Doctoral Thesis

ECONOPHYSICS METHODS FOR EXOTIC OPTION PRICING: FROM THEORY TO IMPLEMENTATION (経済物理学のアプローチによる複雑オプションの価格付け―理論から実装まで―)

Aurelien Cassagnes

ECONOPHYSICS METHODS FOR EXOTIC OPTION PRICING: FROM THEORY TO IMPLEMENTATION

AURELIEN CASSAGNES
Master Computer Sciences

A THESIS SUBMITTED FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

DEPARTMENT OF HUMAN AND ENGINEERED ENVIRONMENTAL STUDIES
UNIVERSITY OF TOKYO

2015

Declaration

I hereby declare that this thesis is my original work and it has been written by me in its entirety. I have duly acknowledged all the sources of information which have been used in the thesis.

This thesis has also not been submitted for any degree in any university previously.

Aurelien Cassagnes
February 2015

Acknowledgments

I would like to express my gratitude first to Prof. Yu Chen that granted me the opportunity and then trusted me to do this Ph.D under his supervision. His optimism and critical interdisciplinary insights have been a major help in my work. He managed for three long years to not laugh at my fuzziest ideas, and waited until they became crisper; for this patience I would like to thank him deeply. I must also thank him for having the optimal words to lift my mood prone to swift swings at the hardest point of this Ph.D curriculum.
I would like to express my thanks to Prof. Hirotada Ohashi for his kindness and support. By providing the necessary environment and technical means to this thesis, he is a major actor of this thesis and should be thank as such.
I would like to thank also Prof. Ao Ping from the Shanghai SJTU University that allowed me to enjoy twice their brilliant research group. The discussions I had with Ruoshi Yuan and Tang Ying have been key for me, therefore I would like to thank them.

Then, I would like to thank all the professors that along my rather peculiar academic adventure contribute to the present results. Maybe the most important professor I had, Prof. Christian Gboy, a brilliant educator and all around good man. His kindness, simplicity and unique way to demystify the pompous rhetoric usually surrounding mathematics was the key to me going into graduate level studies. If it were not for him taking care of left out students in my hometown, I would just not be writing those lines. He is thanked with all my heart.
Professor Kenji Doya from Okinawa was also important in the path that led me to this thesis completion. He took me under his supervision as an engineer and gave me the proper environment to foster the necessary hunger for pursuing an academic degree in such a prestigious university as the University of Tokyo. I thank him for his kindness and unparalleled skill to remain humble while being at the same time so brilliant. Professor Jacques Delaunay from the University of Tokyo who gave me the first taste of a research environment, and was also my first gateway to Japan that turned out to be such a welcoming place for me. I also would like to thank him for the informal coffee we shared, providing me opportunities to vent out some excessive stress *en Francais*. Professor Pierre Alain Toupance from my time at IUT.A Lyon has also been key to my current situation. By not punching me when I asked every questions that could be asked, and then some more, during his math class he proved to me that no mathematics question is really dumb when you are learning.

Less formally now, I would like to thank the many people that have been as many companion in my adventure: Alexandre, Asatani, Benoit, Liu, Lolo, Marie, Masako, Mayu, Michel, Miho, Victor, the Hongo Starbucks, etc.

Finally, I thank my family and my brother Alban to whom this thesis is dedicated. For their support and for just being there and still standing I thank them.

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 The problem of exotic option pricing

The fast-paced world we live in has faced a transition where the financial system has changed from being a relatively self contained entity with little need for extra complexity, to a place where financial agents are highly interconnected and with sometimes diverging motivation, where increasingly sophisticated tools are required throughout the industry. Those financial instruments among which the exotic derivatives belong have built-in complexity which is a specific answer to an investor with very specific need. When investors were in need of a tool to hedge risk bankruptcy for some risky loans, credit default swap was introduced; weather derivatives were introduced as a way to protect oneself against extreme weather events. Complexity in the environment of financial derivatives is a side effect of the ongoing adjustment between investors challenges and industry propositions.

In this thesis we will focus on the pricing problem for the so-called exotic options. They are difficult to categorize but, it is usually a term that encompasses all option derivative contracts whose payoff is path-dependent. One can think about

- Asian options: The payoff of this contract on exercise is dependent on either a geometric, an arithmetic, or even a harmonic average of the asset lifetime prices.
- Barrier options: The payoff, whatever it is specified to be, will be nullified (knock-out) if the asset reaches a specified level called barrier level. The other flavor is the "knock-in" option where the payoff is null unless the asset reaches this barrier level.
- Lookback options: The option if exercised at maturity date will pay a function on the maximum of the asset historical prices.

In other words, the value of an exotic option contract will not only depend on an underlying security value, but also on the way that this particular value was reached. Because of their sophisticated payoff structure, they pose considerable challenge when trying to work out their fair value. The fair value for an option being such that it does not introduce any riskless profit opportunity for agents in the financial system. Otherwise it would be hard to argue that the financial world is an efficient system operating in a state of equilibrium.

It is then of critical importance that any derivative contracts being released for a specific use case, is delivered along with a pricing formula. Relying on simulation to price options for which a formula has not been found should only be a temporary solution for they lead to fewer insights than an analytical formula.

1.2 Challenges ahead

The future evolution of a risky asset can be (to a large extent) modeled as random: one can know for sure the value of a ton of cocoa as of now, yet be lost as to the price for the same asset tomorrow. To harness this first technical problem, a set of assumptions are cast onto the asset dynamic. This leads to a model such as the Black-Scholes model for the evolution of an asset, a model that we hope is tractable enough to let us work with the random character of our asset. As can be seen in Fig.1.1 even though all paths are obviously random we can see that their distribution obeys a rather normal-enough curve. Looking back at the derivative contract description we can extract the payoff function connected to this contract. If the payoff can be computed from the underlying asset final price then the

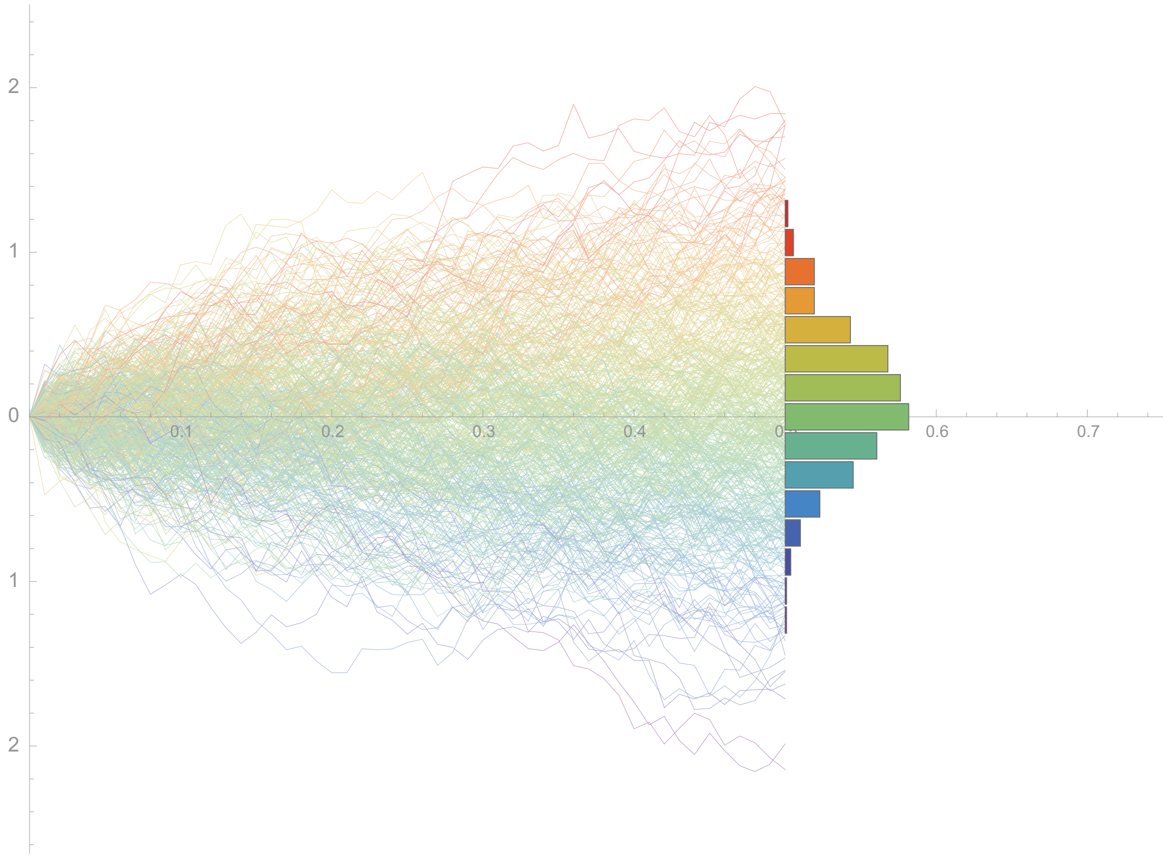


Figure 1.1: A set of risky assets evolving according to Black-Scholes assumptions, and their intermediate binned distribution.

challenge is trivial, since our assumed model tells us about discrete tick-time distribution. However, if the payoff is written in a way that without knowing the whole path we can not answer "what is the payoff on this asset ?", then we are left with a major technical challenge: how can I calculate the probability that an asset take a singular path between the infinitely many paths that I can draw between two points ? In Fig.1.2, even though any of those paths posess the same final value, they each would have a different payoff on their minimum value until expiration !

This is the real challenge, doing computation on individual paths when they are infinitely many of them and each possibly different in regard to our payoff function.

