From Table 4.11, it is possible to notice how the Euler methods can approximate the real value of the inverse with a high accuracy for at least $t \leq 100$. Then, in Figure 4.4, we can see that around $t = 150$ the algorithms stops to produce good estimations, as noticed in the inversion of F_{04}.
At this point, we can say that for the inversion of F_{05} the Euler algorithm provides for the best approximations with the variation of the time.

For completeness, we present the analysis of the CPU execution time even if only one of the three algorithms produces significant estimations.

	CPU execution time		
Time	**GS**	**Euler**	**Talbot**
50	2,08542	1,03040	0,82062
100	2,11334	1,11004	0,79928
150	2,13715	1,00946	0,89287
200	2,20365	1,06283	0,79681
250	2,30464	1,10963	0,79353

Table 4.12: Execution time of the three inversion algorithms for different values of t ($M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$) expressed in milliseconds.

As for the previous inversion, the fastest routine is the Talbot one followed by the Euler algorithm and the last is always the GS (whose execution time is two times the one of the other two inversion algorithms).

In the end, we can say that the Euler algorithm is the best inversion algorithm for the inversion of the Laplace transform F_{05}.

4.1.7 Considerations

In the analysis of the first three Laplace transforms (F_{01}, F_{02} and F_{03}) we have found a range of value for the number of coefficients to sum for each t and, exploiting the information found in the literature, we have fixed a value for it. The choice of a standard coefficient M worked for all the functions considered; however, we suggest testing more than one value when you apply one of the three algorithms to a new function and considering the fixed value of M as a starting point rather than a final destination.

The analysis highlights the accuracy and the efficiency of the three-inversion algorithm for several values of t. Furthermore, considering the results obtained and the literature[49], the following considerations can be made:

- Talbot and Euler algorithm converges to the real value of the function faster than Gaver-Stefhest. Therefore, we can say that the empirical results confirms what said in the theory in chapter 2.
- The CPU execution time needed by Talbot algorithm is the lowest, however also Euler algorithm has a good execution time.
- The performance of the three algorithms depends upon the studied function and this is the reason why when we have to invert a Laplace Transform we use more than one algortihm at the same time.

[49] In particular see Abate and Valko (2004) [9], Abate Choudhury and W. Whitt (1999) [14].

4.2 Lookback Options: results

In this section, we are going to compare the estimation of the price of standard floating lookback call options[50] using four different methods:

- Analytic approach (closed-form formula, see equation 3.11),
- Cox Rubistein binomial tree,
- Monte Carlo simulation,
- Numerical inversion of the Laplace Transform, exploiting the three-inversion algorithms studied. In other words, we want to invert the Laplace transform of the lookback option price found in Chapter 3, in particular see equation 3.11, using the three inversion algorithm studied in chapter 2 and analyzed in the first section of this chapter.

The analysis of the accuracy of the estimations provided by these three methods is assessed comparing them with the fair price calculated using the known closed-form formula for the price of floating lookback call options.

Firstly, we analyze the estimation of a lookback price looking at the accuracy of the approximations and the time implied by the computer for the three mentioned pricing methods. What we expect is a better approximation and a higher efficiency for Monte Carlo simulation and numerical inversion models.

Then, in the second part of the paragraph, we are going to study only the Laplace transform approach varying σ, seeing if it can still provide good estimations for extreme values of σ.

[50] For simplicity, from now on, we are going to indicate the floating lookback call as lookback option.

4.2.1 Lookback options: comparison among different pricing methods

We have chosen a standard lookback option at the money, on an underlying asset with spot price $S = m = 100,\ r = 0.05,\ \sigma = 0.35,\ q = 0$, that will expire in one year.
The analytical formula gives the following result:

$$LB_{price} = \mathbf{26.9234}$$

with an execution time equal to 0,53 milliseconds.

First, let us see how the Cox-Ross-Rubistein Binomial Tree and the Monte Carlo simulation can approximate the price of a lookback price with the specified characteristics.

$LB_{an\_price} = \mathbf{26.9234}$ **(fair price obtained using the closed-form formula 3.11)**							
Cox-Rubistein Binomial Tree				**Monte Carlo Simulation**			
No. Steps	**Est. Price**	**Relative Error**	**CPU Time**	**No. Steps/ Replication**	**Est. Price**	**Relative Error**	**CPU Time**
1k	26,5232	1,49E-02	2	5k/10k	26,6283	1,10E-02	4
2k	26,6396	1,05E-02	15	5k/50k	26,7209	7,52E-03	22
3k	26,6913	8,62E-03	58	5k/100k	26,6838	8,90E-03	44
4k	26,7223	7,47E-03	142	10k/10k	26,6735	9,28E-03	9
5k	26,7434	6,69E-03	264	10k/50k	26,5438	1,41E-02	44
6k	26,7590	6,11E-03	465	100k/100k	**26,9221**	**4,83E-05**	**95**

Table 4.13: Binomial Tree and Monte Carlo simulation estimation of the price of a call lookback, with $S = m = 100,\ r = 0.05,\ \sigma = 0.35,\ q = 0$ and relative execution time, expressed in seconds.

From Table 4.13, we can see that the binomial tree does not provide for an accurate estimation as all its approximations have only two digits of accuracy. In addition, we can notice that the binomial tree is a method that needs a high execution time to produce good estimations.
On the other hand, Monte Carlo simulation seems to produce better estimates as it can provide for an estimation with four digits of accuracy in a minute and a half. We can affirm that the Monte Carlo simulation method provides for a

more efficient estimation than CRR binomial tree for the pricing of a floating lookback option. Now, let us see how the inversion algorithms perform.

	$LB_{an\_price} = 26.9234$ (fair price obtained using the closed-form formula 3.11)								
	Gaver-Stefhest			**Euler**			**Talbot**		
M	**Est. Price**	**Relative Error**	**CPU Time**	**Est. Price**	**Relative Error**	**CPU Time**	**Est. Price**	**Relative Error**	**CPU Time**
4	27,5251	2,24E-02	0,6	27,0199	3,58E-03	1,2	**26,9267**	**1,22E-04**	**0,6**
6	26,9477	9,03E-04	0,7	**26,9280**	**1,70E-04**	1,5	**26,9237**	**9,32E-06**	**0,7**
8	**26,9236**	**4,79E-06**	**0,7**	**26,9236**	**8,21E-06**	1,6	**26,9234**	**8,21E-08**	**0,6**
10	**26,9234**	**1,78E-06**	**0,9**	**26,9234**	**3,97E-07**	1,3	**26,9234**	**1,37E-08**	**0,6**
12	**26,9234**	**1,49E-07**	**0,9**	**26,9234**	**1,96E-08**	1,3	**26,9234**	**1,53E-09**	**0,6**
14	**26,9234**	**1,25E-08**	**1,2**	**26,9234**	**9,73E-10**	1,6	**26,9234**	**1,02E-10**	**0,7**
16	**26,9234**	**9,43E-08**	**1,4**	**26,9234**	**4,75E-11**	1,8	**26,9234**	**4,61E-12**	**0,8**
18	**26,9235**	**1,64E-06**	**1,6**	**26,9234**	**2,62E-11**	1,9	**26,9234**	**8,91E-14**	**0,9**
20	**26,9238**	**1,23E-05**	**1,9**	**26,9234**	**5,24E-10**	2,1	**26,9234**	**8,17E-14**	**1,0**
22	26,9044	7,06E-04	2,3	**26,9234**	**7,13E-10**	2,2	**26,9234**	**3,70E-15**	**1,1**
24	27,4681	2,02E-02	2,6	**26,9234**	**3,71E-09**	2,4	**26,9234**	**6,02E-13**	**1,2**
26	20,787	2,28E-01	2,9	**26,9234**	**1,03E-08**	2,7	**26,9234**	**2,64E-13**	**1,4**
28	-1,26E+02	5,68E+00	3,5	**26,9234**	**6,65E-08**	2,8	**26,9234**	**4,27E-13**	**1,3**
30	6,08E+03	2,25E+02	3,8	**26,9234**	**4,64E-07**	3,0	**26,9234**	**5,76E-13**	**1,4**
32	-1,66E+05	6,17E+03	4,1	**26,9233**	**4,75E-06**	3,2	**26,9234**	**3,60E-12**	**1,5**
34	3,97E+06	1,47E+05	4,8	**26,9239**	**1,59E-05**	3,4	**26,9234**	**8,95E-12**	**1,6**
36	-7,30E+07	2,71E+06	5,2	**26,9214**	**7,43E-05**	3,6	**26,9234**	**7,99E-11**	**1,7**
38	1,17E+09	4,35E+07	5,7	**26,9230**	**1,73E-05**	3,8	**26,9234**	**5,13E-11**	**1,8**
40	-1,45E+10	5,39E+08	6,3	26,9528	1,09E-03	4,0	**26,9234**	**2,66E-10**	**1,8**
42	1,37E+09	5,09E+07	6,8	26,8426	3,00E-03	4,3	**26,9234**	**3,02E-10**	**1,9**
44	9,55E+12	3,55E+11	8,0	26,7200	7,54E-03	4,5	**26,9234**	**2,65E-10**	**2,1**
46	-3,92E+14	1,46E+13	8,1	25,8962	3,83E-02	4,5	**26,9234**	**2,47E-09**	**2,2**
48	1,15E+16	4,27E+14	8,6	30,7311	1,42E-01	4,6	**26,9234**	**7,73E-10**	**2,2**
50	-2,46E+17	9,14E+15	9,1	-9,22E+01	4,42E+00	4,9	**26,9234**	**1,53E-08**	**2,3**
52	3,73E+18	1,39E+17	10,1	2,97E+01	1,04E-01	5,1	**26,9234**	**2,86E-09**	**2,4**
54	-3,46E+19	1,29E+18	10,7	-2,29E+03	8,58E+01	5,3	**26,9234**	**9,06E-09**	**2,5**
56	-3,28E+20	1,22E+19	11,5	-1,82E+02	7,76E+00	5,4	**26,9234**	**1,78E-07**	**2,6**
58	1,90E+22	7,06E+20	12,6	-3,93E+04	1,46E+03	5,8	**26,9234**	**1,05E-07**	**2,7**
60	-4,87E+23	1,81E+22	13,1	1,79E+04	6,65E+02	5,9	**26,9234**	**1,15E-07**	**2,9**
62	8,95E+24	3,32E+23	13,9	2,17E+05	8,06E+03	6,1	**26,9234**	**1,20E-07**	**2,9**
64	-1,38E+26	5,13E+24	14,8	-1,93E+05	7,17E+03	6,2	**26,9235**	**1,42E-06**	**3,1**
66	4,74E+26	1,76E+25	15,5	-1,37E+06	5,09E+04	6,5	**26,9233**	**3,13E-06**	**3,1**
68	3,19E+28	1,18E+27	16,3	3,29E+07	1,22E+06	6,5	**26,9231**	**1,23E-05**	**3,1**
70	-1,92E+30	7,13E+28	17,8	-4,06E+06	1,51E+05	7,1	**26,9233**	**6,31E-06**	**3,3**

Table 4.14: Estimation of the price of a call lookback, with $S = m = 100,\ r = 0.05,$ $\sigma = 0.35,\ q = 0$ and relative execution time, expressed in milliseconds.

We can immediately notice, see Table 4.14, that the estimations given by the numerical inversion methods are the most accurate among the used models. All the three inversion algorithms provide for an estimation with at least four digits of accuracy. In particular, the Talbot algorithm seems to be the best performer as it gives a good approximation for all the tested values of M.
It is also possible to notice what we have already seen in the first paragraph: the Gaver-Stefhest and Euler algorithm degenerates for relatively low values of M. The former starts to degenerate around $M = 28$, however it begins losing digits of accuracy at $M = 20$, while the latter starts to get away from the fair price at $M = 50$ even if it begins losing precision at $M = 46$.
Furthermore, for value of $M \leq 6$ both the algorithms provide for an estimate with less than two digits of accuracy.
In the end, we look at execution time. The Talbot algorithm performs better than the other two inversion algorithms taking half the time of Euler algorithm and Gaver-Stefhest. It is also interesting to notice that for $M \in [8,12]$ Talbot is only few milliseconds faster than Gaver-Stefhest, however in this interval Talbot produces an approximation with seven digits of accuracy which is the maximum accuracy given by Gaver-Stefhest.

In conclusion, we can affirm that the Laplace Transform method provides the best approximations for the price of a floating lookback call options, if we consider the closed-form formula as our yardstick.
As we expected, the worst estimate come from the Cox-Rubistein binomial tree, which provide for an approximation with zero digits of accuracy for 6000 time steps.
Instead, Monte Carlo Simulation can provide for an estimation with four digits of accuracy (using $100{,}000$ time steps and $100{,}000$ replications), however it takes one minute and half to calculate it.

Now, that we have seen which one of the three pricing model provide for a more accurate estimation, let us consider only the three inversion algorithms and see how they behave when the volatility changes.

4.2.2 Numerical inversion for different values of σ

It is important now to study the behavior of the three inversion algorithms varying σ, as the price of an option strictly depends on the value of the underlying volatility.

First, we have chosen a fixed value of M for each inversion algorithm:

- Gaver-Stefhest: $M_{GS} = 18$;
- Euler: $M_E = 32$;
- Talbot: $M_T = 64$.

These values have been taken looking at the results obtained in Table 4.14, in the previous paragraph and in the literature. Notice that we have chosen high values of M since we do not know if for high/low value of the volatility the algorithms need more terms to converge to the fair price calculated by the closed-form formula 3.11.