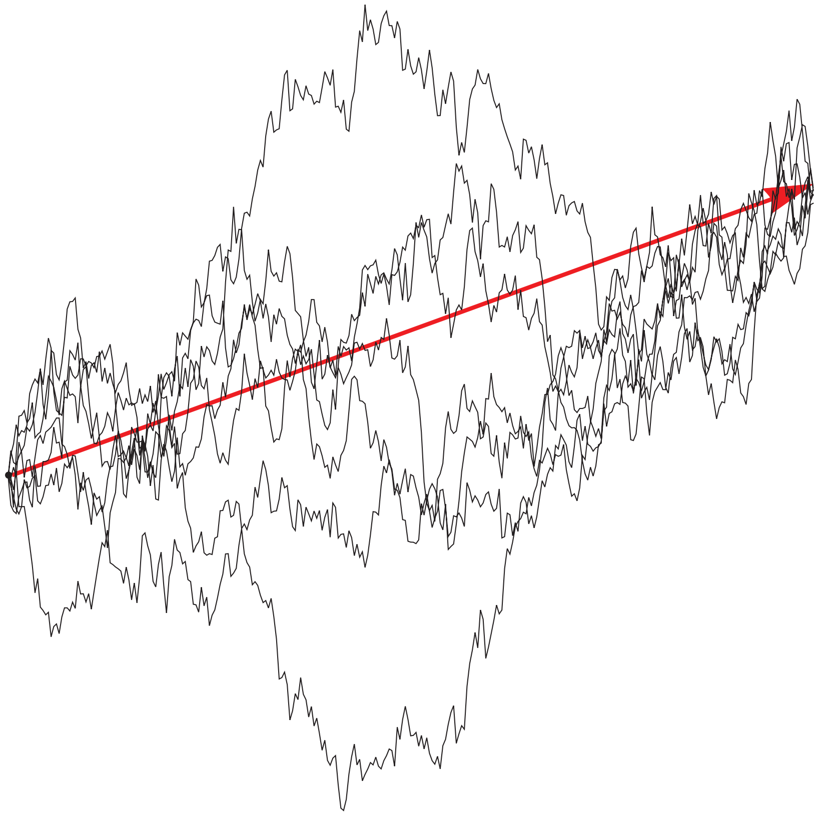


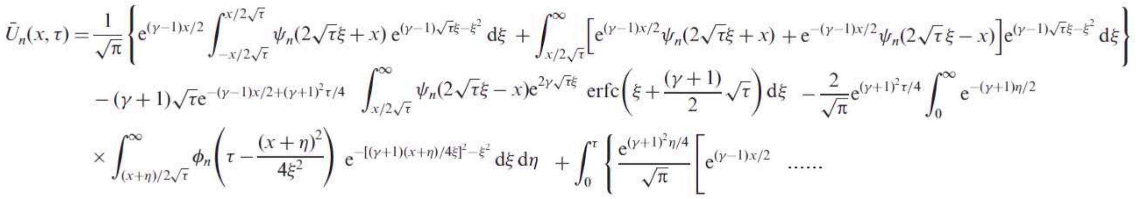
Figure 1.2: Between two values for a risky asset, said asset can take any number of paths, here we give a very limited sample of the possible paths.

1.3 Mainstream option pricing strategies

The exotic option pricing problem is connected to more than a single mathematical field or subfield. Obviously none of those fields can be really said to be disconnected from the rest and one could argue that they both derive from measure theory, but they are arguably still quite differentiated enough that we can point them out as standing on their own. We can cite as relevant to attack the pricing problem:

- Probability theory: Using either a martingale approach, or by working out explicitly the distribution for some functionals it is possible to do computation related to the expectation of the payoff. Monte Carlo simulation can also be used to compute an estimation of the fair price when no formula are available, or are computationally costly.
- Stochastic calculus: A stochastic differential equations describing the evolution of a function on the stochastic process under consideration is written. Then using the Markov condition, an ODE or PDE is extracted and solved.

An important share of the relevant literature usually attack the problem from one point of view and do not mix approach, or at the very least do not attempt to bring in concepts from outside of those two branches : In [Vec14], the harmonic average option is priced using a martingale approach; In [LL09] double barrier Parisian options are priced using probability theory;In [PP09], Rainbow Asian options are priced using stochastic calculus; etc.
They are without argument both powerful, and their reach is not limited to trivial problems. We however feel that this power comes at the cost of intuition, clarity or even relevance. What we mean by trading power for relevance is that formula may be derived that will hardly ever be implemented because of their penalizing complexity. One can see in Fig.1.3, *one* of the coefficients required to compute the price of an American option, derived using stochastic calculus in [Zhu06]; while the final formula itself involves an infinite sum of those terms. This



$$\bar{U}_n(x,\tau)=\frac{1}{\sqrt{\pi}}\left\{e^{(\gamma-1)x/2}\int_{-x/2\sqrt{\tau}}^{x/2\sqrt{\tau}}\psi_n(2\sqrt{\tau}\xi+x)\,e^{(\gamma-1)\sqrt{\tau}\xi-\xi^2}\,d\xi+\int_{x/2\sqrt{\tau}}^{\infty}\left[e^{(\gamma-1)x/2}\psi_n(2\sqrt{\tau}\xi+x)+e^{-(\gamma-1)x/2}\psi_n(2\sqrt{\tau}\xi-x)\right]e^{(\gamma-1)\sqrt{\tau}\xi-\xi^2}\,d\xi\right\}$$
$$-(\gamma+1)\sqrt{\tau}e^{-(\gamma-1)x/2+(\gamma+1)^2\tau/4}\int_{x/2\sqrt{\tau}}^{\infty}\psi_n(2\sqrt{\tau}\xi-x)e^{2\gamma\sqrt{\tau}\xi}\,\mathrm{erfc}\left(\xi+\frac{(\gamma+1)}{2}\sqrt{\tau}\right)d\xi\;-\frac{2}{\sqrt{\pi}}e^{(\gamma+1)^2\tau/4}\int_0^{\infty}e^{-(\gamma+1)\eta/2}$$
$$\times\int_{(x+\eta)/2\sqrt{\tau}}^{\infty}\phi_n\left(\tau-\frac{(x+\eta)^2}{4\xi^2}\right)e^{-[(\gamma+1)(x+\eta)/4\xi]^2-\xi^2}\,d\xi\,d\eta\;+\int_0^{\tau}\left\{\frac{e^{(\gamma+1)^2\eta/4}}{\sqrt{\pi}}\left[e^{(\gamma-1)x/2}\quad\ldots\ldots\right.\right.$$

Figure 1.3: Extract of the American option fair price formula

example illustrates how current approaches sometimes disregard practical considerations that should, at the contrary, be a major concern. That is not to say that all complex problems (and pricing an American option **is** a complex problem) have a simple formula waiting to be found, yet, how good is a formula that can not be realistically implemented ? We, on the other hand, embrace a multi-disciplinary approach that aims to go beyond tightly connected fields of mathematics. Trading some of the rigor of the mainstream approaches for a boost in relevance and clarity, we will tackle the option pricing problem using insights available in physics, and applies it when it makes sense to our finance problem. Because problem of physics have been tested and tested over and over again, the intractable formula have been widely pruned out. Moreover, the superior intuition that comes by reframing an abstract probabilistic problem into a physic's one could hardly be argued.

1.4 The Econophysics approach

We will tackle this problem from the theoretical ground up to its efficient implementation using the interdisciplinary econophysics approach. The econophysics field can be roughly dated back to 1995, when Eugene H. Stanley coined the term. It describes the field where physicists and physics-minded researcher aimed to solve problem from economics using tools and concepts mostly found in physics field. Figuring prominently in econophysics publications are results derived using statistical physics. For example the explanation of the so called fat-tail characteristics in index distribution [MS95], or the modeling of ripples and contamination in market crashes as aftershocks in seismical events[Sor09].

Recently another trend has emerged in econophysics, called rather unofficially "Quantum finance"[Baa04]. As the name puts it, quantum finance is the attempt to solve or explain problem from finance using theory and tools from quantum physics. It ranges from ambitious quantum models of the financial market, where Schrodinger equation replaced the Black-Scholes equation, in order to account for uncertainty and inefficiency[Hav02],[CPVR10], to the more practical matter of option pricing using quantum mechanics as popularized by Baaquie[Baa04][Bel97][BKS00], Linetsky[Lin98] and Dash[J.D88][J.D89]. Baaquie used the connection between the Schrodinger equation and the diffusion equation that permeates option pricing theory, in order to work out an Hamiltonian for the problem under study, then use well developed techniques from quantum mechanics to solve the problem. The path integral formulation of quantum mechanics is used by Dash, Linetsky and Baaquie to solve problems where functional integration of the payoff can be better handled.

We should point out that even in this interdisciplinary field, it is usual that implementation aspects are left out altogether. Unlike computational physics, econophysics is highly focused on big-data analysis, statistical modeling, behavioral explanation of economics phenomena, and rarely discuss (if at all) how to efficiently implement the models or formula derived. We wish to avoid that pitfall and keep an eye on efficient implementation. One could make the reasonable argument that we push toward computational econophysics in our philosophy.

1.5 Thesis goal

It is usually the case that theoretical publications do not deal with the implementation of the proposed formula, and alternatively, implementation studies often limit their scope to the straightforward vanilla derivatives as a benchmark tool. The contributions of this thesis lie in both the theoretical and implementation space, and pursue this endeavour in an interdisciplinary fashion.

In the theoretical part, we will demonstrate how particularly suitable quantum mechanics path integrals are to study path dependent options. By taking as a scope of study complex financial derivatives and deriving formula that lead to an easy implementation we will prove that econophysics is a valid and pragmatically sound framework to work out complex financial problems. Our first contribution is an improvement and extension on an existing exotic option pricing formula using path integrals. Our second theoretical contribution is the design of a new type of exotic option that we price using path integrals, demonstrating their power and flexibility.

Then the pricing problem in its computational and implementation complexity is studied for the following reason. When pricing formula are newly derived in a finance publication and implementation is discussed, it is usually only to state the time it takes to get the results on a vanilla CPU implementation. There is very limited consideration to improving the computational speed for their original results. We felt that it is not good enough and that optimization should be discussed when formula are not straightforward to code. To reach that goal we decided to use a GPU card as an acceleration device easily affordable for test purpose and will aim at proposing new insights in the implementation of pricing formula. We will have two original contributions to present: first a case study of heterogeneous CPU/GPU designs is done when price is available in a series form. Then, for the studies where new formula are introduced and must be benchmarked, we propose a novel GPU algorithm to speed up Monte Carlo simulations.

1.6 Thesis overview

In the second chapter, the required background from probability theory is introduced. Stochastic processes are introduced as a suitable tool to describe randomness in financial assets dynamics. The Wiener process is described along with the stochastic integration in the Ito interpretation. The defining properties of Ito calculus that will come handy later on are also showcased.