Then using the same data of the previous paragraph ($S_0 = m = 100$, $T = 1$, $r = 0.05$, $q = 0$) and varying σ, we obtain Table 4.15.

σ	LB_{an_price}	GS $M_{GS} = 18$	Relative Error	Euler $M_E = 32$	Relative Error	Talbot $M_T = 64$	Relative Error
0,05	6,8878	6,8878	3,12E-06	6,8877	6,05E-06	6,8877	7,63E-06
0,10	10,3013	10,3013	2,37E-06	10,3011	1,42E-05	10,3013	8,56E-07
0,15	13,7904	13,7905	1,21E-05	13,7903	3,70E-06	13,7904	3,91E-07
0,20	17,2168	17,2169	3,32E-06	17,2168	2,93E-06	17,2168	8,02E-07
0,25	20,5522	20,5523	6,64E-06	20,5521	1,96E-06	20,5522	1,70E-07
0,30	23,7884	23,7886	8,80E-06	23,7884	2,80E-06	23,7884	3,02E-07
0,35	26,9234	26,9234	1,31E-06	26,9233	3,89E-06	26,9235	1,95E-06
0,40	29,9573	29,9574	4,82E-06	29,9572	1,87E-06	29,9573	2,21E-06
0,45	32,8911	32,8911	2,89E-07	32,8909	4,49E-06	32,8911	6,34E-07
0,50	35,7264	35,7265	2,85E-06	35,7263	3,33E-06	35,7264	6,23E-07
0,55	38,4652	38,4653	1,22E-06	38,4651	2,90E-06	38,4653	1,24E-06
0,60	41,1095	41,1096	7,87E-07	41,1094	3,81E-06	41,1095	6,72E-07
0,65	43,6614	43,6615	1,71E-06	43,6613	2,72E-06	43,6614	2,87E-07
0,70	46,1230	46,1231	2,66E-06	46,1229	2,25E-06	46,1230	3,55E-07
0,75	48,4965	48,4965	9,33E-08	48,4964	1,29E-06	48,4965	2,93E-07
0,80	50,7841	50,7842	1,23E-06	50,7840	2,36E-06	50,7841	7,75E-08
0,85	52,9880	52,9880	4,80E-08	52,9880	7,09E-07	52,9880	2,82E-07
0,90	55,1104	55,1104	5,07E-07	55,1104	2,72E-07	55,1103	3,35E-07
0,95	57,1534	57,1535	1,93E-06	57,1533	8,08E-07	57,1534	5,33E-07

Table 4.15: Estimation of the price of a floating lookback option varying σ using the three inversion algorithms.

From this analysis, we can say that the approximations produced by the three inversion algorithms are accurate as they provide an estimate with a mean of six/seven digits of accuracy, which is much more than the accuracy required (four digits). Therefore, we can affirm that the inversion algorithms works well also for extreme values of the volatility.

In the end, we want to see which one of the three algorithm is the fastest and if their execution time is lower or higher than the one of the closed-form formula.

	CPU execution time			
σ	**Closed-form formula**	**GS**	**Euler**	**Talbot**
0,05	0,23	2,19	3,34	4,08
0,10	0,16	3,04	5,26	4,29
0,15	0,15	2,87	5,83	4,00
0,20	0,15	2,51	3,96	3,19
0,25	0,15	2,07	3,26	2,89
0,30	0,15	1,85	3,18	2,92
0,35	0,15	2,01	3,34	3,03
0,40	0,15	2,65	3,44	3,10
0,45	0,15	2,15	3,35	4,56
0,50	0,14	2,05	3,30	2,97
0,55	0,14	2,23	3,33	3,16
0,60	0,17	1,82	3,37	2,87
0,65	0,17	1,77	3,08	3,02
0,70	0,16	1,78	3,14	3,25
0,75	0,22	1,86	3,14	4,85
0,80	0,41	1,88	3,63	3,47
0,85	0,28	1,83	3,14	2,92
0,90	0,36	1,87	3,15	2,90
0,95	0,49	1,83	3,21	3,34

Table 4.16: Execution time of the three inversion algorithms for different values of σ ($M_{GS} = 18$, $M_{Euler} = 32$ and $M_{Talbot} = 64$) expressed in milliseconds

From Table 4.16 we can see that Gaver-Stefhest takes generally less time than the other two algorithms. It is also possible to notice that all the three algorithms takes more time when they have to compute the price of a lookback call for $\sigma \leq 0.10$.

Nevertheless, all the three inversion routines have execution times higher than the analytic formula. This means that for the pricing of a floating lookback option, it is convenient to use the closed-form formula.
However, from that analysis we have showed that for lookback options the Laplace Transform approach can be a good alternative to the other two proposed financial models. Therefore, we can try to apply this method to options, whose closed-form formula is not known, for example arithmetic Asian options, and see if it can provide for better approximations with respect to the CRR-binomial tree and the Monte Carlo simulation model.

CHAPTER 5

ASIAN OPTIONS

We have concluded the previous chapter talking about Asian options and explaining why they need an entire section.

In this chapter, for pricing a continuous arithmetic Asian call option we are going to follow the procedure suggested by Fusai (2000)[24] and Dufresne (2004)[27], which is a review of the original method used by German and Yor (1993)[26].

The entire unit is divided into four parts. The first part wants to present what is the precise problem that we have to solve for pricing an Asian option[51]. Then, in the second part, we presents Bessel processes whose importance comes from the role played in Yor's theorem demonstration. This theorem is an essential element for pricing an arithmetic Asian option in the chosen method and it will be presented in the third part.

In the last part, all the obtained result are used together to obtain the price of an arithmetic Asian option.

Notice that it is not intention of this paper to provide a detailed explanation of theorems and definitions that together lead to the price of an arithmetic Asian option; rather the goal of the chapter is to give an idea of the mathematical reasoning steps that there are under that procedure.

[51] From now on, for the sake of simplicity, the continuous arithmetic Asian call option will be simply indicated as arithmetic Asian option.

5.1 The core of the problem

We have just said that our target is to price a continuous arithmetic Asian call option. The payoff for that option has been already defined in formula 3.12. From that equation is immediate to see that the distribution of the sum does not have a simple form.

As a first step, it can be useful to express the formula of an Asian payoff in a different way.

Kowing that the price of the underlying asset follows a Geometric Brownian motion and using the result found in chapter 1 section 1.3.2, we can affirm that:

$$S(t) = S(0) \cdot e^{\left(\alpha - \frac{1}{2}\sigma^2\right)t} \cdot e^{\sigma W(t)}$$

Substituting this result into the formula of an arithmetic Asian payoff, we get:

$$\left(\int_{O}^{T} \frac{1}{T} S(u) du - K\right)^{+} = \left(\frac{S(0)}{T} \int_{0}^{T} e^{\left(\alpha - \frac{1}{2}\sigma^2\right)u} \cdot e^{\sigma W(u)} du - K\right)^{+} \qquad (5.1)$$

In formula 5.1, we have three parameters but using the scaling property of the Brownian motion[52] and fixing one of the three it is possible to simplify the equation. As suggested by Yor (2001) [28], we apply the following definitions and substitutions:

- $\sigma = 2$ (the fixed parameter);
- $\int_0^T e^{\mu u + \sigma W(u)} \doteq \frac{4}{\sigma^2} A_t^{(v)}$
- $\mu = \alpha - \frac{1}{2}\sigma^2$
- $t = \frac{\sigma^2 T}{4}$
- $v = \frac{2\mu}{\sigma^2}$

So, we can define the following quantity:

$$A_t^{(v)} = \int_0^t e^{2(vu + \sigma W(u))} du \qquad (5.2)$$

[52] **Scaling property of a Brownian motion:**
For each $s > 0$ the quantity

$$s^{1/2} W_{st}, \quad t \geq 0$$

is a Brownian Motion starting at 0.

However, we can apply directly the transformation to formula 5.2 for computing the price of an Asian option rather than to the formula of the payoff. So, considering the martingale approach to BSM model, we get that:

$$CAA_{Call} = e^{-rT} E_0^Q \left[\left(\int_O^T \frac{1}{T} S(u) du - K \right)^+ \right]$$

$$= e^{-rT} E_0^Q \left[\left(\frac{S(0)}{T} \int_O^T e^{\left(\alpha - \frac{1}{2}\sigma^2\right)t} \cdot e^{\sigma W(t)} du - K \right)^+ \right]$$

To obtain the quantity expressed in equation (5.1), the following computations have to be made:

$$CAA_{Call} = e^{-rT} E_0^Q \left[\left(\frac{4}{\sigma^2} \frac{S(0)}{T} \int_O^T \frac{\sigma^2}{4} e^{\left(\alpha - \frac{1}{2}\sigma^2\right)t} \cdot e^{\sigma W(t)} du - K \right)^+ \right]$$

$$= e^{-rT} E_0^Q \left[\left(\frac{4}{\sigma^2} \frac{S(0)}{T} A_t^{(v)} - K \right)^+ \right]$$

We want to leave the term $A_t^{(v)}$ alone, so we take out from the brackets the quantity $\frac{\sigma^2}{4} \frac{T}{S(0)}$ and we obtain the following equation:

$$CAA_{Call} = e^{-rT} \frac{4}{\sigma^2} \frac{S(0)}{T} E_0^Q \left[\left(A_t^{(v)} - \frac{\sigma^2}{4} \frac{T}{S(0)} K \right)^+ \right]$$

We can define a new constant variable $q = \frac{\sigma^2}{4} \frac{T}{S(0)} K$. In the end, we obtain:

$$CAA_{Call} = e^{-rT} \frac{4}{\sigma^2} \frac{S(0)}{T} E_0^Q \left[\left(A_t^{(v)} - q \right)^+ \right] \tag{5.3}$$

In the next two paragraphs, we want to explain the mathematical steps that lead to the solution of this formula that means finding a solution for $E_0^Q \left[\left(A_t^{(v)} - q \right)^+ \right]$.

To do that we begin by making a digression on Bessel processes and then presenting the Yor's theorem, mentioned in Dufresne (2004)[26] in the chapter "the law of $A_t^{(v)}$ at an independent exponential time".

5.2 Bessel process

This section takes the cue and follows the line of reasoning suggested by Dufresne (2004)[26].

We start giving the following definition.

Definition 5.1: Bessel process

For $x \geq 0$, the square root of $(BESQ)^{\delta}(x^2)$ is called Bessel process of dimension δ started at x, denoted by $BES^{\delta}(x)$.

To understand the previous statement we need to give a definition of quadratic Bessel process, which in the statement is indicated as $(BESQ)^{\delta}(x^2)$.

First, we proceed giving a definition for a quadratic Bessel process for integer dimensions.

The following notation will be adopted:

- $W = (W^1, \dots, W^{\delta})$ a δ −dimensional Brownian Motion starting at $x \in \mathbb{R}^{\delta}$
- ρ is the norm of W.

Using Definition 1.12 (the Itô formula for a multidimensional Itô process) on the variable ρ^2, we get:

$$\rho_t^2 = \rho_0^{2}[53] + 2\int_0^t \rho_s d\beta_s + \delta t$$

where β_t is a one dimensional process:

$$\beta_t = \sum_{i=1}^{\delta} \int_0^t \left(\frac{B_s^i}{\rho_s}\right) dB_s^i.$$

[53] Notice that $|\rho_0^2| = |x|^2$

It can be demonstrated, using the definition of quadratic variation, that the defined process is a Brownian motion.

Finally, we can extend the definition to all possible positive dimensions, $\delta \geq 0$, and for $x \in \mathbb{R}^+$ obtaining the following equation

$$Z_t{}^{54} = x + 2\int_o^t \sqrt{|Z_s|}d\beta_s + \delta t \qquad (5.4)$$

Then a quadratic Bessel process is the unique strong solution to stochastic differential equation 5.4.

We now can give a formal definition.

Definition 5.2: Quadratic Bessel process

Let $x, \delta \geq 0$. The unique strong solution of

$$Z_t = x + 2\int_o^t \sqrt{|Z_s|}d\beta_s + \delta t$$

is called quadratic Bessel process of dimension δ starting at x, indicated with $Z \sim BESQ^{(\delta)}(x)$.

The two Bessel processes have different properties, such as additivity and scaling properties, which are a natural derivation of their definitions. The most important result for our studies regard the Bessel process density function, which is found to be equal to:

$$\rho_t^{(v)}(x,y) = \frac{y}{t}\left(\frac{y}{x}\right)^v e^{-\frac{(x^2+y^2)}{2t}} I_v\left(\frac{xy}{t}\right) 1_{\{y>0\}} \quad for\ x,t > 0; v \geq 0$$

where $v = \frac{\delta}{2} - 1$is the index of a squared Bessel process.

The importance of that formula will be explained later.

---

[54] The new variable Z_t indicates the norm of a multidimensional Brownian motion for which the dimension δ can be every positive number.

5.3 The law of $A_t^{(v)}$ at an independent exponential time[55]

We start giving the following definition, which is essential to demonstrate Yor's theorem.

Definition 5.3: Resolvent of a Markov process

Let us consider a time homogeneous Markov process X under the objective measure P, defined as:

$$dX(t) = \mu X(t)dt + \sigma X(t)dW(t).$$

and a real valued sufficiently integrable function g. Then $\forall\ \alpha \geq 0$ the **resolvent** is an operator defined by the expression

$$[R_\alpha g](x) = E_x^P\left[\int_0^\infty e^{-\alpha s} g(X_s)\, ds\right] \qquad (5.5a)$$

where the sub index x denotes the condition $X_0 = x$.