In the third chapter, an introduction to financial markets is given that concentrates on exotic derivatives. The basic working of such a market is explained, then we concentrate on describing the features of an exotic option contract. The Black-Scholes model of a market is introduced along with its philosophy, our work in this thesis will be done inside this model entirely. Since our theoretical contributions both use path integrals as a major technical tool to price exotic options, the fourth chapter is devoted to presenting path integrals. From the Wiener path integral to the Feynman path integral, the potential use for option pricing are recalled. The fifth and sixth chapters represent our two original theoretical contributions to econophysics. We build on and improve an existing work targeting single outside barrier Asian option, extending it to the double barriers case. We also propose an original option, the Wasabi option and study its pricing problem. From the seventh chapter starts the implementation part of this thesis. General Purpose Graphic Processing Units (GPGPU) as a computation framework for distributed computation is introduced. The relevant terminology and concepts are presented along with some code snapshots in order to be as practical as required The eighth chapter is a case study we conducted on the various heterogeneous CPU/GPU designs available for a distributed computation when the target is a basket option and price is available as an series. Empirical evidence are put forward for an heterogeneous implementation with distribution along the data axis. We conclude our original contributions by introducing in the ninth chapter a method to speed up the Monte Carlo simulation that are frequently used for benchmarking the accuracy of a novel pricing formula. This novel algorithm that we will call "Shuffled Monte Carlo" will be benchmarked on vanilla products first in order to deliver empirical evidence that our method is sound. Then it will be used on more involved products such as the Wasabi option where a Monte Carlo schems is actually justified. Our method using the GPU as a computation target, will exhibit up to halving of computation time.

The conclusion will summarizes the relevance of our contributions to the field of econophysics and more generally to computational finance.

Part I

Theory

Chapter 2

Stochastic Calculus

The proper way to describe the dynamics of a financial asset with randomness built into it is using stochastic processes. To be able to handle those concepts first ask us to take a detour into measure and probability theory, before moving on to the stochastic processes themselves. So let us start our journey with some definitions pertaining to probability theory. This chapter is intended to give a rather elementary introduction to stochastic calculus, therefore some details that we think the readers are already familiar with are left out, but can be easily found for example in [Shr04]

2.1 Probability space

The first step we will take in building a proper understanding of stochastic processes and stochastic calculus involves first defining the space in which those objects live. To that effect we must define a probability space, and introduce the necessary and connected concepts. We do not claim to give a full overview of measure theory but only the subset directly relevant to our study, readers that want to see the introduced concepts further elaborated are redirected to [Coh13] for example.

Definition 2.1.1: Sigma Algebra

Let Ω be a set, a σ-algebra $\mathcal{F}$ on Ω is the family of subsets of Ω defined by the following properties:

1. $\emptyset \in \mathcal{F}$
2. $\mathcal{A} \in \mathcal{F} \Rightarrow \mathcal{A}^C \in \mathcal{F}$ where $\mathcal{A}^C := \Omega \setminus \mathcal{A}$
3. $\mathcal{A}_i \in \mathcal{F} \Rightarrow \bigcup_{i \in \mathbb{N}}^{\infty} \mathcal{A}_i \in \mathcal{F}$

Let $\mathcal{A}$ be a set and $\mathcal{F}$ a σ-algebra on Ω, $\mathcal{A}$ is called a measurable set if $\mathcal{A} \in \mathcal{F}$.

The second property states that a σ-algebra is closed under complementation. The last property states that a σ-algebra is closed under **countable** unions. By using property 2 and 3, it is direct that a σ-algebra is also closed under countable intersections.

As an example of a sigma algebra that we will deal with numerous times: the Borel σ-algebra

$\mathcal{B}$ is the smallest σ-algebra containing all the open sets in $\mathbb{R}$ formed by countable union and intersection of open sets in $\mathbb{R}$, a set $A \in \mathcal{B}$ is called a Borel set.

The concept of a sigma algebra is important in that we must restrict ourselves when dealing with the problem of measuring events in some given space. For space with countably many events (e.g. rolling a dice) it is easy enough to assign a single value to each such that the sum is one, but we are faced with challenges when dealing with uncountably many events (e.g. choosing a real number between 0 and 1). Thus it is of critical importance to be conservative and careful when working in an uncountable space in which we aim to properly assign probability to events, and maybe work with a family smaller than a σ-algebra, a π-system that generate the σ-algebra we are interested into. Readers who want to dig deeper into those technicalities are redirected to the discussion preceding the Caratheodory extension theorem in any measure theory textbook.

Let us now define a measure

Definition 2.1.2: Measure

A measure μ is an extended real-valued function defined on the measurable space $(\Omega, \mathcal{F})$ such that

1. $\mu(\emptyset) = 0$
2. $\mu(A) \geq 0, \forall A \in \mathcal{F}$
3. for $\{A_n\}_{n=1}^{\infty}$ a countable, disjoint sequences of A_n in $\mathcal{F}$, we have:
   $\mu\left(\bigcup_{n=1}^{\infty} A_n\right) = \sum\limits_{n=1}^{\infty} \left(\mu(A_n)\right)$

If $\mu(\Omega) = 1$ then the measure is called a probability measure, often written $\mathbb{P}$, and $\mathbb{P}(A)$ is then the probability of event A for $A \in \mathcal{F}$

We will need later on to be able to transform measures, and in order to do so we need to introduce some conditions under which measures are closed under integration with respect to a measurable function. The content of the Radon Nikodym theorem states the condition under which it is feasible.

Theorem 2.1.1: Radon Nikodym theorem

Let μ and ν be two measures on the same measurable space, μ is continuous with respect to ν if

$$\nu(A) = 0 \Rightarrow \mu(A) = 0 \tag{2.1}$$

and we write $\mu \ll \nu$.

If those measures are σ-finite then there is an a.e unique integrable function h such that

$$\mu(A) = \int_A h(s)\, \nu(ds) \tag{2.2}$$

h is called **the Radon Nikodym derivative** of μ with respect to ν and is written $d\mu/d\nu$

Now building on all the concepts introduced so far we can properly define a probability space.

Definition 2.1.3: Probability space

A probability space is a triplet $(\Omega, \mathcal{F}, \mathbb{P})$ with Ω a set, $\mathcal{F}$ a σ-algebra on Ω and $\mathbb{P}$ a probability measure on $\mathcal{F}$

As an example in a non discrete space, let us take the experience of picking a real number at random between 0 and 1, uniformly. Then $\Omega = [0, 1]$, take $\mathcal{F}$ to be the σ-algebra generated by borel sets in Ω and $\mathbb{P}$ defined as $\mathbb{P}([a, b]) = b - a$. The measure $\mathbb{P}$ just described is often called *Lebesgue measure*, and written $\mathcal{L}$.

Definition 2.1.4: Measurable function

With $(\Omega, \mathcal{F})$ a measurable space, a real-valued function $X : \Omega \to \mathbb{R}$ is said to be measurable with respect to $\mathcal{F}$ if $\{s \in \Omega : X(s) \leq a\} \in \mathcal{F}$,all $a \in \mathbb{R}$. If the space in question is a probability space then the measurable question is called a random variable.

Now that we have properly defined what is a random variable and in what kind of space it lives, we can move on and start building stochastic processes.

2.2 Stochastic process in continuous time

In few words, a stochastic process is merely a dynamic process with randomness built into it. However on a more precise levels, the mathematical definition though quite concise, entails many technicalities. Let's try now to define a stochastic process precisely enough

Definition 2.2.1: Stochastic process

Let $(\Omega, \mathcal{F}, \mathbb{P})$ be a probability space , and I a real interval of the form $[0, T]$ or $\mathbb{R}^+$. A measurable stochastic process on $\mathbb{R}$ is a collection $(X_t)_{t \in I}$ of $\mathbb{R}-$valued random variables such that the map

$$X : I \times \Omega \longrightarrow \mathbb{R}, X(t, \omega) := X_t(\omega)$$

is measurable with respect to the product σ-algebra $\mathcal{B}(I) \otimes \mathbb{R}$

To fix intuition it is rather useful to think of stochastic processes as paths or trajectories, drawn from a probability space (an example can be seen on Fig.2.1). Hence keeping t fixed we have a random variable

$$\omega \longrightarrow X_t\left(\omega\right); \omega \in \Omega$$

Also, with ω fixed, we have a function

$$t \longrightarrow X_t\left(\omega\right), t \in I$$

describing the trajectory of our stochastic process. We will often omit the ω argument of $X_t\left(\omega\right)$.

We are going now to introduce one of the most fundamental stochastic process, and a building block used to describe uncertainty in stock market as well as other noisy processes.

Definition 2.2.2: Brownian motion

A standard Brownian motion (or Wiener process) $W\left(t\right)$ on the probability space $\left(\Omega, \mathcal{F}, \mathbb{P}\right)$ verifies

- $W\left(t\right)$ has independent and stationary increments, i.e. for $0 \leq t \leq T$, $W_T - W_t$ has normal distribution $\mathcal{N}_{0,T-t}$ and the random variables

  $$W_{t_1} - W_{t_0}, ..., W_{t_n} - W_{t_{n-1}}$$

  are independents for any $0 \leq t_0 < ... < t_n$
- The joint distribution is given by

  $$\mathbb{P}\left(\left(W_{t_1}, ..., W_{t_n}\right) \in B_1 \times ... \times B_n\right) = \int\limits_{B_1} ... \int\limits_{B_n} \phi\left(x_1, t_1\right)\phi\left(x_2 - x_1, t_2 - t_1\right)$$
  $$...\phi\left(x_n - x_{n-1}, t_n - t_{n-1}\right)dx_1 dx_2 ... dx_n$$

  where $\phi\left(x,t\right) := \left(2\pi t\right)^{\frac{-1}{2}} e^{\frac{-|x|^2}{2t}}$ is the normal distribution density function.

Another useful characterization is that it is a Gaussian process with null mean and covariance function $\min\left(t,s\right)$,all its finite-dimensional distributions are multivariate normal.

Now, the most crucial property that Brownian motion exhibits is connected to quadratic variation, so let us first define clearly what is quadratic variation

Definition 2.2.3: Quadratic variation

Let $f(x)$ a function of a real variable, its quadratic variation over $[0,t]$ is the following limit when it exists

$$[f,f]_t = \lim_{\delta_n \to 0, n \to \infty} \sum_{i=1}^{n} \left(f\left(t_i^n\right) - f\left(t_{i-1}^n\right)\right)^2 \tag{2.3}$$

with $0 = t_0^n < ... < t_n^n = t$ and $\delta_n = \max_{1 \leq i \leq n}\left(t_i^n - t_{i-1}^n\right)$

It is known from any calculus textbook that continuous functions have zero quadratic variation, that is not true for the Brownian motion. Now we will prove the following proposition of utmost importance

Proposition 2.2.1: Brownian motion quadratic variation

Brownian motion has a quadratic variation $[W,W]_t = t$ a.s

Proof. Let $\Delta W_{t_i} = W_{t_{i+1}} - W_{t_i}$ and $t_i \equiv t_i^n := i.2^{-n}t$ for $0 \leq i \leq 2^n$

$$\begin{aligned}
\mathbb{E}\left[\left([W,W]_t - t\right)^2\right] &= \mathbb{E}\left[\left(\sum_{i=0}^{2^n-1}\left(\Delta W_{t_i}\right)^2 - \sum_{i=0}^{2^n-1} t_i\right)^2\right] = \mathbb{E}\left[\left(\sum_{i=0}^{2^n-1}\left(\Delta W_{t_i}\right)^2 - t_i\right)^2\right] \\
&= \sum_{i=0}^{2^n-1}\mathbb{E}\left[\left(\left(\Delta W_{t_i}\right)^2 - t_i\right)^2\right] = \sum_{i=0}^{2^n-1}\mathbb{E}\left[\left(\Delta W_{t_i}\right)^4 - 2t^2 2^{-2n} + 2^{-2n}t^2\right] \\
&= \sum_{i=0}^{2^n-1} 2t^2 2^{-2n} = 2t^2 2^{-n} \to 0 \text{ as } n \to \infty
\end{aligned}$$

therefore $[W,W]_t \to t$ in $L^2(\mathbb{P})$ □

The Wiener process will be the only stochastic process we will be dealing with later on when working with stochastic representation of the financial market. There are other stochastic processes available if one is motivated to build a different market representation, e.g. the Gamma process where increments are not normally but Gamma distributed, or the Poisson process used in queuing theory . Yet, since most of the tools that we will use later on are connected to the normality of the Wiener process, we will not elaborate on the other type of processes.

An important property of the Wiener process is its martingale property, simply put, expectation of its future whereabout according to where it is now is merely the current position. If we try for a moment to set aside mathematical rigor, and reframe our discussion in financial terms: the expectation of a future gain knowing our current position, if the game is not rigged, should not be better or worse than our current position. It clarifies what is expected

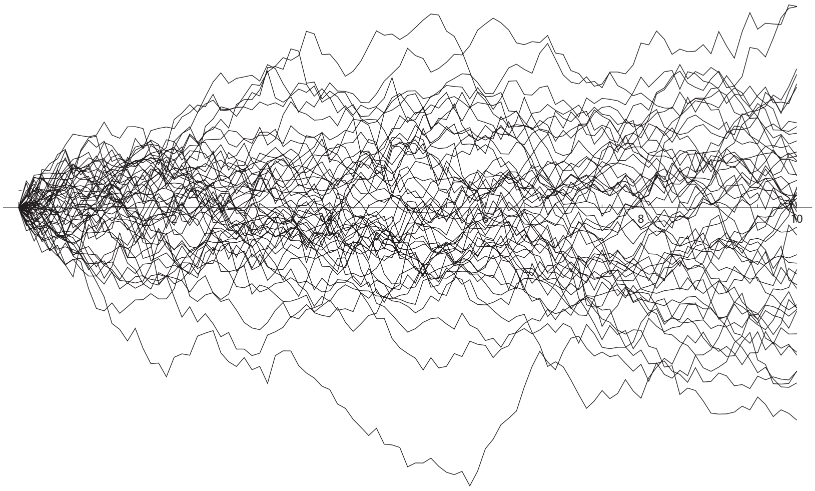


Figure 2.1: Example of 50 Brownian trajectories generated by computer

of a fair game. To define it mathematically and precisely we first need to introduce the concept of a filtration to make sense of the martingale definition

Definition 2.2.4: Martingale

A filtration $\{\mathcal{F}_t\}_{t\geq 0}$ on the space $(\Omega, \mathcal{F})$ is a family of σ-algebras $\mathcal{F}_t \subseteq \mathcal{F}$ such that

$$0 \leq t < T \Rightarrow \mathcal{F}_t \subset \mathcal{F}_T$$

A stochastic process X_t on $(\Omega, \mathcal{F}, \mathbb{P})$ is a martingale w.r.t a filtration $\{\mathcal{F}_t\}_{t\geq 0}$ and a measure $\mathbb{P}$ if the following holds

1. X_t is $\mathcal{F}_t$-measurable for all t
2. $\mathbb{E}\left[|X_T|\right] < \infty$
3. $\mathbb{E}\left[X_T|\mathcal{F}_t\right] = X_t$ for all $T \geq t$

The third property is obviously where we drew our analogy with a fair financial market. The filtration that will be used is the one generated by the Brownian motion itself: $\mathcal{F}_t = \sigma\{W_s; s \leq t\}$ contains the history of the process up until time t.

Let us prove right now that the Wiener process has the martingale property, as it helps fixing concept introduced up until now. Let $0 \leq s \leq t$

Proof.

$$\mathbb{E}\left[W_t|\mathcal{F}s\right] = \mathbb{E}\left[W_t + W_s - W_s|\mathcal{F}s\right] = \mathbb{E}\left[W_t - W_s|\mathcal{F}s\right] + \mathbb{E}\left[W_s|\mathcal{F}s\right] = 0 + W_s = W_s \quad (2.4)$$

□

where the second to last equality uses the measurability of W_s, the independence of $W_t - W_s$ and $\mathcal{F}_s$, and finally the normality of increments. The Brownian motion has a lot more of different and interesting properties but since they will barely pertain to us, we will not elaborate more and instead redirect interested readers to [KS91].

Another rather theoretical point should be explained here, for it will prove to be relevant in option pricing. It may happen that some calculations are cumbersome with respect to the measure that is currently under study, therefore it may make sense for some case to change the measure, then do a simpler equivalent computation. Additionally, we may be in a situation where for theoretical reason we would like to remove a drift component from a Wiener process. This leads us to the following version of the Girsanov theorem

Theorem 2.2.1: Girsanov's theorem

Let Z_t^λ be the exponential martingale associated to the L^2 process λ by

$$Z_t^\lambda = \varepsilon\left(\int_0^t \lambda dW_s\right) := \exp\left[-\int_0^t \lambda_s dW_s - \frac{1}{2}\int_0^t |\lambda_s|^2 ds\right] \tag{2.5}$$

and consider the measure $\widetilde{\mathbb{P}}$ defined by

$$\left.\frac{d\widetilde{\mathbb{P}}}{d\mathbb{P}}\right|_{\mathcal{F}_t} = Z_t^\lambda \tag{2.6}$$

we have the result that the process

$$\widetilde{W}_t = W_t + \int_0^t \lambda_s ds \tag{2.7}$$

is a $\widetilde{\mathbb{P}}$-Brownian motion.

2.3 Ito calculus

Starting with the motivation to introduce models of financial market that takes into account the randomness of price behaviour, we introduced stochastic processes. Now we need some rules to do anything with said stochastic processes. The calculus for stochastic processes is obviously different from the regular calculus involving only deterministic functions. That being said we still have to demonstrate why it differs and lay out the some fundamental rules. This section and the next one are devoted to such a task.

If we have some deterministic functions $f(X)$ and $X(t)$ and f has a derivative then $\frac{\partial f}{\partial t} = \frac{\partial f}{\partial X}\frac{\partial X}{\partial t}$ by regular calculus chain rule. However, if $X(t)$ were to be a stochastic

process of the Ito type (we will make clear later on what is meant by a Ito process)then the usual chain rule does not apply any more.
Kiyoshi Ito (清 伊藤,1915-2008) was the first one to figure a way to extend the methods of calculus to stochastic processes by deriving a chain rule formula that applied to stochastic processes[Ito44]. We now define what is a Ito process and then actually proves that it is well defined, before laying out the formula for chain rule in stochastic settings

Definition 2.3.1: Ito process

Let W_t be a Brownian motion on $(\Omega, \mathcal{F}, \mathbb{P})$, an Ito process $X(t) \equiv X_t$ on the same space is a stochastic process of the form

$$X_t = X_0 + \int_0^t \mu(s,\omega)\,ds + \int_0^t \sigma(s,\omega)\,dW_s \tag{2.8}$$

with the following integrability conditions

$$\mathbb{P}\left[\int_0^t \sigma(s,\omega)^2\,ds < \infty, \forall t \geq 0\right] = 1 \tag{2.9}$$

$$\mathbb{P}\left[\int_0^t |\mu(s,\omega)|ds < \infty, \forall t \geq 0\right] = 1 \tag{2.10}$$

and the stochastic integral $\int_0^t \sigma(s,\omega)\,dW_s$ is taken in the Ito sense.

For brevity sake, the differential form of the Ito process is often favored keeping in mind that only the integral form in (2.8) is well defined

$$dX_t = \mu(t,\omega)\,dt + \sigma(t,\omega)\,dW_t$$

It remains to make sense of what is actually an Ito integral, and its construction will lead us to a deeper understanding of what is really peculiar (among other things) to stochastic calculus.

2.3.1 Constructing the Ito integral

We want to define properly the following Ito integral

$$I(t) := \int_0^t \sigma(s,\omega)\,dW_s \tag{2.11}$$

The first step is to start proving its existence for a simple enough class of integrands, then show that a larger class of function can be approximated by the simpler ones. The exposition follows [Øks03] and [All07].

Assume that a square integrable simple process $\sigma(s,\omega)$ posses the following elementary representation

$$\sigma(t,\omega) = \sum_j e_j(\omega)\, \mathbb{1}_{[t_j,t_{j+1})}(t) \tag{2.12}$$

with $t_j \equiv t_j^n := j.2^{-n}t$ for $0 \leq j \leq 2^n$ and $e_j(\omega)$ are $\mathcal{F}_{t_j}$-measurable random variables (except for e_0 that is constant since it is the same for every paths). The Ito integral for elementary function of the form described by (2.12) is defined as

$$\int\limits_0^t \sigma(s,\omega)\, dW_s = \sum_{j=0}^{2^n-1} e_j(\omega)\left[W_{t_{j+1}} - W_{t_j}\right](\omega) \tag{2.13}$$

We would like now to extend the integral definition to a more interesting class of integrands by approximating them with the elementary functions we just described. Since at one point we will have to do a limiting operation and check for convergence, we need to have some bound on the integral. Hence to continue any further we need the following theorem

Theorem 2.3.1: Ito isometry for simple functions

The Ito integral as defined in (2.13) obeys the following isometry

$$\mathbb{E}\left[\left(\int\limits_0^t \sigma(s,\omega)\, dW_s\right)^2\right] = \mathbb{E}\left[\int\limits_0^t \sigma^2(s,\omega)\, ds\right] \tag{2.14}$$

Proof. Writing again ΔW_j for $W_{t_{j+1}} - W_{t_j}$, and omitting the ω argument

$$\mathbb{E}\left[\left(\int\limits_0^t \sigma(s)\, dW_s\right)^2\right] = \mathbb{E}\left[\left(\sum_{j=0}^{2^n-1} e_j \Delta W_j\right)^2\right] = \mathbb{E}\left[\sum_j e_j^2 \Delta W_j^2\right] + 2\mathbb{E}\left[\sum_{i\neq j} e_i e_j \Delta W_j \Delta W_j\right] \tag{2.15}$$

We look first at the cross-terms. Taking $i < j$ then $e_i e_j \Delta W_i$ is $\mathcal{F}_{t_i}$-measurable, while ΔW_j is independent of $\mathcal{F}_{t_i}$, thus $\mathbb{E}\left[e_i e_j \Delta W_j \Delta W_j\right] = \mathbb{E}\left[e_i e_j \Delta W_j\right]\mathbb{E}\left[\Delta W_j\right] = 0$, since increments have null expectations.