We can recognize that the expression inside the brackets is equal to the definition of Laplace Transform given in the second chapter (see definition 5.2).

It is also immediate to notice that in our case:

$$g(X_s) = A_t^{(v)}$$

In addition, it is also possible to rewrite the definition of the resolvent in the following way:

$$E_x^P\left[\int_0^\infty e^{-\alpha s} g(X_s)\, ds\right] = \frac{1}{\lambda} E_x^P\left[\int_0^\infty \lambda e^{-\alpha s} g(X_s)\, ds\right]$$

$$= \frac{1}{\lambda} E_x^P\left[X_{S_\lambda}\right] \qquad (5.5b)$$

where S_λ and the process X_t are independent.

[55] The title is taken from Dufresne (2004)[26]

The goal is to find the density of the resolvent, which at that point it is equivalent to finding out the distribution of X_{S_λ}, and in our case of $A^{(v)}_{S_\lambda}$.

In Yor(2001)[28][56] the author proved that $A^{(v)}_{S_\lambda}$ is distributed as the combination of two well-known functions: gamma and beta. His result is reported more formally in the next theorem.

Theorem 5.1

The distribution of $A^{(v)}_t$ is characterized by:

$$2A^{(v)}_t \sim \frac{B_{(1,\alpha)}}{G_{(\beta,1)}}$$

or

$$2A^{(v)}_t \overset{\mathcal{L}}{=} \frac{B_{(1,\alpha)}}{G_{(\beta,1)}}$$

where $B \sim Beta(1, \alpha)$ and $G \sim Gamma(\beta, 1)$ are independent with:

- $\gamma = \sqrt{2\lambda + v^2}$
- $\alpha = \frac{v+\gamma}{2}$
- $\beta = \frac{\gamma-v}{2}$

A complete demonstration of the theorem can be found in Dufresne. That proof is an easier version of the one made by Yor(2001)[28] thanks to the probability density function and the properties of a Bessel process.

[56] The book published by Mar Yor in 2001 is a collection of his past papers.

5.4 Price of an arithmetic Asian option

Now we are able to solve equation (5.3) first applying the Laplace transform to

$$E^{57}\left(A_t^{(v)} - q\right)^+$$

which becomes equal to

$$\int_0^{+\infty} e^{-\lambda s} E\left(A_t^{(v)} - q\right)^+ ds\,.$$

Then using Fubini's theorem we can demonstrate that:

$$\begin{aligned}
\int_0^{+\infty} e^{-\lambda s} E\left(A_t^{(v)} - q\right)^+ ds &= \int_0^{+\infty} e^{-\lambda s} \int_{-\infty}^{+\infty} \left(A_t^{(v)} - q\right)^+ da\, ds \\
&= \int_0^{+\infty} e^{-\lambda s} \int_q^{+\infty} \left(A_t^{(v)} - q\right) da\ \ ds \\
&= \int_0^{+\infty} e^{-\lambda s} \int_q^{+\infty} A_t^{(v)} da\ \ ds - \int_0^{+\infty} e^{-\lambda s} \int_q^{+\infty} q\ da\ \ ds \\
&= \frac{1}{\lambda} \int_q^{+\infty} \int_0^{+\infty} \lambda e^{-\lambda s} A_t^{(v)} ds\ \ da + \frac{1}{\lambda} \int_q^{+\infty} q \int_0^{+\infty} \lambda e^{-\lambda s} ds^{\,58}\ da
\end{aligned}$$

In the first part of the equation, it is possible to recognize the definition of the resolvent given above, see formula (5.5b). Furthermore, we are able to solve the inner integral on the right side of the equation.

Then the result is:

$$\begin{aligned}
\int_0^{+\infty} e^{-\lambda s} E\left(A_t^{(v)} - q\right)^+ ds &= \frac{1}{\lambda} \int_q^{+\infty} A_{S_Y}^{(v)}\ da - \frac{1}{\lambda} \int_q^{+\infty} q\ da \\
&= \frac{1}{\lambda} \int_{-\infty}^{+\infty} \left(A_{S_Y}^{(v)} - q\right)^+ da \\
&= \frac{1}{2\lambda} E\left(2A_{S_Y}^{(v)} - 2q\right)^{+\,59}
\end{aligned}$$

---

[57] The index and the subscript have been deleted to simplify the notation.

[58] $\int_0^{+\infty} -\lambda e^{-\lambda s} ds = e^{-\lambda s}\Big|_0^{+\infty}$
$= -1$

[59] It has been divided and multiplied for two to connect that formula with Yor's theorem.

This new equation can be solved using Yor's theorem and initially not considering the constant term.

First, we see how we can compute the expected value for $\left(2A_{S_Y}^{(v)} - 2q\right)^+$:

$$E\left(2A_{S_Y}^{(v)} - 2q\right)^+ = E\left(\frac{B_{(1,\alpha)}}{G_{(\beta,1)}}{}^{60} - 2q\right)^+$$

$$= \int_{-\infty}^{+\infty}\int_0^1 \left(\frac{u}{x} - 2q\right)\frac{x^{\beta-1}}{\Gamma(\beta)}e^{-x}\alpha(1-u)^{\alpha-1}du 1_{\left\{\frac{u}{x}>2q\right\}}\,dx$$

$$= \int_0^1\int_0^{u/2q} \left(\frac{u}{x} - 2q\right)\frac{x^{\beta-1}}{\Gamma(\beta)}e^{-x}\alpha(1-u)^{\alpha-1}dx\ du$$

$$= \int_0^1\int_0^{u/2q} \left(\frac{u}{x} - 2q\right)\frac{x^{\beta-1}}{\Gamma(\beta)}e^{-x}\alpha(1-u)^{\alpha-1}du\ dx$$

Then, applying again Fubini's theorem:

$$E\left(2A_{S_Y}^{(v)} - 2q\right)^+ = \int_0^{1/2q} \frac{x^{\beta-1}}{\Gamma(\beta)}e^{-x}\,dx \int_{2qx}^1 \left(\frac{u}{x} - 2q\right)\alpha(1-u)^{\alpha-1}du.$$

In addition, it is provided a finite solution for the second integral in Fusai (2000) [24]. So, exploiting that result we get:

$$E\left(2A_{S_Y}^{(v)} - 2q\right)^+ = \int_0^{1/2q} \frac{x^{\beta-1}}{\Gamma(\beta)}e^{-x}\,dx\left[\frac{(1-2qx)^{\alpha+1}}{(1+\alpha)x}\right]$$

$$= \frac{1}{\left(1+\frac{v+\gamma}{2}\right)}\int_0^{1/2q} \frac{x^{\frac{\gamma-v}{2}-2}(1-2qx)^{\frac{v+\gamma}{2}+1}}{\Gamma\left(\frac{\gamma-v}{2}\right)}e^{-x}\,dx$$

Now, inverting the substitution made in theorem 5.1 and exploiting the property of the Gamma distribution for which:

$$\Gamma(\beta) = (\beta-1)\Gamma(\beta-1),$$

[60] The demonstration for the probability density function of B/G can be found in Dufresne (2004)[26]

we obtain:

$$E\left(2A_{S_\lambda}^{(v)} - 2q\right)^{+} = \frac{1}{\frac{\lambda - 2v - 2}{2}} \int_0^{1/2q} \frac{x^{\frac{\gamma-v}{2}-2}(1-2qx)^{\frac{v+\gamma}{2}+1}}{\Gamma\left(\frac{\gamma - v}{2} - 1\right)} e^{-x}\, dx$$

So now we can write the equation for the Laplace transform of $E\left(2A_{S_\lambda}^{(v)} - 2q\right)^{+}$

$$\frac{1}{2\lambda} E\left(2A_{S_Y}^{(v)} - 2q\ \right)^{+} = \frac{1}{\lambda(\lambda - 2v - 2)} \int_0^{1/2q} \frac{x^{\frac{\gamma-v}{2}-2}(1-2qx)^{\frac{v+\gamma}{2}+1}}{\Gamma\left(\frac{\gamma - v}{2} - 1\right)} e^{-x}\, dx$$

In the end, we can write the price of an arithmetic Asian option as follows:

$$CAA_{Call} = e^{-rT} \frac{4}{\sigma^2} \frac{S(0)}{T}\ \mathcal{L}^{-1}\left(\frac{1}{\lambda(\lambda - 2v - 2)} \int_0^{1/2q} \frac{x^{\frac{\gamma-v}{2}-2}(1-2qx)^{\frac{v+\gamma}{2}+1}}{\Gamma\left(\frac{\gamma - v}{2} - 1\right)} e^{-x}\, dx\right) \quad (5.6)$$

To find the value of the option we need to invert the integral using a numerical inversion, as the inverse of the transform is difficult to compute because of the Gamma argument form.

5.5 Numerical inversion of the Laplace transform of the arithmetic Asian option price

In this paragraph, our main target is to invert the Laplace transform of the Asian option price found at the end of the previous section, see formula 5.6, using the Euler algorithm. We have implemented only the Euler algorithm as in the literature it is the most widespread. It is also important to say that we have used the Euler routine exploiting a different representation of formula 2.6, see below. Furthermore, we have applied the Gaussian quadrature[61] to solve the definite integral in formula 5.6 as suggested in the literature, in particular see Fusai (2000) [24] and Fu, Madan and Wang (1999) [33].

Then, we want to compare the estimations obtained applying the Euler algorithm to formula 5.6 with other two pricing methods: the Cox-Ross-Rubistein binomial tree and the Monte Carlo simulation.

Unfortunately, for arithmetic Asian options we do not have a closed-form formula that can be used to make an analytical analysis of our estimations. Therefore, we have used Rogers and Shi (1992) [34]'s upper and lower bounds to determine if the estimations provided by three pricing methods are good.[62]

The entire section is based mainly on two papers: Fusai (2000) [24] and Fu, Madan and Wang (1999) [33].

[61] It is a high precision numerical integration method used to solve definite integrals of a function whose $n+1$ elements are known in an interval $[a, b]$. It is not aim of this dissertation to discuss the theory of numerical integration, for more details see Kythe and Schäferkotter (2004) [37].

[62] Remember that we have defined an approximation to be good when it has at least four digits of accuracy. However, in this chapter, a good approximation is also the estimate that lies within the given bounds.

5.5.1 The Euler algorithm proposed by Abate and Whitt (1995)

This version of the Euler algorithm was presented for the first time in Abate and Whitt (1995) [36] and the new parameters are (l, m, n, A) where:

- l, whose default value is one, is a parameter used to control the roundoff error,
- $m + n$ is the number of terms to sum in the algorithm,
- $A = \frac{2\ln(10)M}{3}$.

The approximant is given by the "new" formula:

$$f_e(m, n, t) = \sum_{k=0}^{m} \binom{m}{k} 2^{-n} s_{n+k}$$

where

- $s_n = \sum_{k=0}^{n} (-1)^k a_k$
- $a_k(t) = \frac{e^{\frac{A}{2l}}}{2lt} b_k(t)$
- $b_0(t) = F\left(\frac{A}{2lt}\right) + 2\sum_{j=1}^{l} Re\left\{F\left(\frac{A}{2lt} + \frac{ij\pi}{lt}\right) e^{\frac{ij\pi}{l}}\right\}$
- $b_k(t) = 2\sum_{j=1}^{l} Re\left\{F\left(\frac{A}{2lt} + \frac{ij\pi}{lt} + \frac{ik\pi}{t}\right) e^{\frac{ij\pi}{l}}\right\} \quad k \geq 1$

Setting $m = 11, n = 15$ and $A = 18.4$ the algorithm provides for at least three significant digits.

For further details see Abate, Choudhury and Whitt (1999) [14].

The found formulae will be applied to formula 5.6 to find the price of an Asian option. However, before applying the Euler inversion algorithm, we have to solve the integral in formula 5.6. In order to do that, we have implemented the Gaussian quadrature to compute the definite integral in formula 5.6. (In MATLAB, we have calculated the value of the definite integral using the function ltprice).

5.5.2 Asian option: numerical analysis

We begin by defining the characteristics of the Asian option we want to study. The option is at the money and it will expire in one year. It has an underlying with a spot price equal to 100, an interest rate of 0.05 and the variance equal to 0.2.

Before presenting the estimation obtained by the three pricing methods, we have to calculate the bounds as described by Rogers and Shi (1992) [34] and Thompson (1999) [35]. Notice that we have calculated the bounds for different values of σ as they will be used for further analiysis.

σ	Upper Bound	Lower Bound
0,01	2,41821	2,41821
0,02	2,42423	2,42422
0,03	2,47393	2,47390
0,04	2,57668	2,57664
0,05	2,71622	2,71617
0,06	2,87917	2,87910
0,07	3,05728	3,05718
0,08	3,24558	3,24545
0,09	3,44100	3,44083
0,10	3,64157	3,64134
0,15	4,68682	4,68611
0,20	**5,76443**	**5,76271**
0,25	6,85462	6,85118
0,30	7,95053	7,94436
0,35	9,04922	9,03892
0,40	10,14937	10,13297
0,45	11,25050	11,22527
0,50	12,35269	12,31491
0,55	13,45642	13,40119
0,60	14,56247	14,48348
0,65	15,67187	15,56126
0,70	16,78581	16,63403
0,75	17,90561	17,70133
0,80	19,03261	18,76274
0,85	20,16819	19,81784
0,90	21,31357	20,86624
0,95	22,46981	21,90757

Table 5.1: Lower and upper bound for arithmetic Asian options computed using the VBA codes provided by Fusai (2000) [24]

Next, we summarize the results obtained by Monte Carlo simulation and binomial tree for $\sigma = 0.20$.

Monte Carlo		CRR Binomial Tree	
No. Steps	**Est. Price**	**No. Steps**	**Est. Price**
250	5,7709	10	5,7312
500	5,7848	**25**	**5,7670**
1000	5,7591	50	5,7865
5000	**5,7642**	75	5,7935
10000	5,7421	100	5,7970

Table 5.2: Numerical results from Monte Carlo simulation, using control variates, with $R1 = 5000$ and $R2 = 50000$ and CRR binomial[63] tree that evaluates the path of the function 10 times in a step.

The estimates from the binomial tree lie all outside the bounds; the nearest approximation is the one obtained using $N = 25$ that differs from the upper bound of a relative error of $4{,}51E - 04$.

On the other hand, the Monte Carlo simulation method provides for an estimation inside the bounds when the number of steps are 5000. However, the time implied to obtain that result is around 28 seconds. In order to know if the execution time is "too much or too little" we have to wait and see what is the execution time of the Euler inversion algorithm.

Using the Euler algorithm with the suggested parameters, $m = 11, n = 15$ and $A = 18.4$, we get the following result.

σ	Est. Price	CPU time
0,2	5,7631	0,18

Table 5.3: Estimated price of the studied Asian option obtained using the Euler inversion algorithm. The time is expressed in seconds.

The estimation lies between the two extremes, so we can say that it is a good approximation. Furthermore, we can see that its computational time is 150 time less than the one of Monte Carlo simulation.

[63] This estimation are computed using the VBA code found at http://www.quantcode.com/modules/mydownloads/singlefile.php?cid=9&lid=503

Finally, we can say that the inversion algorithm works better than Monte Carlo Simulation and the binomial tree model as it provides for a more accurate estimate with a lower computational time

Let us now see if the Euler algorithm can provide for good approximations of the price also for other values of σ, especially the extreme ones (all the other parameters stay unchanged)

σ	**Upper bound**	**Estimated Price**	**Lower bound**	**CPU time**
0,05	2,7162	**-1,75E+11**	2,7162	0,21
0,10	3,64157	3,64155	3,6413	0,17
0,15	4,68682	4,68625	4,6861	0,19
0,20	5,76443	5,76309	5,7627	0,18
0,25	6,85462	6,85191	6,8512	0,18
0,30	7,95053	7,94563	7,9444	0,19
0,35	9,04922	9,04093	9,0389	0,20
0,40	10,14937	10,13597	10,1330	0,19
0,45	11,25050	11,22954	11,2253	0,17
0,50	12,35269	12,32079	12,3149	0,18
0,55	13,45642	13,40902	13,4012	0,18
0,60	14,56247	14,49366	14,4835	0,18
0,65	15,67187	15,57421	15,5613	0,18
0,70	16,78581	16,65022	16,6340	0,18
0,75	17,90561	17,72126	17,7013	0,16
0,80	19,03261	18,78695	18,7627	0,18
0,85	20,16819	19,84691	19,8178	0,19
0,90	21,31357	20,90078	20,8662	0,17
0,95	22,46981	21,94823	21,9076	0,17

Table 5.4:Estimation of a call Asian option using Euler inversion algorithm for different values of σ. The CPU time is expressed in seconds.

From table 5.4 it is possible to notice that for $\sigma < 0.1$ the algorithm cannot provide for a good estimation while for all the other values of the volatility the Euler method provide for good approximations, as they lie between the two bounds. Notice also that the estimation is always near to the lower bound and so we can suppose that if we do not get the fair price at least we obtain an underestimate of the fair price.

The next important step is to analyze the behavior of the Euler algorithm for $\sigma \in (0, 0.1]$. The results found are summarized in Table 5.5 and they highlight that the inversion algorithms stops to provide a good estimation when $\sigma < 0.09$.

We thought to change the value of the parameters involved in the Euler algorithm, but we did not as in Fusai (2000) [24] and Fu, Madan and Wang (1999) [33], it is demonstrated that it is not possible to improve the estimation setting different values of m.

At this point, we tried to compute the price for $\sigma \leq 0.1$ using Monte Carlo simulation.

In Table 5.5, it is possible to see that Monte Carlo can provide for better estimations. In fact, it has provided an approximation inside the bounds for $\sigma = 0.05$ while all the others have at least three digits of accuracy if compared with the lowest bound.[64]

σ	Upper bound	Euler	Monte Carlo	Lower bound	CPU	
					Euler	Monte Carlo
0,01	2,41821	**-1,20E+184**	2,41870	2,41821	0,17	25,60
0,02	2,42423	**7,68E+68**	2,42465	2,42422	0,15	26,20
0,03	2,47393	**1,17E+37**	2,47452	2,47390	0,15	25,80
0,04	2,57668	**5,27E+19**	2,57713	2,57664	0,15	25,30
0,05	2,71622	**-1,75012E+11**	**2,71267**	2,71617	0,15	26,50
0,06	2,87917	**78067666,626**	2,88512	2,87910	0,15	26,80
0,07	3,05728	**-1503,701**	3,06535	3,05718	0,15	25,40
0,08	3,24558	**7,175**	3,24578	3,24545	0,16	24,90
0,09	3,44100	3,440	3,43722	3,44083	0,16	26,50
0,10	3,64157	3,642	3,64644	3,64134	0,16	33,50

Table 5.5: Comparison between Euler inversion and Monte Carlo simulation for $\sigma \in (0, 0.1]$

To conclude our analysis, we have seen what happens to the Euler estimation when the studied option is in the money and out of the money, see Table 5.6 and Table 5.7 and for the following values of the volatility:

- a “standard case”, $\sigma = 0.2$;

[64] We have calculated the relative error taking the lower bound as “real” value and we have seen that all the relative errors are in magnitude of $E-04$.

- the limit case $\sigma = 0.09$,

The results are summarized in Table 5.6 and 5.7.

K	Upper bound	Est. Price	Lower bound	CPU time
90	12,6007	12,59599	12,5956	0,32
95	8,82073	8,81876	8,81839	0,32
105	3,50890	3,507332	3,506838	0,35
110	1,99272	1,98990	1,98924	0,33

Table 5.6: Estimation of an Asian option price with $K = 90{,}95{,}105{,}110, S_0 = 100$, $r = 0.05,\ \sigma = 0.2$, obtained using Euler inversion algorithm

K	Upper bound	Est. Price	Lower bound	CPU time
90	11,93984	**12,05069**	11,93914	0,39
95	7,327005	**7,329785**	7,326264	0,36
105	1,10190	1,101654	1,101562	0,38
110	0,03025	0,225258	0,029103	0,39

Table 5.7: Estimation of an Asian option price with $K = 90{,}95{,}105{,}110, S_0 = 100$, $r = 0.05,\ \sigma = 0.2$, obtained using Euler inversion algorithm

For the first case, see Table 5.6, we can see that the estimates of in the money and out of the money options are all between the two extreme values while, for the limit case, see Table 5.17, we can notice that the prices of in the money options are overestimated. However, they differ from the nearest bound for a maximum of $0{,}11085$ so we have to understand if Monte Carlo simulation provides for approximations that are more accurate.

The prices provided by Monte Carlo simulation are:

- $11{,}9381$ for $K = 90$
- $7{,}3241$ for $K = 95$.

It is immediate to see that the best approximation for $K = 90$ is the one calculated by Monte Carlo Simulation while for $K = 95$ both methods produce an estimation with a similar accuracy so we will choose the inversion algorithm as it is the fastest ($\cong 0.40$ seconds for Euler and 28 seconds for Monte Carlo).

Considerations

Our results confirm the results obtained by Fusai (2004) [24]. In particular, that the inversion of the Asian option Laplace transform provides for good approximations of the Asian option price.

However, for $\sigma < 0.1$ the model stops to produce any valid estimates. Therefore, we have to use Monte Carlo methods as it provides for good approximations also for low values of the volatility.

A similar consideration can be made also for in the money Asian options whose price cannot always be well approximated by the Laplace transform approach. For this reason, for in the money arithmetic Asian options we have always to compare the approximations obtained by Monte Carlo and the Laplace Transform approach in order to choose the best estimation.

CONCLUSIONS

Our work highlights the efficiency of the Laplace Transform approach in option pricing.
We have based our studies on the Black-Scholes-Merton financial model, as it still represents a valid instrument for option pricing, although its known lacks, and it can be seen as a starting point for the study of more complex financial models, such as the exponential Levy Models, which generalizes the BSM model assuming that the asset prices do “jumps”.
The other basic aspect of this thesis is represented by the Laplace Transform theory and its inversion. We have presented the three main inversion algorithms that have been used for inverting the Laplace Transform when the analytical method, based on the Residue Theorem, fails or is hard to be applied.
Then, we focused on the pricing of four different options: Europeans, Barriers, Lookback and Asian. For each of these options we have studied, looking at the literature, a procedure that allows us to find their price using the Laplace Transform and its inversion.
For the first two types of option, the Laplace Transform could have been inverted using the analytical method and so it was possible to reach a closed-formula for their price. The same could not have been done for the other two options. Therefore, once found the theoretical formulas for the Laplace Transform of Lookback and Asian price, we have implemented MATLAB codes that are able to invert the transform using the Gaver-Stefhest, the Euler and the Talbot algorithm.
By the way, before analyzing the inversion problem in option pricing we thought that it could have been useful to study the performance of the three algorithms in the inversion of Laplace Transforms with known inverse.

Relying on other papers and the results we have achieved implementing the three inversion algorithms in MATLAB, we are able to say that the Talbot and Euler inversion routines are in general the most efficient. However, for some of the studied Laplace Transform the Gaver-Stefhest algorithm was the only one that was able to provide for an approximation of the inverse.

This is the reason why we suggest testing all the three algorithms together, when it is possible, as the performance of the algorithms strongly depends on the considered Laplace Transform.

Then, we have applied the three inversion algorithms, mentioned above, to the Laplace Transform of the price of a floating standard lookback option and we have compared the results found with the ones provided by other two financial models: the Cox-Ross-Rubistein binomial tree and the Monte Carlo simulation model. Afterwards, we have determined which of the models provided the best estimations comparing the approximations found with the fair price calculated using the closed-form formula. Finally, we have showed that the inversion algorithms, with a lower CPU execution time, can provide for more accurate approximations than the CRR-binomial model and the Monte Carlo simulation. In the end, we have seen if the Laplace Transform approach was efficient to price arithmetic Asian options.

For this type of options we do not have any closed-formula, so the goodness of the estimations was determined looking at a confidence interval that includes the fair price of the option.

Notice that given the difficulty of the argument, we have decided to invert the Laplace Transform of the arithmetic Asian option exploiting only the Euler algorithm, as suggested by the literature.

The results, obtained through the comparison of the Monte Carlo model, CRR binomial model and the Euler inversion algorithm, demonstrated that the inversion routine was able to provide for the best estimates for almost all the studied cases, with the exception of the one where the value of the stock volatility was very low and for in-the-money options.

For the former case, we have seen that Monte Carlo simulation is able to produce good approximations while for the latter we have seen that Monte

Carlo cannot always provide for better estimations, so we suggested testing both models and then choose the one with the highest accuracy.
In the end, we can say that the Laplace Transform approach can be seen as an efficient pricing instrument for the studied options.

APPENDIX A

A.1 Probability theory: functions and distributions

In chapter 2, we found different types of function used to define the basic properties of the Laplace transform and the tables obtained applying them. Here, there is a brief explanation of these function properties.

i. Heaviside function

It is identified with $H(x)$ and it is also known as “unit step function”.

It is defined as

$$H(x) = \begin{cases} 1 & for\ x \geq 0 \\ 0 & otherwise \end{cases}$$

ii. Dirac Delta function

The delta function can be seen as the derivative of the Heaviside function so it can be defined as

$$\delta(x) = dH(x)$$

Alternatively, it can be considered as a function on the real line that is zero everywhere except at the origin

$$\delta(x) = \begin{cases} +\infty & x = 0 \\ 0 & x \neq 0 \end{cases}$$

and which is also constrained to satisfy the identity

$$\int_{-\infty}^{+\infty} \delta(x)dx = 1$$

iii. Floor function

The function floor identifies the largest integer less than or equal to x:

$$floor(x) = \lfloor x \rfloor$$

iv. Square wave

It is a periodic waveform composed by instantaneous transition between two levels.

An analytic formula for the square wave, $S(x)$, with an half amplitude A, period T, and offset x_0 is reported below

$$S(x) = A(-1)^{\left\lfloor \frac{2(x-x_0)}{T} \right\rfloor}$$

v. Triangular wave

As the square wave, it is a periodic waveform. Its analytic representation is the following

$$f(x) = \frac{2}{\pi}\sin^{-1}[\sin(\pi x)]$$

The last two functions can be represented using the Fourier series.

A.2 Proof of $\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right) = \frac{e^{-a^2/4t}}{\sqrt{\pi t}}$

The function $erfc\left(\frac{a}{2\sqrt{t}}\right) = 1 - erf\left(\frac{a}{2\sqrt{t}}\right)$ [65] is called complementary error function and its Laplace Transform can be found making use of tables like the one shown in Spiegel (1965)[7].

$$\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right) = erfc\left(\frac{a}{2\sqrt{t}}\right)$$

Applying Theorem 6 (Derivative Theorem), we have:

$$\mathcal{L}\left(\frac{d}{dt}\, erfc\left(\frac{a}{2\sqrt{t}}\right)\right) = s\,\mathcal{L}\left(erfc\left(\frac{a}{2\sqrt{t}}\right)\right) - erfc\,(0^+)$$

Noting that $erf\left(\frac{a}{2\sqrt{t}}\right) \to 0$ as $t \to 0^+$, we get

$$\mathcal{L}\left(\frac{d}{dt}\, erfc\left(\frac{a}{2\sqrt{t}}\right)\right) = e^{-\sqrt{s}a}$$

[65] Erf is the function error.