Now in the first sum, e_j^2 is $\mathcal{F}_{t_j}$-measurable and the increment ΔW_j is independent of $\mathcal{F}_{t_j}$, thus we also have $\mathbb{E}\left[e_j^2 \Delta W_j^2\right] = \mathbb{E}\left[e_j^2\right]\mathbb{E}\left[\Delta W_j^2\right] = \mathbb{E}\left[e_j^2\right](t_{j+1} - t_j)$.

Plugging in those results yield

$$\mathbb{E}\left[\sum_j e_j^2 \Delta W_j^2\right] + 2\mathbb{E}\left[\sum_{i\neq j} e_i e_j \Delta W_j \Delta W_j\right] = \sum_j \mathbb{E}\left[e_j^2\right](t_{j+1}-t_j) = \mathbb{E}\left[\int_0^t \sigma^2(s,\omega)\,ds\right] \tag{2.16}$$

□

Suppose we want to extend the Ito integral to a continuous square integrable function $\sigma(t,\omega)$, the way to proceed is to build simple processes $\sigma_n(t,\omega)$ approximating $\sigma(t,\omega)$ (see for illustration Fig.2.2, demonstrate that their Ito integral form a Cauchy sequence in $L^2(\mathbb{P})$, then since this space is complete [Coh13] the Ito integral of $\sigma(t,\omega)$ will be defined as the existing limit.

Let's start by defining the simple process

$$\sigma_n(t,\omega) := \sum_{j=0}^{2^n-1} \sigma(t_j,\omega)\,\mathbb{1}_{[t_j,t_{j+1})}(t) \tag{2.17}$$

with its Ito integral defined by (2.13), then

$$\begin{aligned}
&\| \int_0^t \sigma_n(s,\omega)\,dW_s - \int_0^t \sigma_m(s,\omega)\,dW_s \|_{L^2} = \mathbb{E}\left[\left(\int_0^t \sigma_n(s,\omega)\,dW_s - \int_0^t \sigma_m(s,\omega)\,dW_s\right)^2\right]\\
&= \mathbb{E}\left[\left(\int_0^t (\sigma_n(s,\omega)-\sigma_m(s,\omega))\,dW_s\right)^2\right] = \mathbb{E}\left[\int_0^t ((\sigma_n(s,\omega)-\sigma_m(s,\omega)))^2\,ds\right]\\
&\leq 2\mathbb{E}\left[\int_0^t ((\sigma_n(s,\omega)-\sigma(s,\omega)))^2\,ds\right] + 2\mathbb{E}\left[\int_0^t ((\sigma_m(s,\omega)-\sigma(s,\omega)))^2\,ds\right]\\
&\to 0 \text{ as } n,m\to\infty \text{ since } \sigma(t,\omega) \text{ is continuous for all } \omega
\end{aligned} \tag{2.18}$$

The Ito integral $\int_0^t \sigma(s,\omega)\,dW_s$ is taken to be the limit in $L^2(\mathbb{P})$ of this Cauchy sequence.

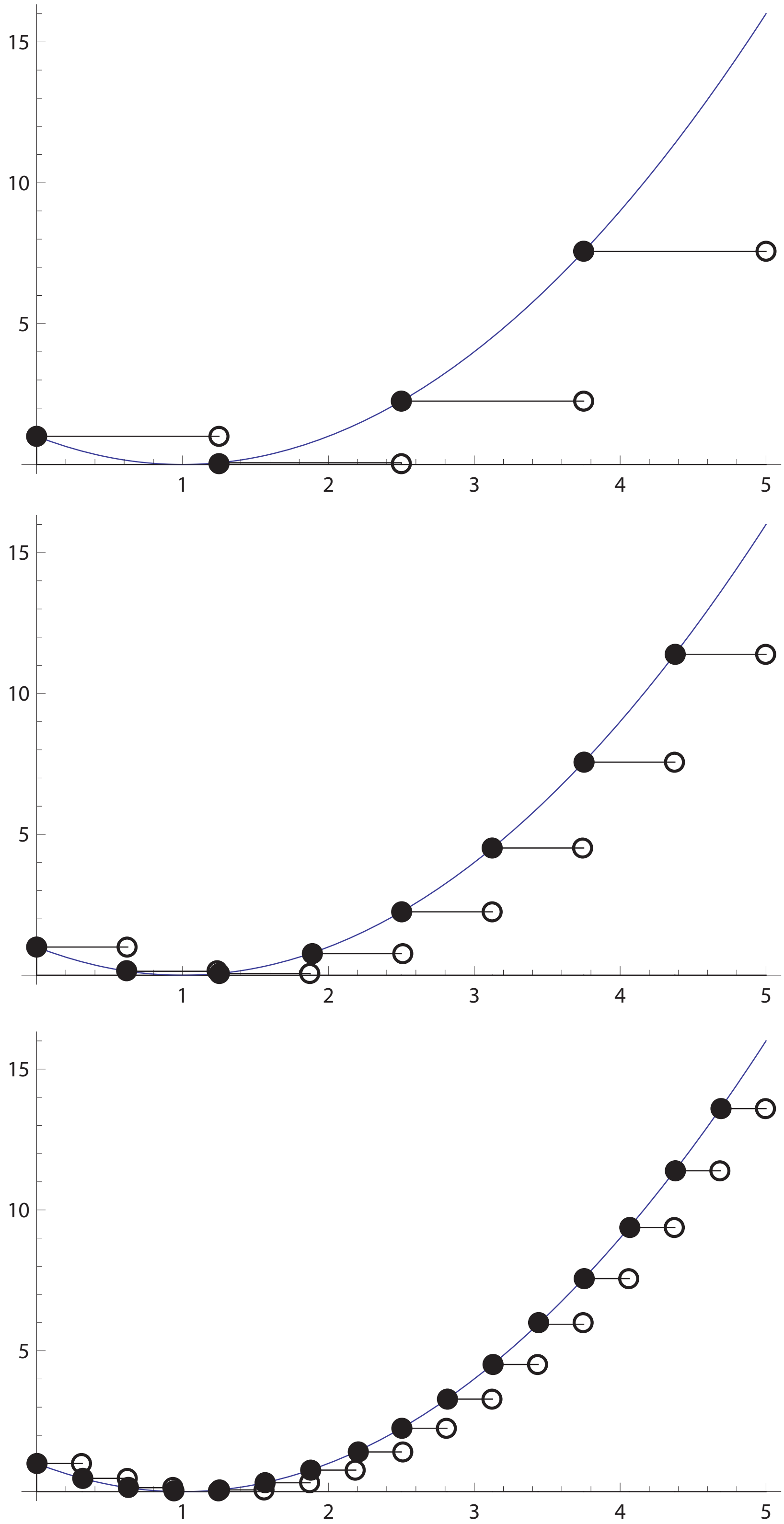


Figure 2.2: Approximation of an arbitrary function (blue line) by piecewise constant simple function (black line and circles), where the partition gets more refined from top to bottom.

Definition 2.3.2: Ito integral for general integrand

Let $\sigma(t,\omega)$ be adapted to $\{\mathcal{F}_t\}$, $\mathcal{B}\times\mathcal{F}$-measurable and square integrable, then its Ito integral is defined as

$$I(t) := \int_0^t \sigma(s,\omega)\,dW_s = \lim_{n\to\infty}\int_0^t \sigma_n(s,\omega)\,dW_s \tag{2.19}$$

with $\{\sigma_n\}$ a sequence of elementary functions such that

$$\mathbb{E}\left[\int_0^t ((\sigma_n(s,\omega)-\sigma(s,\omega)))^2\,ds\right] \to 0 \text{ in } L^2(\mathbb{P}) \text{ as } n\to\infty \tag{2.20}$$

It satisfies the following property

1. The paths of $I(t)$ are continuous w.r.t its upper limit
2. For each t, $I(t)$ is $\mathcal{F}_t$-measurable
3. $I(t)$ is a martingale, it has null expectation
4. **(Ito isometry)** $\mathbb{E}\left[I^2(t)\right] = \mathbb{E}\left[\int_0^t \sigma^2(s,\omega)\,ds\right]$

Proof for any of those properties and others can be found in [Shr04] for example.

2.3.2 Ito lemma

Now that we have a proper definition for an Ito process, we would like to do calculus on it. As we said earlier, Ito calculus obeys different rules than regular calculus. To start with we would like to see for a function $f(X)$ with X_t an Ito process, what happens when $X_t \to X_{t+\Delta t}$. For a differentiable deterministic function, sure enough $df(X) = f'(X)dX$, however as we saw earlier Brownian motion has non-zero quadratic variation and this will matter here.

Lemma 2.3.1: Ito lemma

Let X_t be an Ito process of the form

$$dX_t = \mu\left(t,\omega\right)dt + \sigma\left(t,\omega\right)dW_t \tag{2.21}$$

If $f\left(t,x\right)$ is $C^2\left(\left[0,\infty\right)\times\mathbb{R}\right)$, then $Y_t := f\left(t,X_t\right)$ is an Ito process, and

$$dY_t = \frac{\partial f}{\partial t}\left(t,X_t\right)dt + \frac{\partial f}{\partial x}\left(t,X_t\right)dX_t + \frac{1}{2}\frac{\partial^2 f}{\partial x^2}\left(t,X_t\right)dX_t^2 \tag{2.22}$$

in differential form where the following rules apply

$$dt.dt = dW_t.dt = 0, dW_t.dW_t = dt \tag{2.23}$$

In integral form and with all rules applied, Ito lemma yields

$$\begin{aligned} f\left(X_t,t\right) = f\left(X_0,0\right) + &\int_0^t \left(\frac{\partial f}{\partial s}\left(s,X_s\right) + \mu\frac{\partial f}{\partial x}\left(s,X_s\right) + \frac{\sigma^2}{2}\frac{\partial^2 f}{\partial x^2}\left(s,X_s\right)\right)ds \\ &+ \int_0^t \mu\frac{\partial f}{\partial x}\left(s,X_s\right)dW_s \end{aligned} \tag{2.24}$$

Remember that only the integral form is well defined, the differential is preferred for shortness of exposition.

This lemma is essential to any of the relevant computation involving Ito processes, and as an example let us start with the following stochastic differential equation (SDE)

$$dS_t = S_t\left(\mu dt + \sigma dW_t\right)\left(\equiv \frac{dS_t}{S_t} = \mu dt + \sigma dW_t\right) \tag{2.25}$$

that models the return of an asset S_t, where μ is a constant rate of return, and σ is a constant volatility parameter describing the random shocks. Since it does bear resemblance to a logarithmic derivative, we apply the Ito lemma to $f(s,t) = \log\left(s\right)$

$$df = \left(\frac{\mu s}{s} - \frac{\sigma^2 s^2}{2s^2}\right)dt + \frac{\sigma s}{s}dW_t = \left(\mu - \frac{\sigma^2}{2}\right)dt + \sigma dW_t \tag{2.26}$$

therefore

$$\log\left(S_t\right) = \log\left(S_0\right) + \left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t \Rightarrow S_t = S_0\exp\left(\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right) \tag{2.27}$$

Equation (2.27) describes a geometric Brownian motion, used in the Black-Scholes model to describe a risky asset.

This example, apart from demonstrating the usefulness of Ito lemma, was also the first SDE encountered so far.

Definition 2.3.3: Stochastic differential equation

An equation of the form

$$dX_t = \mu\left(X_t, t\right) dt + \sigma\left(X_t, t\right) dW_t \tag{2.28}$$

where $\mu\left(X_t, t\right)$ and $\sigma\left(X_t, t\right)$ are given, and X_t is unknown is called a stochastic differential equation (SDE).

Conditions for existence and uniqueness are essentially of growth type on the coefficients and can be found in [Øks03]. An important property of solution to SDE is that they are Markov processes[Shr04], and their conditional expectation have a martingale representation. More precisely we have

Theorem 2.3.2: Markov property for solution of SDE

Let $X\left(u\right)$ be a solution to (2.28) with initial condition given at $t = 0$. Denote by

$$g\left(t, x\right) = \mathbb{E}^{t,x} h\left(X_T\right) \tag{2.29}$$

the expectation (for initial condition $X_t = x$) of a Borel measurable function h. Then for $0 \leq t \leq T$

$$\mathbb{E}\left[h\left(X_T\right) \middle| \mathcal{F}_t\right] = g\left(t, X_t\right) \tag{2.30}$$

Furthermore the stochastic process $g\left(t, X_t\right)$ is a martingale.

Chapter 3

Financial market and derivative

In this chapter our aim is to introduce the core of the financial theory relevant to this thesis. As such it will be mainly concentrated on the derivative market and even more especially on the challenge of pricing exotic options. Though to reach that point we have to make some detours and introduce some more general points relative to the market in general.
The chapter will be structured as such. we will describe a financial market first in an abstract then in a more mathematical fashion, this will lead us to the presentation of the derivative contracts. We will then continue this exposition by the presentation of the Black-Scholes model which will be the setting in which our theoretical work takes place. Finally we will present what is meant by the challenge of option pricing.

3.1 Financial markets

A financial market is where a buyer and a seller meets to discuss terms of goods (or services) they want to trade[Hul08]. It can be commodities, stocks, currency, options, etc. The major segmentation is between two main type of markets

1. Capital markets
    - Equity market
    - Debt market
    - Derivative market
    - Indexes
2. Money markets
    - Cash time deposits
    - Treasury bills
    - Deposit certificate
    - Eurodollars deposits

This study is focused on the buying and selling of derivatives, which can be seen to belong to the capital market. A derivative is defined as a financial instrument whose value depends on the value of another. The asset on which the value of the derivative is based is often called the underlying.

For derivative products, the market where both parties meet (physically or not) can be either a *future exchange* or an *over the counter market*. The first one is the place where standardized derivative contract are traded, for example the Chicago Mercantile Exchange. For more specific and complex products, the exchange take place on "over the counter" market. On this market, financial institutions and corporate clients meet to design contracts that answer to specific needs from the investors. Thus we see here a major explanation for the increasingly complexity of financial instruments, as the market evolve the special requests for tailor made products increase and with it, the sophistication of financial products.

An important concept related to financial market is the one of market equilibrium and market efficiency. It can be traced back to pioneering work by Louis Bachelier (1870-1946), while in more modern era the study by Samuelson [Sam65]bears critical importance. The market efficiency hypothesis states that when a market is at its equilibrium point where assets reached their fair price, all necessary information is embedded in an asset price and any displacement from said equilibrium must be random up to a drift [CLM97]. New information and unexpected events do change the market equilibrium and move the market prices to a new equilibrium state, rather than an equilibrium is forever the same and random events can not change it. The type of information embedded in the price of an asset leads to different form (weak vs. strong) form of market efficiency hypothesis. Whether this double hypothesis (of efficiency and a particular equilibrium) is true or not, is a matter of discussion for economists and will not be our concern here. We mention it here, since the concept of arbitrage explained later on is connected to the efficiency of a market.

A metric associated to any financial instrument is the risk. Obviously any investor buying a risky asset at time t (say a stock with price $S(t)$) and holding it for a duration T expects to make profit out of it, a return R, to compensate for the associated risk

$$R = \frac{S(t+T) - S(t)}{T} \tag{3.1}$$

and typically the greater the risk, the greater should be the expected return on this asset. Mathematically the risk associated with the asset S, called volatility σ in the financial world is measured by

$$\sigma^2 = \mathbb{E}\left[R - \bar{R}\right] \tag{3.2}$$

where $\bar{R}$ is the expected return as opposed to R the actual return. So the volatility is just the standard deviation of an asset return. If any asset is considered to be risk-free, then it should have the same return as a reference risk free asset (typically government issued treasury bill) growing at a so called "risk free rate".

3.1.1 Arbitrage and risk neutral measure

A popular saying when efficient market hypothesis is discussed is the famous “there should be no free lunch with vanishing risk”. It is rather sound, one could hardly argue that a market could be efficient if without an initial investment one could lock in a sure profit. This leads to the notion of arbitrage free market. Arbitrage opportunity actually acts as a mechanism by which the market could revert to equilibrium. Since investors identifying an arbitrage opportunity, say a under-priced stock, would trade in this stock and drive the price higher where arbitrage would vanish.

The mathematical definition of an arbitrage follows

Definition 3.1.1: Arbitrage strategy

Let $V(t)$ be a portfolio strategy then V is an arbitrage strategy if $\exists t > 0$ such that

$$V(0) = 0 \text{ and } \mathbb{P}[V_t \geq 0] = 1 \text{ while } \mathbb{P}[V_t > 0] > 0 \tag{3.3}$$

The proper definition of an admissible portfolio strategy can be found in [Pas11], since we will not focus on the replication problem we will not elaborate on portfolio. The definition is otherwise self explanatory.

We continue the mathematical description of a financial market with the following theorem [Shr04] that makes the connection between arbitrage and the existence of a specific measure

Theorem 3.1.1: First fundamental theorem of asset pricing

$\widetilde{\mathbb{P}}$ is a risk-neutral measure (equivalent martingale measure) if

1. $\widetilde{\mathbb{P}}$ is absolutely continuous with respect to the real world measure $\mathbb{P}$, i.e. they agree on their null sets.
2. Under $\widetilde{\mathbb{P}}$ the discounted stock price is a martingale.

First fundamental theorem of asset pricing: If a market has a risk-neutral measure, then it is arbitrage free

It is necessary to discount with respect to a rate $R(t)$ in order to account for the time value of money: a payoff obtainable in the future has a lesser value thatn if it were available right now. Therefore we will see often stick the term $\exp\left[-\int\limits_t^T R(s)\,ds\right]$ to an expected payoff.

The relevance of the risk-neutral measure, and hence the relevance of the no arbitrage condition, to the problem of pricing the option contract is now made explicit by the following formula

Proposition 3.1.1: Risk neutral pricing formula

Let $\widetilde{\mathbb{P}}$ be a risk-neutral measure, $V(T)$ the payoff at time T of a derivative security and $R(t)$ the interest rate of the money market, we have

$$V(t) = \widetilde{\mathbb{E}}\left[e^{-\int\limits_t^T R(s)ds} V(T)\,|\mathcal{F}_t \right] \tag{3.4}$$

this formula is of paramount importance in the martingale approach to option pricing. Also it allows to easily reach an expectation form, and we will see later in Section 4.2 how it can then be exploited by the Feynman-Kac formula. Now an example can be helpful in seeing how risk neutral measure can be found, take a stock whose price process $S(t)$ follows the dynamic

$$dS_t = S_t\left(\mu dt + \sigma dW_t\right) \Rightarrow S_t = S_0 \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right] \tag{3.5}$$

which is a GBM as seen in (2.27). This process is used in the Black Scholes model since it assumes only positive values which is a realistic assumption about stock prices. Under real-world probability measure $\mathbb{P}$, we have

$$\mathbb{E}\left[e^{-rt}S_t|\mathcal{F}_0\right] = S_0 e^{(\mu-r)t} \neq S_0 \tag{3.6}$$

therefore because the drift term does not cancel out, S_t is not a martingale. Now, if we switch to an equivalent probability measure $\widetilde{\mathbb{P}}$ defined by the exponential martingale

$$\left.\frac{d\widetilde{\mathbb{P}}}{d\mathbb{P}}\right|_{\mathcal{F}_t} = \varepsilon\left(\int\limits_0^t \frac{\mu-r}{\sigma}dW_s\right) \equiv \exp\left[-\frac{t}{2}\left(\frac{\mu-r}{\sigma}\right)^2 - \left(\frac{\mu-r}{\sigma}\right)W_t\right] \tag{3.7}$$

then according to Girsanov theorem 2.2.1, $\widetilde{W_t} = W_t + \frac{\mu-r}{\sigma}t$ is a $\widetilde{P}$-Brownian motion. Now under the new measure

$$\begin{aligned}
\mathbb{E}\left[e^{-rt}S_t|\mathcal{F}_0\right] &= e^{-rt}\mathbb{E}\left[S_0 \exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma W_t\right]\middle|\mathcal{F}_0\right] \\
&= S_0 e^{-rt}\widetilde{\mathbb{E}}\left[\exp\left[\left(\mu - \frac{\sigma^2}{2}\right)t + \sigma\left(\widetilde{W}_t - \frac{\mu-r}{\sigma}t\right)\right]\middle|\mathcal{F}_0\right] \\
&= S_0\widetilde{\mathbb{E}}\left[\exp\left[-\frac{1}{2}\sigma^2 t + \sigma\widetilde{W}_t\right]\middle|\mathcal{F}_0\right] \\
&= S_0 e^{\frac{-\sigma^2 t}{2}}\widetilde{\mathbb{E}}\left[\exp\left[\sigma\widetilde{W}_t\right]\middle|\mathcal{F}_0\right] \\
&= S_0 e^{\frac{-\sigma^2 t}{2}} e^{\frac{\sigma^2 t}{2}} \\
&= S_0
\end{aligned} \tag{3.8}$$

Therefore we saw that by using a particular "risk-neutral" measure instead of the "real world" measure it was possible to reach a martingale form for the discounted asset price process. Thus there is at least one risk neutral measure, which allows us to say that this particular description of the market is arbitrage free. And we can then use this measure to get the fair price for some class of derivatives using the formula in proposition 3.1.1. Change of measure is an especially prevalent technique in the martingale approach to option pricing as we will show later on.