Also for this function the Laplace Transform can be found making use of the table in Spiegel (1965)[7]:

$$\mathcal{L}\left(\frac{a}{2\sqrt{\pi t^3}}\ e^{-a^2/4t}\right) = e^{-\sqrt{s}a}$$

Now we differentiate with respect to s obtaining

$$\frac{d}{ds}\mathcal{L}\left(\frac{a}{2\sqrt{\pi t^3}}\ e^{-a^2/4t}\right) = -\frac{ae^{-\sqrt{s}a}}{2\sqrt{s}}$$

and by Theorem 8 (Multiplication by t^n) by $t\ = 1$,

$$\mathcal{L}\left(-\frac{at}{2\sqrt{\pi t^3}}\ e^{-a^2/4t}\right) = -\frac{ae^{-\sqrt{s}a}}{2\sqrt{s}}$$

Simplifying, we obtain:

$$\boldsymbol{\mathcal{L}^{-1}\left(\frac{e^{-\sqrt{s}a}}{2\sqrt{s}}\right) = \frac{e^{-a^2/4t}}{\sqrt{\pi t}}}\ ■$$

A.3 Cauchy's Theorem

Let C be a simple closed curve. If $f(z)$ is analytic within and on a simple closed curve C, then:

$$\oint_C f(z)dz = 0$$

Expressed in another way, it is equivalent to the statement that

$$\int_{z_1}^{z_2} f(z)dz$$

has value independent of the path joining z_1 and z_2. Such integral can be evaluated as:

$$F(z_2) - F(z_1)\ where\ F'(z) = f\ (z).$$

A.4 Cauchy's Integral Formula

If $f(z)$ is analytic within and on a simple closed curve C and a is any point interior to C, then

$$f(a) = \frac{1}{2\pi i} \oint_C \frac{f(z)}{z-a} dz$$

and

$$f^{(n)}(a) = \frac{n!}{2\pi i} \oint_C \frac{f(z)}{(z-a)^{n+1}} dz$$

The Cauchy's integral formulas are quite remarkable because they show that if the function $f(z)$ is known on the closed curve C then it is also known within C, and the various derivatives at points within C can be calculated. Thus if a function of a complex variable has a first derivative, it has all higher derivatives as well. This of course is not necessarily true for functions of real variables.

A.5 Residues Theorem

Definition A.1: Singular points

A singular point of a function $f(z)$ is a value of z at which $f(z)$ fails to be analytic.

If $f(z)$ is analytic everywhere in some region except at an interior point $z = a$, we call $z = a$ an isolated singularity of $f(z)$.

Example.

If $f(z) = 1\,/(z-3)^2$ then $z = 3$ is an isolated singularity of $f(z)$.

Definition A.2: Poles

If $f(z) = \phi(z)/(z-a)^n$, $\phi(z) \neq 0$ where $\phi(z)$ is analytic everywhere in a region including $z = a$, and if n is a positive integer, then $f(z)$ has an isolated singularity at $z = a$ which is called a pole of order n. If $n = 1$, the pole is often called a simple pole; if $n = 2$ it is called a double pole, etc.

Example.

$$f(z) = z\,/(z-3)^2(z+1)$$

It has two singularities: a pole of order 2 at $z = 3$, and a pole of order 1 at $z = -1$.

Definition A.3: Residues

If $f(z)$ has a pole of order n at $z = a$ and

$$\sum_{n=-\infty}^{+\infty} a_n\,(z-a)^n$$

it is a Laurent series[66] for $f(z)$. Then the coefficient a_{-1} is called the residue of $f(z)$ at the pole $z = a$. It can be found from the formula:

$$Res(f,a) = \lim_{z\to a} \frac{1}{(k-1)!}\frac{d^{k-1}}{dz^{k-1}}\,[(z-a)^k f(z)]$$

where k is the order of the pole. In particular if a is a simple pole then

$$Res(f,a) = \lim_{z\to a}(z-a)f(z)$$

Theorem A.1: Residues theorem

If $f(z)$ is analytic within and on the boundary C except at a finite number of poles a_i having residues Res_i, then

$$\oint_C f(z)dz = 2\pi i \sum Res_i$$

Cauchy's theorem and the integral formulas are special cases of this result that we call the Residues theorem.

Now we use the residues theorem to show (see cap 2.5) that

$$f(t) = \frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} e^{ts}F(s)ds = \sum residues\ of\ \ e^{st}\ F(s)\ at\ poles\ of\ F(s)$$

We have

$$\frac{1}{2\pi i}\oint_C e^{st}F(s)ds = \frac{1}{2\pi i}\int_{\gamma-iT}^{\gamma+iT} e^{st}\,F(s)ds + \frac{1}{2\pi i}\int_\Gamma e^{st}F(s)ds$$

[66] The Laurent series of a complex function is a representation of that function as a power series. For more details see Krantz (1999) [38].

By the residue theorem

$$\frac{1}{2\pi i}\oint_C e^{st}F(s)ds = sum\ of\ residues\ of\ e^{st}F(s)\ at\ all\ poles\ of\ F(s) inside\ C$$

$$= \sum residues\ inside\ C$$

Thus

$$\frac{1}{2\pi i}\int_{\gamma-iT}^{\gamma+iT} e^{st}\,F(s)ds = \sum residues\ inside\ C - \frac{1}{2\pi i}\int_{\Gamma} e^{st}F(s)ds$$

Taking the limit as $R \to \infty$ the integral on Γgoes to zero (note n 7 in cap 2.5) and we find:

$$f(s) = sum\ of\ residues\ of\ e^{st}F(s)\ at\ all\ poles\ of\ F(s)$$

Example.

We evaluate $\mathcal{L}^{-1}\left(\frac{1}{(s+1)(s-2)^2}\right)$ by using the methods of residues.

We start:

$$\mathcal{L}^{-1}\left(\frac{1}{(s+1)(s-2)^2}\right) = \frac{1}{2\pi i}\int_{\alpha-i\infty}^{\alpha+i\infty} \frac{e^{ts}}{(s+1)(s-2)^2}ds$$

$$= \frac{1}{2\pi i}\oint_C \frac{e^{ts}}{(s+1)(s-2)^2}ds$$

$$= \sum residues\ of\ \frac{e^{ts}}{(s+1)(s-2)^2}\ at\ poles\ s=-1, s=2$$

Now, residues at simple pole $s=-1$ is

$$Res(f,a) = \lim_{s\to -1}(s+1)\left\{\frac{e^{ts}}{(s+1)(s-2)^2}\right\} = \frac{1}{9}e^{-t}$$

In addition, residue at double pole $s=2$ is

$$Res(f,a) = \lim_{s\to 2}\frac{1}{1!}\frac{d}{ds}\left\{(s-2)^2\,\frac{e^{ts}}{(s+1)(s-2)^2}\right\} = \lim_{s\to 2}\frac{d}{ds}\left\{\frac{e^{ts}}{(s+1)}\right\}$$

$$= \lim_{s\to 2}\left\{\frac{(s+1)te^{ts}-e^{ts}}{(s+1)^2}\right\} = \frac{1}{3}te^{2t} - \frac{1}{9}e^{2t}$$

Then

$$\mathcal{L}^{-1}\left(\frac{1}{(s+1)(s-2)^2}\right) = \sum residues\ = \frac{1}{9}e^{-t} + \frac{1}{3}te^{2t} - \frac{1}{9}e^{2t}$$

APPENDIX B

B.1 Second Linear Differential Equation

Definition B.1: Homogeneous and non-homogeneous equation

A homogeneous second order linear differential equation is written as:

$$\boldsymbol{y'' + p(t)y' + q(t)y = 0} \qquad (1.1a)$$

While a non-homogeneous second linear differential equation is defined as:

$$y'' + p(t)y' + q(t)y = g(t), \;\; g(t) \neq 0 \qquad (1.1b)$$

The easiest case considers $p(t)$, $q(t)$ and $g(t)$ as constants, and the previous equation can be reformulated as follows:

- $ay'' + by' + cy = 0$ $\qquad (2.1a)$
- $ay'' + by' + cy = f$ $\qquad (2.1b)$

Where a must be different from zero.

Theorem B.1: "General solution of a homogeneous second ODE"

If a, b, c are constants functions and y_1 and y_2 are both solutions of the linear homogeneous equation $ay'' + by' + cy = 0$. Then the function

$$y = c_1 y_1 + c_2 y_2 \qquad (3)$$

Where c_1 and c_2 are constants, is also a solution of the equation.

The characteristic equation

Let us consider the formula (1b). In all cases the solutions consists of exponential functions, or at least terms that can be reworked into exponential functions.

We start hypothesizing that there are some exponential functions of unknown exponents that would satisfy any equation of the form (1b). After that, it will be devised a way to find the specific exponents that would give the solution.

We begin considering $y = e^{rt}$ be a solution of (1b) for some unknown constant r, so that $y' = re^{rt}$ and $y'' = r^2 e^{rt}$. Then we can substitute in (1b) the values and we get

$$ar^2e^{rt} + bre^{rt} + ce^{rt} = 0$$

$$e^{rt}\,(ar^2 + br + c) = 0$$

Since it is known that e^{rt} is always different from zero and it is a solution of (1b), then the above equation is satisfied if and only if

$$(ar^2 + br + c) = 0$$

This equation is also called the characteristic equation. It is immediate to notice that it is a quadratic equation whose resolution method is really well known.

The general solution changes according to the sign of the discriminant:

1. $b^2 - 4ac > 0$

   There exists two distinct real roots. For Theorem 1, a solution of equation (1) is

   $$y = c_1e^{r_1x} + c_2e^{r_2x}$$

2. $b^2 - 4ac = 0$

   There exists two equal real roots. For Theorem 1, a solution of equation (1b) is

   $$y = c_1e^{rx} + c_2e^{rx}$$

3. $b^2 - 4ac < 0$

   There exists two complex roots[67]. For Theorem 1, a solution is

   $$y = e^{\alpha x}(c_1\cos(\beta x) + c_2\sin(\beta x))$$[68]

Now, we have to determine the general solution. To do that it is necessary to define what an initial-value and boundary-value problem are so that we are able to determine a unique solution. Then it will be possible to define the general or fundamental solution.

---

[67] Remember that the complex roots are defined as $r_1 = \alpha + i\beta$ and $r_2 = \alpha - i\beta$.

[68] Notice that sine and cosine can be written into exponential functions:

$$sin\theta = \frac{e^{i\theta} - e^{-i\theta}}{2} \qquad cos\theta = \frac{e^{i\theta} + e^{-i\theta}}{2}$$

B.2 Initial-value and Boundary-value problem

An initial value problem consists on finding a solution to equation (1b) so that the following initial conditions are satisfied:

$$y(x_0) = y_0 \qquad y'(x_0) = y_1$$

where y_0 and y_1 are given constants.

The following theorem guarantees the existence and the uniqueness of a solution of this initial value problem.

Theorem B.2: The existence and uniqueness of the solution of a second ODE initial value problem[69]

Considering the initial value problem:

$$y'' + p(t)y' + q(t)y = 0 \qquad y(t_0) = y_0, \;\; y'(t_0) = y'_0$$

If p, q and g are continuous on the interval $I: \alpha < t < \beta$ containing the point $t = t_0$. Then there exists a unique solution $\varphi(t)$ of the problem and this solution exists throughout the interval I.

For more details see Strauss(2008) [38]

A boundary-value problem for (1b) consists of finding a solution y of the differential equation that satisfies boundary conditions of the form $y(x_0) = y_0$, $y(x_1) = y_1$.

In contrast with the initial value problems, a boundary-value problem does not always have a solution.

Fundamental or general solutions

We have seen that the general solution of a second order homogenous linear equation is in the form of (3), where y_1 and y_2 are two distinct functions both satisfying the given equation.

[69] The theorem is expressed considering the general case.

According to the Theorem B.2, for any pair of initial condition $y(t_0) = y_0$ and $y'(t_0) = y'_0$ there must exist uniquely a corresponding pair of coefficients c_1, c_2 that satisfies the system equations:

$$y_0 = c_1 y_1(t_0) + c_2 y_2(t_0)$$
$$y'_0 = c_1 y'_1(t_0) + c_2 y'_2(t_0)$$

From linear algebra, it is known that the above system has always a unique solution $\forall\, y_0 \wedge y'_0$ if the coefficient matrix of the system is invertible or equivalently if the determinant of the coefficient matrix is different from zero.
The previous statement can be written as follow:

$$det\begin{pmatrix} y_1(t_0) & y_2(t_0) \\ y'_1(t_0) & y'_2(t_0) \end{pmatrix} = y_1(t_0)y'_2(t_0) - y'_1(t_0)y_2(t_0) \neq 0$$

This determinant is also called the Wronskian, $W(y_1, y_2)$.
If the previous conditions are satisfied, the solution of the form (3) is a general solution.

Theorem B.3: Solution of the non-homogenous linear equation[70]

The general solution of a second order non-homogenous linear equation (2a) can be expressed in the form

$$y = y_c + Y$$

Where:

- $\boldsymbol{y_c}$ is a general solution of the homogeneous equation called **complementary solution**.
- $\boldsymbol{Y}$ is any specific function that satisfies the non-homogenous equation called **particular solution**.

We have already discussed the method to find the former. To calculate the latter, two approaches can be used:

- The methods of Undetermined Coefficients
- The variation of Parameters Formula

[70] In addition, in this case the theorem is given for the general formula.

What it has been used in the dissertation is the second of the two methods and this is the reason why we report the theorem right after.

By the way, more details can be found at:

http://tutorial.math.lamar.edu/Classes/DE/NonhomogeneousDE.aspx

Theorem B.4: Variation of parameters formula

Let us suppose that:

- $a, b, c, f: (t_1, t_2) \to \mathbb{R}$ are continuous functions
- $y_1, y_2: (t_1, t_2) \to \mathbb{R}$ are two independent linear solution of the homogeneous equation

$$a(x)y'' + b(x)y' + c(x)y = 0 \quad a(x) \neq 0$$

- $W_{y_1 y_2}$ is the Wronskian of y_1, y_2 defined as

$$W_{y_1 y_2} = y_1(x)y_2'(x) - y_2(x){y_1}'(x)$$

- two functions $u_1$and u_2 are described by

$$u_1(t) = \int -\frac{y_2(x)f(x)}{a(x)W_{y_1 y_2}(x)}dx$$

$$u_2(t) = \int \frac{y_1(x)f(x)}{a(x)W_{y_1 y_2}(x)}dx$$

Then the function

$$y_p = y_1(x)u_1(t) + y_2(x)u_2(t)$$

is a particular solution to the non-homogeneous equation:

$$a(x)y'' + b(x)y' + c(x)y = f(x)$$

The proof can be found at:

http://tutorial.math.lamar.edu/Classes/DE/VariationofParameters.aspx

APPENDIX C

In this appendix it will be presented a different way to present the arbitrage pricing theory and so the BSM model. This approach has been presented as it has been used in chapters 3 and 4 to price Asian and Lookback options and because it is the most complete representation for the arbitrage pricing theory. All the definitions, propositions and theorems in this chapter has been taken from Björk (2009)[1].

C.1 The Arbitrage Pricing Theory under the Martingale Approach

The basic assumptions made by arbitrage pricing theory are the completeness of the market and the absence of arbitrage opportunities. The latter has been mentioned in chapter 1, section 1.4, talking about the assumptions at the basis of BSM model.

The most general hypothesis is that we are in a market composed by $N+1$ asset price processes $S_0, \dots, S_N$ given a priori under the objective probability measure P. Under this assumption, the previous two conditions can be expressed exploiting the concept of martingale. Before doing that, it is important to give some fundamental definitions and the notation we are going to use from now on.

Notation

- $h(t) = [h_0(t), \dots, h_N(t)]$, it indicates a generic portfolio strategy.
- $S_i(t)$ is the i-th element of the vector $S(t) = [S_1(t), \dots, S_N(t)]$.
- $S_0(t)$ is the numeraire and it is nonnegative P-a.s[71] $\forall t \geq 0$.

[71] P-a.s stands for with probability equal to 1 almost surely.

Definition C.1:Portofolio Strategy

A portfolio strategy is any adapted process with dimension $N+1$.

Definition C.2: Value process

A value process is defined as

$$V(t;h)=\sum_{i=0}^{N}h_i(t)S_i(t).$$

Definition C.3: Admissible portfolio strategy

A portfolio strategy $h(t)$ is admissible if and only if $\exists\alpha\in\mathbb{R}^+$ so that

$$\int_0^t h(u)dS(u)\geq-\alpha \quad \forall t\in[0,T].$$

Definition C.4: Self-financing portfolio strategy

A portfolio strategy $h(t)$ is self-financing if:

$$dV(t;h)=\sum_{i=0}^{N}h_i(s)dS_i(t).$$

All these definitions change if we consider the **special case** for which: $S_0(t)=1$. In this case, the portfolio can be expressed as

$$h=[h_0,h_S(t)],$$

where h_0 is the part related to the risk free asset, and $h_S(t)$ to the risky assets. Then the previous three definitions can be expressed in the following way:

- **Definition C.2a**

  A value process is defined as

  $$V(t;h)=h_o(t)+\sum_{i=1}^{N}h_i(t)S_i(t).$$

- **Definition C.3b**

  A portfolio strategy h is admissible if and only if $\exists\alpha\in\mathbb{R}^+$ so that

  $$\int_0^t h_S(u)dS(u)\geq-\alpha \quad \forall t\in[0,T].$$

- **Definition C.4b**

  A portfolio strategy h is self-financing if:

$$dV(t;h) = \sum_{i=1}^{N} h_i(s) dS_i(t).$$

In this context, it also possible to give another important definition.

Definition C.5: Equivalent martingale measure

A probability measure Q on $\mathcal{F}(t)$ is called equivalent martingale measure, or only martingale measure, for the special market model, the numeraire S_0, and the time interval $[0,T]$, if it has the following properties.

- Q is equivalent to P on $\mathcal{F}(t)$.
- All price processes are martingales under Q on $[0,T]$.

Now, to reach the condition that guarantees the absence of arbitrage possibilities it is fundamental to highlight the connection between the two-market models specified above. We start considering the **relative price vector process,** called also normalized economy:

$$Z(t) = \frac{S(t)}{S_0(t)} = [Z_0, \dots, Z_N] = \left[1, \frac{S_1(t)}{S_0(t)}, \dots, \frac{S_N(t)}{S_0(t)}\right].$$

This relationship allow us to pass from a general S-process, for which the numeraire is a nonnegative number, to the special case of a Z-process, for which the numeraire is equal to one.

Then, if for each portfolio process we associate two processes, one in the S system and the other in the Z system, both admissible and self-financing, we can demonstrate that:

I. A portfolio h is S-self-financing if and only if it is also Z-self-financing.

II. The value processes V^S and V^Z are connected by

$$V^Z(t;h) = \frac{1}{S_0(t)} V^S(t;h)$$

Thanks to these two statement, whose proof can be found in Björk (2009)[1], it is possible to enunciate the First fundamental theorem of mathematical finance, which guarantees the absence of arbitrage in the studied model.

Theorem C.1: The First Fundamental Theorem

The market model is arbitrage free if and only if there exists a martingale measure, or in other words if $Q \sim P$, such that the processes in a normalized economy are (local) martingales under Q.

Thanks to Theorem C.1, we have stated the conditions under which there are no arbitrage possibilities.

Now, we have to demonstrate the completeness of the market model. To do that, we start giving the following Lemma that highlights the possibility to replicate a fixed contingent claim using a portfolio based on the underlying asset.

Lemma C.1: Replication of a contingent claim

Consider a given T-claim X, then fix an equivalent martingale measure Q and assume that the normalized claim $\frac{X}{S_0(T)}$ is integrable.

If the Q-martingale M, defined by

$$M(t) = E^Q\left[\frac{X}{S_0(T)}\middle|\mathcal{F}(t)\right]$$

admits the following integral representation:

$$M(t) = x + \sum_{i=1}^{N} \int_{o}^{t} h_i(s) dZ_i(s),$$

then X can be hedged in the S-economy.

Thanks to this lemma, it is possible to formulate the following theorem.

Theorem C.2: Second Fundamental Theorem

Assume that:

- the market is arbitrage free,
- S_0 is a fixed numeraire asset.

Then the market is complete if and only if the martingale measure Q, corresponding to S_0, is unique.

Once we have defined the conditions for the absence of arbitrage opportunities and the completeness of the market, we can rephrase the pricing problem under this new approach.

Our goal is always to determine a reasonable price process $\Pi(t;X)$ for a contingent claim X, which is equivalent to prove that the extended market, composed by $N+1$ a priori given risky assets plus $\Pi(t;X)$, is arbitrage free or rather that for this new market there exists a martingale measure Q. The following theorem proves exactly the absence of arbitrage for the extended market.

Theorem C.3: General Pricing Formula

The fair price for the price process $\Pi(t;X)$, where X is a contingent claim, is given by:

$$\Pi(t;X) = S_0(t)E^Q\left[\frac{X}{S_0(T)}\middle|\mathcal{F}(t)\right]$$

where Q is an equivalent martingale measure for $S(t)$ with S_0 as the numeraire.

Notice that Q is not necessarily unique and the choice of its value can lead to different price processes. In particular, it is possible to see that we can reduce the general pricing formula to the well-known risk valuation formula assuming that S_0 is the money account:

$$S_0(t) = S_0(0)e^{\int_o^t r(s)ds}$$

Where r is the short rate.

Theorem C.4: The Risk Neutral Valuation Formula

Let us assume the existence of a short rate r, the general pricing formula can be expressed as follow:

$$\Pi(t;X) = E^Q\left[e^{\int_o^t r(s)ds}X\middle|\mathcal{F}(t)\right]$$

where Q is an equivalent martingale measure for the money account as the numeraire.

The two main results coming from the martingale approach to APT are given by the following two theorems:

Theorem C.5: The Martingale Representation Theorem

Let W be a Wiener process with dimension δ, and assume that the filtration $\underline{\mathcal{F}}$ is defined as $\mathcal{F}(t) = \mathcal{F}_t^W, t \in [0, T]$.

Let M be any $\mathcal{F}(t)$-adapted martingale. Then there exist uniquely determined $\mathcal{F}(t)$ -adapted processes $h_1, \dots, h_d$ such that M has the following representation:

$$M(t) = M(0) + \sum_{i=1}^{N} \int_{o}^{t} h_i(s) dW_i(s) \quad t \in [0, T]$$

If the martingale M is square integrable, then $h_1, \dots, h_d \in \mathcal{L}^2$.

Theorem C.6: The Girsanov Theorem

Let W^P[72] be a standard Brownian motion of dimension δ on $\left(\Omega, \mathcal{F}, \mathbb{P}, \underline{\mathcal{F}}\right)$ and let φ any adapted column vector process, referred also as Girsanov kernel. Choose a fixed T and define the process L on $[0, T]$ by

$$dL(t) = \varphi(t) L(t) dW^P(t),$$

$$L(0) = 1.$$

Furthermore, assume that

$$E^P[L(T)] = 1,$$

and define a new probability measure Q on $\mathcal{F}(T)$ by

$$L(T) = \frac{dQ}{dP}.$$

Then

$$dW^P(t) = \varphi(t) dt + dW^Q(t)$$

[72] $W^P(t)$, which indicates a standard Brownian motion under the probability measure P, can be indicated also by $\overline{W}$.

where $W^Q(t)$ is a Wiener process under the probability measure Q.

This theorem is very important as it describes how a stochastic process change when we pass from the real probability measure into an equivalent one such as the risk neutral one.

C.2 The BSM under the martingale approach

Using the same notation of chapter one, first we define the following sample space $(\Omega, \mathcal{F}, \mathbb{P}, \underline{\mathcal{F}})$. Then we present the market model:

$$\begin{cases} dS(t) = \alpha(t)S(t)dt + \sigma(t)d\bar{W}(t) \\ dB(t) = r(t)B(t)dt \end{cases}$$

We require that the model is arbitrage free and this implies to find a martingale measure Q (see Theorem C.1).

We start by exploiting the Girsanov Theorem in order to identify an adapted process h to switch from a probability measure P to an equivalent one.
We begin defining:

$$dL(t) = h(t)L(t)d\bar{W}(t)$$

$$dQ = L(T)dP \quad on\ \mathcal{F}(T)$$

Then, if h is a Girsanov kernel:

$$d\bar{W}(t) = h(t)d(t) + dW(t)$$

where $W(t)$ is a Wiener process under the equivalent probability measure Q.
Now, we substitute this result into the equation for the market model obtaining:

$$dS(t) = \ \alpha(t)S(t)dt + \sigma(t)[h(t)d(t) + dW(t)]$$

$$dS(t) = [\ \alpha(t) + \ \sigma(t)h(t)]S(t)dt + \sigma(t)S(t)dW(t).$$

At this point, we have to find the value for which h is an equivalent martingale measure. This is done using the evidence that come from the First Fundamental Theorem, which affirms that Q is a martingale measure if the local rate of return under Q is equal to the short rate $r(t)$. So, in our case

$$r(t) = \alpha(t) + \sigma(t)h(t)$$

It is easy to compute the value of $h(t)$, which is:

$$h(t) = -\frac{\alpha(t) - r(t)}{\sigma(t)}.$$

Nevertheless, it is also immediate to recognize the formula of the risk premium of the stock, usually indicated with λ.
In light of this, we can affirm that the BSM model is arbitrage free.

The next step is the pricing of a derivative using the BSM model under the martingale approach.
Let us consider a simple contingent claim of the form: $X = \Phi\big(S(T)\big)$. Our target is to find out an arbitrage free process $\Pi(t; X)$, see Theorem C.4, for the defined contingent claim.
We relate only the theorems that bring us to the final solution. All the calculations made to reach that theorems are not reported (for more details see Björk (2009)[1]).