3.2 Derivatives

A derivative, as the name suggests, is a security whose value depends on an underlying security value. Such a security could be a stock, a bond, a currency, an index, etc. Derivatives are traded mainly between two places: on a public exchange, or over-the-counter (OTC). Public exchanges comprise places like the Chicago Board of Options Exchange (CBOE), the American Stock and Options Exchange (AMEX), the Chicago Board of Trade(CBOT), the Korea Exchange which is the largest (in transaction volume) in the world, and other places. Over-the-counter derivatives are always traded directly between the two interested parties, typically a financial institution and a corporate client. Exotic derivatives are almost always traded in an OTC manner.

According to the Bank for international settlements statistics[oIS14], the OTC derivative market notional amounts outstanding was valued around 710 trillion dollars in December 2013.

3.2.1 The use of derivatives

The most general uses of derivatives are hedging and speculation.

Hedging is a general term used to cover techniques employed to reduce risks associated with holding position in a contract, or even more broadly, unforeseeable risks (e.g. weather extreme events). In the later case, let us think about a Japanese company that will need to buy some amount of goods coming from France 6 months from now, the EUR/JPY currency exchange rate is fine as of today but is highly fluctuating and therefore there is a high risk that it will rise over a sustainable threshold. The company decides to lock-in the current rate by purchasing some contracts that allow the exchange 6 months from now Euro for Yen, at the writing day's rate. If 6 months from now, the rate did rise then the company effectively lock a profit. If however the rate fell, then the company would have been better off not purchasing those contracts. The initial goal not being to make a profit, then the actual evolution does not really matter to the company hedging risk. Hedging also occurs within a portfolio when some instruments are included in order to offset the random fluctuations of other instruments in this portfolio. In a perfectly hedged portfolio, fluctuations of participating instruments perfectly cancel out, leading to a portfolio whose dynamics is deterministic, therefore free from risk associated to random movements.

Speculation is another broad term that describes situation where one enters a contract in order to make a profit from the stock moving in an expected direction. If one expects a stock S, today valued at 100$ to go down in the future, then one can purchase the right to sell in the future at the current value: 100$. If the price does indeed go down, say to 80$, then the holder of those contracts will purchase stocks on the market at 80$ and use its contract to sell them at 100$, thus making a 20$ benefit on each contract he holds.

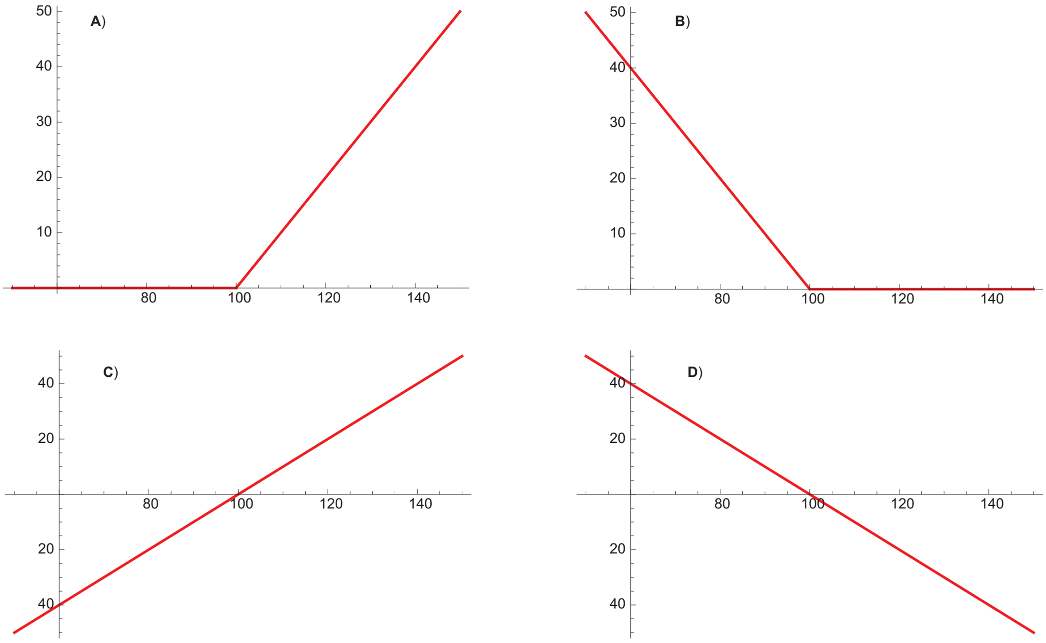


Figure 3.1: Behaviour of the payoff function for some derivatives, for each panel the strike has been set at 100 (in arbitrary unit), the horizontal axis is the stock value at expiration while the vertical axis is the payoff value. Panel **A**: Call option, **B**: Put option, **C**: Long forward, **D**: Short forward.

3.2.2 A tour of derivative contracts

We will give here some descriptions of very popular and straightforward derivatives, while visual examples for their payoffs is available on .

Future contracts are contracts traded on public exchange such as the CBOT or the CME. This highly standardized contract is an agreement to buy or sell a specified asset at a prescribed future date at a certain price. The party buying is said to have a long position (*to be long*), while the party agreeing to sell the asset is said to have a short position in the contract (*to be short*). The exchange is in charge of matching the buyer and the seller, it also takes under the task of clearing house dealing with margin requirements, daily settlements [Hul08]. Future contracts must usually specify: the asset, the contract size , the delivery arrangements, the delivery months, price quotes.

Forward contracts share the same purpose, and most of the characteristics of the Future contracts. However unlike a future contract, a forward is traded directly OTC, in order to account for scenarii where standardized features do not meet an investor special needs.

An **option** contract is a derivative that gives the holder, the right and no obligation, to buy (Call option) or sell (Buy option) an asset at a future exercise date for a certain price. The party writing the option is short, while the party purchasing the contract is long. The immediate difference between an option and a future, is the added perks of not exercising the contract if it were not beneficial. The other striking difference between the former derivatives and an options is the vastness of subtype when it comes to describing options. The axis along which classification occurs are the exercise type and the payoff function. An option can either have a fixed exercise date, then they are called European options, if an option can be exercised at any time then it is called American. In between, with a set of exercise dates to chose from is the Bermudan option. The spectrum lying on the payoff axis is not that self contained. The first distinction is made between path dependent option and independent. For example an option written on an underlying asset S_t that pays at exercise date T: $\max_{0<t<T}[S_t - 100\$]$, is path dependent since the payoff function needs the whole history of the path to be computed. However a payoff like $[S_T - 100\$]$ just needs the price at exercise date T and does not care about how it reached that particular value. It is mainly among the path dependent (a subset of the exotic options) that lies complexity in options, and we give here some example of exotic options

- Barrier option: knocks in/out if an asset price crosses a defined threshold.
- Asian option: pays a function of the average of the asset price over its lifetime.
- Lookback: pays a function on the max/min of the asset until exercise date.
- Rainbow option: payoff is written on the best worst performance of a set of underliers.

For an option, two major challenges arise: the pricing problem and the hedging problem. The pricing problem is concerned with finding the fair value (arbitrage free) for an option. It can be straightforward for Vanilla (e.g. simple non path dependent) options, to nearly analytically untractable for the exotic case. The hedging problem is concerned with hedging the risk associated with holding or writing an option, it can consist of rebalancing the ratio of assets we hold in a portfolio to more complex techniques. Since we are in this work only concerned with the so-called pricing problem of exotic options we will not elaborate on the hedging problem.

We are now going to introduce the model in which our options live and show how the pricing problem arise and can be solved easily for Vanilla options.

3.3 Black-Scholes model

The Black-Scholes model is the result of the work by Robert Merton, Fischer Black and Myron Scholes, published in 1973[BS73]. It describes, under certain assumptions, a PDE describing the time evolution for the price of an option. The original model assumptions were the following

1. The rate of return r on a risk free asset is constant and known.
2. The log of the return for an asset price process S_t is modeled by a brownian motion, with constant drift μ (the expected return on stock) and constant volatility σ $\frac{dS_t}{S_T} = \mu dt + \sigma dW_t$
3. The stock does not pay dividend.

The first assumption just requires that the rate of return for, say a government issued bond, is constant. This is not actually true as rates fluctuate over time, however since options are usually not hold over a lengthy period (usually one year), the rate will not vary much therefore the assumption is not that coercive. The second assumption means that while the price itself is log-normally distributed, the returns are normally distributed. The assumption of a constant volatility is however quite restrictive as it limits severely complex dynamics such as volatility clustering[Man63] and has been the impetus for the development of a variety of other stochastic volatility models: Heston model[Hes93], CEV model[Cox75],etc. The last assumption is not restrictive and will just change the drift term if accounted for. Under those assumptions they found that the option fair value $V(.,.)$ for an option written on S_t satisfies

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

We will show in the following subsection a way to derive this PDE. Meanwhile, let us point out the following important proposition[JYC09] that will allows us to use proposition 3.1.1 for option pricing

Proposition 3.3.1: The Black-Scholes model is complete

In the Black Scholes mode there is an equivalent risk free measure $\widetilde{\mathbb{P}}$

$$\frac{d\widetilde{\mathbb{P}}}{d\mathbb{P}}\bigg|\mathcal{F}_t = \varepsilon\left(\int_0^t \frac{\mu - r}{\sigma} dW_s\right) \tag{3.10}$$

furthermore, this measure is unique. Then the risk free dynamics is given by

$$\frac{dS_t}{S_T} = rdt + \sigma dW_t \tag{3.11}$$

As we can see, the effect of switching to the risk neutral measure is to replace the original drift term μ with the risk free rate r.