Theorem C.6: Solution of the BSM PDE

Let us consider the contingent claim defined above. To avoid any arbitrage possibilities, the pricing function $F(t, x)$ must solve the following system of equations:

$$\begin{cases} F_t(t,x) + rS(t)F_x(t,x) + \dfrac{1}{2}\sigma^2 S^2(t)F_{xx}(t,x) - rF(t,x) = 0 \\ F(T,x) = \Phi(x) \end{cases}$$ [73]

[73] Remember that $x = S(t)$

Theorem C.7: The Risk Neutral Evaluation Formula

The pricing function $F(t,x)$ has the following representation:

$$F(t,x) = e^{-r(T-t)} E_{t,x}^{Q}\left[\Phi\left(\mathrm{S(T)}\right)\right]$$

The expected value is taken w.r.t the martingale measure Q. Remember that $S(t)$ is distributed as:

$$dS(t) = r(t)S(t)dt + \sigma(t)S(t)dW(t).$$

MATLAB CODES

D.1 Inversion algorithms

Gaver-Stefhest inversion algorithm

```
function GSlt=gavsteh(f,t,M)

%% Gaver-Stefhest inversion algorithm
%
%   f        The name of the function to be transformed.
%   t        The transform argument (usually a snapshot of time).
%   ilt      The value of the inverse transform
%   L         Optional number of coefficient ---> depends on computer word length
%             used
%            (examples: L=8, 10, 12, 14, 16, so on..)
%
%Reference: http://www.mathworks.com/matlabcentral/fileexchange/9987-gaver-
stehfest-algorithm-for-inverse-laplace-transform

% Set M to 18 if user didn't specify an M.
if nargin < 3

   M = 18;
end


nn2 = M/2;
nn21= nn2+1;


for n = 1:M
   z = 0.0;
   for k = floor( ( n + 1 ) / 2 ):min(n,nn2)
      z = z + ((k^nn2)*factorial(2*k))/ ...
         (factorial(nn2-k)*factorial(k)*factorial(k-1)* ...
         factorial(n-k)*factorial(2*k - n));
   end
   v(n)=(-1)^(n+nn2)*z;
end


sum = 0.0;
```

```
ln2_on_t = log(2.0) / t;
for n = 1:M
   p = n * ln2_on_t;
   sum = sum + v(n) * f(n*ln2_on_t);
end
   GSlt = sum * ln2_on_t;
```

Euler inversion algorithm

```
function Eul = euler_inversion(f_s, t, M)
%%Euler inversion algorithm
%
%     f_s      Handle to function of s
%      t       Times at which to evaluate the inverse Laplace transformation of f_s
%      M       Optional, number of terms to sum for each t (32 is a good guess);
%              highly oscillatory functions require higher M, but this can grow
%              unstable; see test_talbot.m for an example of stability.
%
% Copyright 2012, The MathWorks, Inc. by Tucker McClure

   % Make sure t is n-by-1.
   if size(t, 1) == 1
      t = t';
   elseif size(t, 2) > 1
      error('Input times, t, must be a vector.');
   end

   % Set M to 32 if user didn't specify an M.
   if nargin < 3
      M = 32;
   end

   % Vectorized Euler's algorithm
   bnml = @(n, z) prod((n-(z-(1:z)))./(1:z));

   xi = [0.5, ones(1, M), zeros(1, M-1), 2^-M];
   for k = 1:M-1
      xi(2*M-k + 1) = xi(2*M-k + 2) + 2^-M * bnml(M, k);
   end
   k = 0:2*M; % Iteration index
   beta = M*log(10)/3 + 1i*pi*k;
   eta  = (1-mod(k, 2)*2) .* xi;
```

```
    % Make a mesh so we can do this entire calculation across all k for all
    % given times without a single loop (it's faster this way).
    [beta_mesh, t_mesh] = meshgrid(beta, t);
    eta_mesh = meshgrid(eta, t);

    % Finally, calculate the inverse Laplace transform for each given time.
    Eul =    10^(M/3)./t ...
         .* sum(eta_mesh .* real(arrayfun(f_s, beta_mesh./t_mesh)), 2);

end
```

Talbot inversion algorithm

```
function Tal = talbot_inversion(f_s, t, M)
%%Talbot inversion algorithm
%
%    f_s      Handle to function of s
%     t       Times at which to evaluate the inverse Laplace transformation of f_s
%     M       Optional, number of terms to sum for each t (64 is a good guess);
%             highly oscillatory functions require higher M, but this can grow
%             unstable; see test_talbot.m for an example of stability.
%
% Copyright 2012, The MathWorks, Inc. by Tucker McClure

    % Make sure t is n-by-1.
    if size(t, 1) == 1
        t = t';
    elseif size(t, 2) > 1
        error('Input times, t, must be a vector.');
    end

    % Set M to 64 if user didn't specify an M.
    if nargin < 3
        M = 64;
    end

    % Vectorized Talbot's algorithm

    k = 1:(M-1); % Iteration index

    % Calculate delta for every index.
    delta = zeros(1, M);
```

```
    delta(1) = 2*M/5;
    delta(2:end) = 2*pi/5 * k .* (cot(pi/M*k)+1i);

    % Calculate gamma for every index.
    gamma = zeros(1, M);
    gamma(1) = 0.5*exp(delta(1));
    gamma(2:end) =    (1 + 1i*pi/M*k.*(1+cot(pi/M*k).^2)-1i*cot(pi/M*k))...
                .* exp(delta(2:end));

    % Make a mesh so we can do this entire calculation across all k for all
    % given times without a single loop (it's faster this way).
    [delta_mesh, t_mesh] = meshgrid(delta, t);
    gamma_mesh = meshgrid(gamma, t);

    % Finally, calculate the inverse Laplace transform for each given time.
    Tal = 0.4./t .* sum(real(   gamma_mesh ...
                      .* arrayfun(f_s, delta_mesh./t_mesh)), 2);

end
```

Comparison 1

Comparison among the GS, Euler and Talbot inversion algorithms changing the value of the coefficient M.

```
function [ Inv_Laplace_Trans,tempo ]= invLT_table( funct, t, M_gv, M_Eul, M_Tal )
%%Comparison among the three inversion algorithms (GS, Euler and
%   Talbot)
%   This function returns the inverse of the Laplace Transform using
%   the Gaver-Stefhest, Euler and Talbot algorithm and their CPU
%   execution time.
%
%   funct        is the function I wanto to invert
%     t          the time at which I wanto to invert
%   [L_gv]       vector of the coefficents in the Gaver-Stefhest Algorithm
%                (optional)
%   [M_Eul]      vector of the coefficients in the Euler Algorithm (optional)
%   [M_Tal]      vector of the coefficients in the Talbot Algorithm (optional)

dimLgv=length(M_gv);
GV=zeros(dimLgv,1);
Eul=zeros(dimLgv,1);
Tal=zeros(dimLgv,1);
tempo=zeros(dimLgv,3);
Inv_Laplace_Trans=zeros(dimLgv,3);
```

```
 for i=1:dimLgv
    tic
    GV(i)=gavsteh(funct,t,M_gv(i));
    tempo(i,1)=toc;
    tic
    Eul(i)=euler_inversion(funct, t, M_Eul(i));
    tempo(i,2)=toc;
    tic
    Tal(i)=talbot_inversion(funct, t, M_Tal(i));
    tempo(i,3)=toc;

    Inv_Laplace_Trans(i,:)=[GV(i) Eul(i) Tal(i)];
end

Inv_Laplace_Trans=[Inv_Laplace_Trans, tempo];

end
```

Comparison 2

Comparison among the GS, Euler and Talbot inversion algorithms changing the value of the coefficient t.

```
function [ Inv_Laplace_Trans,tempo ] = invLT_t_var( f_tr, f_real, t, L_gv, L_Eul,
L_Tal)
%%Performance of GS, Euler and Talbot algorithm varying t
%This function returns the inverse of the Laplace Transform varying t
%   f_tr      is the function I wanto to invert
%   f_real    is the function real value of the inverse
%   t         the time at which I wanto to invert
%   [L_gv]    the coefficents in the Gaver-Stefhest Algorithm (optional)
%   [M_Eul]   the coefficients in the Euler Algorithm (optional)
%   [M_Tal]   the coefficients in the Talboto Algorithm (optional)

dimt=length(t);
GV=zeros(dimt,1);
Eul=zeros(dimt,1);
Tal=zeros(dimt,1);
realval=zeros(dimt,1);
tempo=zeros(dimt,3);
Inv_Laplace_Trans=zeros(dimt,4);

for i=1:dimt
    tic
    GV(i)=gavsteh(f_tr,t(i),L_gv);
```

```
    tempo(i,1)=toc;
    tic
    Eul(i)=euler_inversion(f_tr, t(i), L_Eul);
    tempo(i,2)=toc;
    tic
    Tal(i)=talbot_inversion(f_tr, t(i), L_Tal);
    tempo(i,3)=toc;
    realval(i)=f_real(t(i));

    Inv_Laplace_Trans(i,:)=[realval(i), GV(i), Eul(i), Tal(i)];
end

% Plot to see how the estimation of the three algorithms change varying t
plot(t(:),GV(:), 'o', t(:), Eul(:),'*', t(:), Tal(:),'+', t(:), realval(:), 'k');
legend('Gaver-Stefhest','Euler','Talbot','real value');
xlabel('t');
ylabel('f(t)');
title ('f05 from F05 using the three algorithms ');
colormap autumn
end
```

D.2 Floating lookback options

Cox-Rubistein-Ross binomial tree

```
function [ PriceFLBCallBinTree ] = FLBCall_BT( Spot, T, r, q, sigma, N )
%% Cox-Rubistein-Ross Binomial tree
%  This function calculates the price of a standard floating lookback using
%  the CRR binomial tree

%Definition of the variables
dt = T/N;
u = exp(sigma*sqrt(dt));
d = 1/u;
p = (exp((r-q)*dt) - d)/(u-d);

for i=1:N+1
    for j=i:N+1
        V(i,j) = 1 - d^(j-i);
        if j==N+1
        Option(i,j) = V(i,j);
        end
    end
end
```

```
for j=N:-1:1
   for i=1:j
       if i~=j
         Option(i,j) = exp(-r*dt)* (p*Option(i,j+1)*u + (1 - p)*Option(i+2,j+1)*d);
       else
         Option(i,j) = exp(-r*dt)* (p*Option(i,j+1)*u + (1 - p)*Option(i+1,j+1)*d);
       end
    end
end

PriceFLBCallBinTree = Spot*Option(1,1);
```

Monte Carlo simulation

```
function [ MonteCarloLBCall, muci ] = MCLBCall( Spot, alpha, rate, div, sigma, T,
N, Rep )
%%Price of a floating lookback call option calculated by Monte Carlo method
%
%      N         Number of time steps
%    Rep        Number of replication

price = zeros(Rep,1);
for i=1:Rep
   PathS = BMMC( Spot, T, rate, div, sigma, N, 1 );
   minS=min(PathS);
   price(i) = exp(-rate*T)*(PathS(end)- (alpha*minS));
end

[MonteCarloLBCall, ~, muci]=normfit(price);

end
```

Laplace Transform approach

```
function [ LaplaceTransFLBCall ] = LFTLBCall( Spot, min, alpha, Time_mat,
sigma, rate, div  )
%%Price of a floating lookback call option using the Laplace Transform approach,
%   see paragraph 3.2.3, equation 3.12.

%   Definition of the functions of the Laplace Transform formula 3.12
v1=@(lambda) (1/sigma^2)*(-(div-rate-0.5*sigma^2)+sqrt((div-rate-0.5*...
sigma^2)^2+2*sigma^2*(lambda+div)));
```

```
v2=@(lambda) (1/sigma^2)*(-(div-rate-0.5*sigma^2)-sqrt((div-rate-0.5*...
sigma^2)^2+2*sigma^2*(lambda+div)));
phi_1=@(lambda) (Spot*v1(lambda))/(v1(lambda)-v2(lambda))*(lambda/...
(lambda+div)+(1-v1(lambda))/v1(lambda)*lambda/(lambda+rate))*...
((alpha*min/Spot)^v2(lambda)-alpha^(v2(lambda)-v1(lambda))*v2(lambda)/...
 v1(lambda)*(alpha*min/Spot)^v1(lambda));
phi_2=@(lambda) (Spot*v2(lambda))/(v1(lambda)-v2(lambda))*(lambda/...
(lambda+div)+(1-v2(lambda))/v2(lambda)*lambda/(lambda+rate)-...
alpha^(v2(lambda)-v1(lambda))*(lambda/(lambda+div)+(1-...
v1(lambda))/v1(lambda)*lambda/(lambda+rate)))*((alpha*min/Spot)^v1(lambda));

if (Spot<alpha*min)
LTfl=@(lambda) phi_1(lambda)/lambda;
else
   LTfl=@(lambda) (phi_2(lambda)+lambda*Spot/(lambda+div)-lambda*...
   alpha*min/(lambda+rate))/lambda;
end

%   Inversion of the formula 3.12
LaplaceTransFLBCall=invLT_table(LTfl, Time_mat, 18,32,64);

end
```

Closed-form formula

```
function [ AnLookbackFrOptCall, tempo ] = FLCall_an( Spot , Time_mat, t, sigma,
rate, div, min, alpha )
%%  Price of a fractional call Lookback
%    This function calculates the analytical price of a fractional
%    call lookback reported in Chapter 3, see formula 3.11.
%    The formula is taken by Kimura(2007)[27].

%If alpha is not define we set the value equal to 1
if nargin < 8
   alpha = 1;
end
strike_new=alpha*min;
delta_t=Time_mat-t;

%Define the variables
gamma=2*(rate-div)/(sigma^2);
sig=(sigma^(2))/2;
d_p=(log(min/strike_new)+(rate-div-sig)*delta_t)/(sigma*sqrt(delta_t));
d_n=(log(min/strike_new)-(rate-div+sig)*delta_t)/(sigma*sqrt(delta_t));
```

```
tic
part1=BSMcall(Spot,strike_new,delta_t,sigma,rate,div);
if div~=rate
   part2=((alpha*Spot)/gamma)*((exp(-
rate*delta_t)*(min/Spot)^(gamma)*normcdf(d_p))-...
        (exp(-div*delta_t)*(alpha^(gamma))*normcdf(d_n)));
else
   part2=alpha*Spot*exp(-rate*delta_t)*sigma*sqrt(delta_t)*(d_p*normcdf(d_p)+...
       normpdf(d_p));
end
time(1)=toc;

AnLookbackFrOptCall=[part1+part2 , time];

end
```

D.3 Arithmetic Asian call options

The Monte Carlo simulation and the Laplace Transform approach functions refers to an continuous arithmetic Asian call option.

Monte Carlo Simulation

```
function [ P, ConfInt ] = APOcv( S0, K, T, r, sigma, N, R1, R2 )
%Pricing of an arithmetic Asian call option using MC with control variates

%First cycle
Paths1=BMMC(S0, T, r, sigma, N, R1);
Y=sum(Paths1,2);
Payoff1=max(mean(Paths1(:,(2:N+1)),2)-K,0).*exp(-r*T);
varY=var(Y);
cov_PY=cov(Payoff1,Y);
c_opt=-(cov_PY(2,1)/varY);

%Second cycle
dt=T/N;
Payoff_cv=zeros(R2,1);
ExpY=S0*(1-exp(r*(N+1)*dt))/(1-exp(r*dt));
for i=1:R2
   Paths2=BMMC(S0, T, r, sigma, N, 1);
   Payoff_r = exp(-r*T)*max(mean(Paths2(2:N+1))-K, 0);
   Payoff_cv(i)= Payoff_r + c_opt*(sum(Paths2)-ExpY);
end
```

```
[P, std, ConfInt]=normfit(Payoff_cv);

end
```

Laplace Transform approach

```
function LTA=ltprice(cx, Spot, strike, expiry, today, average,sigma,rf,l)
%% Algortihm to define the Laplace Transform of an arithmetic asian call option
%  (Translation of the VBA Fusai's Code)

%  Definition of the variables
   v=(2*rf)/(sigma^(2))-1;
   q=(sigma^(2))*(strike*expiry-today*average)/(4*Spot);
   mu=sqrt(v^2+2*l);
   pow=0.5*(mu-v-4);

   if real(l)>=8+4*v
      term1=-1/(2*q)*cx;
      term2=pow*log((1/(2*q))*cx);
      powx3=0.5*(mu+v+2);
      term3=powx3*log(1-cx);
      loggamma=cgammln(0.5*(mu-v-2));
      den=log(l-2-2*v)+log(l);
      den=den+loggamma;

      funct=exp(term1+term2+term3-den);
   end
  if real(l)>=(2+2*v) && real(l)<(8+4*v)
         cx=exp(1/(1+pow)*log(cx));
         term1=-1/(2*q)*cx;
         powx3=0.5*(mu+v+2);
         term3=log(1-cx)*powx3;
         loggamma=cgammln(0.5*(mu-v-2));
         den=log(l-2-2*v)+log(l);
         den=den+loggamma;
         den=den+log(2*q)*pow;

         funct= exp(term1+term3-den)/(1+pow);

  end
      LTA=funct/(2*q);

end
```

```
function LaplaceTransfAsian_GY=lt_gemanyor(Spot, strike, expiry, today, average,
sigma, rf, l, n)
% We calculate the value of the integral in formula 5.6 and then we join these results
% with the ones obtained by the function ltprice.
%  (Translation of the VBA Fusai's Code)


   sum_r=0;
   sum_i=0;


   xw_gaussian=gaulegLT(0,1,n);


   for i=1:n
      xu(i)=xw_gaussian(i,1);
      wu(i)=xw_gaussian(i,2);
      sum_r=sum_r+wu(i)*real(ltprice(xu(i),Spot, strike, expiry, today, average,
sigma, rf, l));
      sum_i=sum_i+wu(i)*imag(ltprice(xu(i),Spot, strike, expiry, today, average,
sigma, rf, l));
   end
   LaplaceTransfAsian_GY=complex(sum_r,sum_i);
end



function AsianCallGY=AS_GY(Spot, strike, rf, sigma, expiry, today, average, aa ,
terms, totterms, n)
%  Inversion of the Laplace transform price of an Arithmetic Asian call option
%  exploiting the Euler algorithm defined in chapter 5, see formula 5.6.
%  (Translation of the VBA Fusai's Code)


Euler=0;
tau=expiry-today;
h=sigma^2*tau/4;
sum_r=zeros(1,totterms-terms+1);


%We compute all MaxK termini to be used in the Euler algorithm
%Then we save the partial sums in a vector to be used to extrapolate the final result


sum = 0.5* lt_gemanyor(Spot, strike, expiry, today, average, sigma, rf, aa / (2 *
h),n);

for k = 1:totterms
   term = ((-1) ^ (k))* lt_gemanyor(Spot, strike, expiry, today,
        average, sigma, rf, complex(aa / (2 * h), k * pi / h), n);
```

```
    sum = term+sum;
    if terms <= k
       sum_r(k – terms + 1) = real(sum);
    end
end


for k = 0☹totterms – terms)
    Euler = Euler + nchoosek(totterms – terms, k) * ((2) ^ (
            -(totterms – terms))) * sum_r(k + 1);
end

SumAW = exp(aa / 2) * Euler / h;

    AsianCallGY = exp(-rf * (expiry – today)) * 4* Spot * SumAW / (expiry * sigma
* sigma);
end
```

Complex functions

Complex functions needed in the *ltprice.m* code.

```
function Sol=gaulegLT(x1, x2, n )
%% Gaussian quadrature function for complex functions

EPS = 0.00000000003;
m = (n + 1) / 2;
xm = 0.5 * (x2 + x1);
xl = 0.5 * (x2 - x1);

for i = 1:m
    z = cos(pi * (i - 0.25) / (n + 0.5));
    b=true;
    while b
          p1 = 1;
          p2 = 0;
          for j = 1:n
            p3 = p2;
            p2 = p1;
            p1 = ((2 * j - 1) * z * p2 - (j - 1) * p3) / j;
          end
        pp = n * (z * p1 - p2) / (z * z - 1);
        z1 = z;
        z = z1 - p1 / pp;
        b=abs(z-z1)>EPS;
```

```
    end

    x(i) = xm - xl * z;
    x(n + 1 - i) = xm + xl * z;
    w(i) = 2 * xl / ((1 - z * z) * pp^2);

    w(n + 1 - i) = w(i);

end

    Sol = [x',w'];

end



function loggammacomplex=cgammln(num)
%% Function to calculate the logarithm of the gamma function of complex numbers

    cof(1) = 76.1800917294715;
    cof(2) = -86.5053203294168;
    cof(3) = 24.0140982408309;
    cof(4) = -1.23173957245015;
    cof(5) = 0.120865097386618 * 10 ^ -2;
    cof(6) = -0.5395239384953 * 10 ^ -5;

    sq2pg= 2.506628274631;

    y=num;
    c5 =5.5;
    tmp = num+c5;
    tmp = num+ 0.5* log(tmp)- tmp;
    ser = 1.00000000019001;

    for j=1:6
        y = y + 1;
        ser = ser+cof(j)/ y;
    end

    ser = real(sq2pg* ser);
    loggammacomplex =log(ser/ num)+ tmp;
end
```

BIBLIOGRAPHY

[1] T. Björk, Arbitrage Theory in Continuous Time, United States of America: Oxford University Press, 2009.

[2] S. Shreve, Stochastic Calculus for Finance, Continuous Time Models, vol. II, United States of America: Springer, 2004.

[3] J. L. Schiff, The Laplace Transform: theory and applications, New York: Springer-Verlag, 1999.

[4] U. Graf, Applied Laplace Transform and z-Transform for Scientists and Engineers, Basel: Springer Basel AG, 2004.

[5] A. M. Cohen, Numerical Methods for Laplace Transforms Inversion, vol. V, United States of Ameria: Springer, 2007.

[6] R. Churchill and J. Brown, Complex Variables and Applications, United States of America: The McGraw-Hill Companies, 2009.

[7] M. R. Spiegel, Schaum's Outlines: Laplace Transforms, United States of America: The McGraw-Hill Companies, 1965.

[8] C. J. Wellekens, "Generalization of Vlach's method for the numerical inversion of the Laplace Transform," *Electronic Letters,* vol. 6, pp. 742-744, 1970.

[9] J. Abate and P. Valkó, "Multi-precision Laplace inversion," *International Journal for Numerical Methods in Engineering,* vol. 60, pp. 979-993, 2004.

[10] D. Gaver, "Observing stochastic processes and approximate transform inversion," *Operations Research,* vol. 14, pp. 444-459, 1966.

[11] H. E. Salzer, "Formulas for the Partial Summation of Series," *Mathematics of Computation,* vol. 10, pp. 149-156, 1956.

[12] H. Stehfest, "Algorithm 368: Numerical Inversion of Laplace Transforms," *Communications of the ACM,* vol. 13, pp. 47-49, January 1970.

[13] A. Kuznetsov, "On convergence of Gaver-Stehfest Algortihm," *SIAM Journal on Numerical Analysis,* vol. 51, no. 6, pp. 2984-2998, 2013.

[14] J. Abate, G. L. Choudhury and W. Whitt, "An introduction to numerical inversion and its application to probability models," in *Computational Probability*, Boston, Springer, 1999, pp. 257-323.

[15] T. Sakurai, "Numerical Inversion for Laplace Transform of Function with Discontinuities," *Journal of Applied Probability and Advances in Applied Probability,* vol. 36, no. 2, pp. 616-642, 2004.

[16] A. Murli and M. Rizzardi, "Algorithm 682: Talbot's Method for the Laplace Inversion Problem," *ACM Transctions on Mathematical Software,* vol. 16, no. 2, pp. 158-168, 1990.

[17] P. Wilmott, S. Howison and J. Dewynne, The Mathematics of Financial Derivatives, Cambridge: Cambridge University Press, 1995.

[18] H. Geman and M. Yor, "Pricing and Hedging Double-Barrier Options: A probabilistic Approach," *Mathematical Finance,* vol. 6, no. 4, pp. 365-378, 1996.

[19] A. Pelsser, *Pricing Double Barrier Options: An Analytical Approach,* 1997.

[20] F. Spritzer, "A combinatorial lemma and its application to probability theory," *Transactions of the American Mathematical Society,* vol. 82, no. 2, pp. 323-339, 1956.

[21] B. Davis and B. Martin, "Numerical Inversion of the Laplace Transform: A survey and comparison of methods," *Journal of Computational Physics,* vol. 33, no. 1, pp. 1-32, 1979.

[22] G. Petrella and S. Kou, "Numerical pricing of discrete barrier and lookback options via Laplace transforms," *The Journal of Computational Finance,* vol. 8, no. 1, pp. 1-35, 2004.

[23] A. Ohgren, "A remark on the pricing of discrete lookback options," *Journal of Computational Finance,* vol. 4, pp. 141-147, 2001.

[24] G. Fusai, "Applications of Laplace Transform for Evaluating Occupation Time Options and Other Derivatives," October 2000. [Online]. Available: http://go.warwick.ac.uk/wrap/2706.

[25] M. Y. Hélyette Geman, "Bessel processes, Asian Options, and Perpetuities," *Mathematical Finance,* vol. 3, no. 4, pp. 349-375, 1993.

[26] D. Dufresne, "https://minerva-access.unimelb.edu.au/," 3 November 2004. [Online]. Available: http://hdl.handle.net/11343/33807.

[27] T. Kimura, *American Fractional Lookback Options: Valuation and Premium Decomposition,* Sapporo: Hokkaido University, 2007.

[28] M. Yor, Exponential Functionals of Brownian Motion and Related Processes, New York: Springer-Verlag, 2001.

[29] B. Josso and L. Larsen, June 2012. [Online]. Available: http://www.kappaeng.com/PDF/Laplace_transform_numerical_inversion.pdf.

[30] J. AbatE and W. Whitt, "A Unified Framework for Numerically Inverting Laplace Transform," *INFORMS Journal on Computing,* vol. 18, no. 4, pp. 408-421, 2006.

[31] T. McClure, "Examples of Numerical Approximation of the Inverse Laplace Transform," The Mathworks Inc., 2012.

[32] Z. L. Krougly, D. J. Jeffrey and D. Tsarapkina, "Software Implementation fo Numerical Algorithms in Arbitrary Precision," in *15th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, Timosoara, 2013.

[33] M. C. Fu, D. B. Madan and T. Wang, "Pricing Continuous Asian Options: A Comparison of Monte Carlo and Laplace Transform Inversion Method," *Journal of Computational Finance,* vol. 2, no. 2, pp. 49-74, 1999.

[34] L. C. G. Rogers and Z. Shi, "The value of an Asian option," *Journal of Applied Probability,* vol. 32, pp. 1077-1088, 1992.

[35] G. W. P. Thompson, *Topics in Mathematical Finance,* Queens' College, 1999.

[36] J. Abate and W. Whitt, "Numerical Inversion of Laplace Transform of Probability Distributions," *Journal of Computing,* vol. 7, pp. 36-43, 1995.

[37] P. K. Kythe and M. R. Schäferkotter, Handbook of Computational Methods for Integration, United States of America: Chapman&Hall/CRC, 2004.

[38] S. G. Krantz, Handobook of Complex Variables, Boston : Birkhauser, 1999.