3.3.1 Deriving and solving the Black Scholes equation

The first derivation we give here is based on the well-known (any textbook) portfolio argument. Let $V(S, t)$ be the value of an option written on S at time t. Based on the Ito lemma then the differential of V is

$$dV = \left(\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} + \mu S\frac{\partial V}{\partial S}\right)dt + \sigma S\frac{\partial V}{\partial S}dW_t \tag{3.12}$$

We build a portfolio Π made of one option and $-\Delta$ stock, Δ being decided at the beginning of a "dt increment" then held constant over this increment (i.e. admissible strategy). The differential of the portfolio is

$$d\Pi = dV - \Delta dS \tag{3.13}$$

Plugging (3.12) and the definition for dS in (3.13) yields

$$d\Pi = \left(\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} + \mu S\frac{\partial V}{\partial S} - \Delta\mu S\right)dt + \sigma S\left(\frac{\partial V}{\partial S} - \Delta\right)dW_t \tag{3.14}$$

Here we can see that by taking $\Delta = \frac{\partial V}{\partial S}$, the stochastic component vanishes and our portfolio becomes deterministic. Then, by arbitrage argument we can show that this riskless (deterministic) portfolio should not grow faster than the risk free rate, otherwise one could lock in a sure profit. Therefore we should have

$$d\Pi = r\Pi dt \tag{3.15}$$

equating (3.14) and (3.15), we have

$$\begin{aligned} r\Pi dt &= \left(\frac{\partial V}{\partial t} + \frac{\sigma^2}{2}S^2\frac{\partial^2 V}{\partial S^2} + \mu S\frac{\partial V}{\partial S} - \Delta\mu S\right)dt \\ \Leftrightarrow r\left(V - \frac{\partial V}{\partial S}S\right) &= \frac{\partial V}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} \\ \Leftrightarrow rV - \frac{\partial V}{\partial S}rS - \frac{\partial V}{\partial t} - \frac{\sigma^2 S^2}{2}\frac{\partial^2 V}{\partial S^2} &= 0 \end{aligned} \tag{3.16}$$

rearranging and changing signs around yields the Black Scholes equation (3.9). Let's note that the equation does not contain the drift parameter μ, therefore its value is independent of its growth rate, also, the share Δ of the stock to hold has been seen to be equal to $\frac{\partial V}{\partial S}$. The various derivatives of V with respect to S and t are called greeks[Pas11] and describe the sensitivity of the option price to variation of underlying asset value.

To solve the Black Scholes equation (3.9), say for a call option written at time t for an exercise date T, a strike price K with a payoff that is $\max\left(S_T - K, 0\right)$ we need the boundary conditions

$$\begin{aligned} &V\left(0, T\right) = 0 \\ &\lim_{S_T \to \infty} V\left(S_T, T\right) = S \\ &V\left(S_T, T\right) = \max\left(S_T - K, 0\right) \end{aligned} \tag{3.17}$$

we start with the transformation $U \leftarrow e^{-rt}V$, after simplification (3.9) becomes

$$\frac{\partial U}{\partial t} + \frac{\sigma^2 S^2}{2}\frac{\partial^2 U}{\partial S^2} + rS\frac{\partial U}{\partial S} = 0 \tag{3.18}$$

then we use $x \leftarrow \ln\left(S\right), s \leftarrow T - t$

$$\frac{\partial u}{\partial s} = \left(r - \frac{\sigma^2}{2}\right)\frac{\partial u}{\partial x} + \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial x^2} \tag{3.19}$$

now we make the substitutions $y \leftarrow x + r - \frac{\sigma^2}{2}, \tau \leftarrow s$ yields

$$\frac{\partial u}{\partial \tau} = \frac{\sigma^2}{2}\frac{\partial^2 u}{\partial y^2} \tag{3.20}$$

the terminal condition in the payoff is now an initial condition. Solving this diffusion pde in the usual fashion leads to the fair price for a call option

$$V\left(S, t\right) = S\Phi\left(\frac{\ln\left(\frac{S}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)\left(T - t\right)}{\sigma\sqrt{\left(T - t\right)}}\right) - e^{-r(T-t)}K\Phi\left(\frac{\ln\left(\frac{S}{K}\right) + \left(r - \frac{\sigma^2}{2}\right)\left(T - t\right)}{\sigma\sqrt{\left(T - t\right)}}\right) \tag{3.21}$$

where $\Phi\left(x\right)$ is the normal cumulative distribution function.

The second method we present to derive the Black Scholes PDE will appeal to a martingale argument. We know from proposition 3.1.1 that the fair price (under risk-neutral measure) for our option can be written

$$\widetilde{\mathbb{E}}\left[e^{-rT}V(S_T)\middle|\mathcal{F}_t\right] \tag{3.22}$$

from theorem 2.3.2 we know that there is a function g such that

$$g(t, S_t) = \widetilde{\mathbb{E}}\left[e^{-rT}V(S_T)\middle|\mathcal{F}_t\right] \tag{3.23}$$

Computing its differential, plugging in the definition for dS and using Ito's lemma yields

$$\begin{aligned} dg &= e^{-rT}\left(-rVdt + dV\right) \\ &= e^{-rT}\left[-rVdt + T_t dt + V_S\left(Srdt + S\sigma d\tilde{W}_t\right) + \frac{\sigma^2 S^2}{2}dt\right] \\ &= e^{-rT}\left[\left(V_t + SrV_S + \frac{\sigma^2 S^2}{2} - rV\right)dt + S\sigma V_S d\tilde{W}_t\right] \end{aligned} \tag{3.24}$$

Since the process $g(t, S_t)$ is a martingale, hence a driftless process, we must have

$$-rV + V_t + SrV_S + \frac{\sigma^2 S^2}{2} = 0 \tag{3.25}$$

rearranging terms yields the Black Scholes PDE.

Chapter 4

Path Integrals

In this chapter we will describe the path integral concepts that will be relevant to this thesis and our work on option pricing. Since it is a field in and of itself, we will have to severely limit the exposition but readers can find further development in [FH12],[MC01] and [Kle09]. We will start by first giving an intuitive description of the path integral and its connection with Brownian motion with an orientation towards physics. Then we will lay out the technical concepts as originally introduced by Norbert Wiener (1894-1964) to model the Brownian motion. We will discuss the fundamental Feynman-Kac theorem before continuing our journey into physics with the description of path integral in quantum mechanics. The fundamental property of a quantum system is that is is described only probabilistically. The position of a quantum particle for example, and that is in contrast with classical mechanics is given as a distribution. It can be found in many states and each of those states is more or less likely. The probabilistic component is what makes some concepts and tools from quantum mechanics and stochastic process overlapped. We will finish this chapter by introducing the application of path integral to finance.

4.1 Wiener path integrals and stochastic processes

We already described some aspects of the Brownian motion in section 2.2 from the viewpoint of probability theory. This time around we will start our discussion with some physics consideration. Since the work of Albert Einstein (1879-1955)[Ein05] it is known that Brownian particles density $W(x,t)$ obeys a diffusion equation (in one dimension here to avoid cumbersome notation)

$$\frac{\partial W(x,t)}{\partial t} = D\frac{\partial^2 W(x,t)}{\partial x^2} \tag{4.1}$$

for a diffusion constant D, in this chapter we will take for reasons explained later on $D \equiv \sigma^2/2$. Normalized fundamental solution to (4.1) with i.c $\lim_{t\to 0} W(x,t) = \delta(x-x_0)$ for an

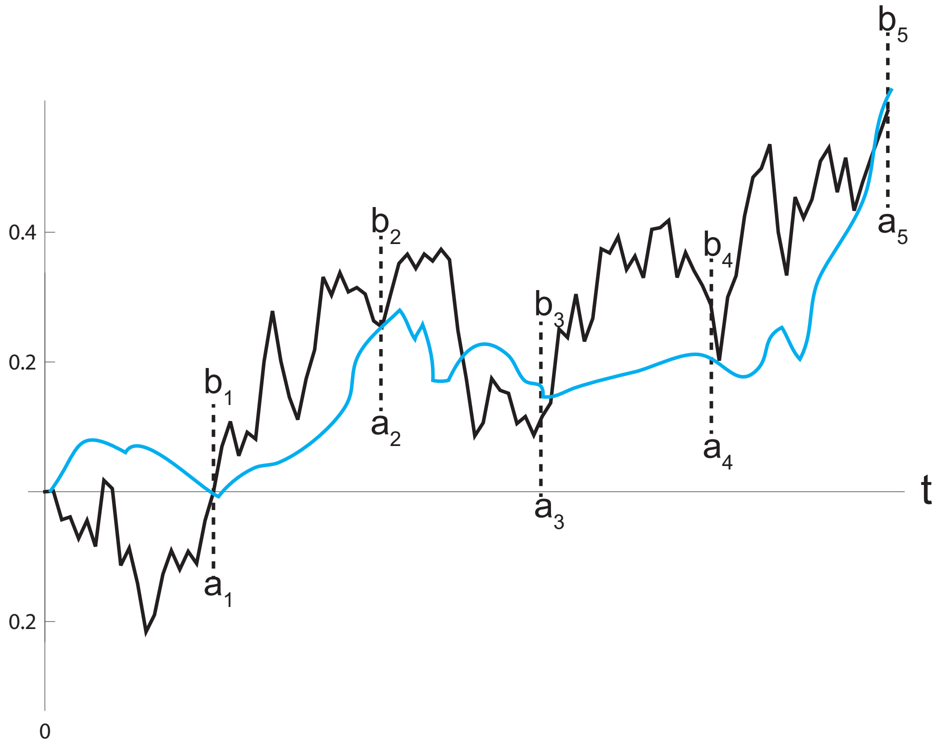


Figure 4.1: Example of two Brownian paths passing through a succession of gates, at some times $0 = t_0 < t_1 < ... < t_5$.

initial position x_0 is well known to be [Eva98]

$$W(x,t) = \frac{\theta(t)}{\sqrt{4\pi Dt}} \exp\left[-\frac{x^2}{4Dt}\right] \tag{4.2}$$

which is just a form of the Gaussian distribution. Since the Brownian motion is homogeneous both in space and time, the transition density $\mathbb{P}(y,T|x,t_0)$ describing the probability to reach y at time T starting from x at time t_0, only depends on the difference $(T-t_0)$ and $|y-x|$. Therefore we'll write interchangeably $\mathbb{P}(y,T|x,t_0)$ and $W(y,T|x,t_0) := W(|y-x|, T-t_0)$. As an example and to illustrate our introduction, the probability that a standard (starting at 0) Brownian particle X_t be at time $t > 0$ anywhere in the interval $[a,b]$ is given by

$$\mathbb{P}(X_t \in [a,b]\,|X_0 = 0) = \int_a^b \frac{1}{\sqrt{4\pi Dt}} \exp\left[-\frac{x^2}{4Dt}\right] \tag{4.3}$$

Obviously there are many (infinitely many) Brownian paths that pass through this interval at time t, as can be seen on Fig. 4.1. If we were to be interested in a particular path (say the black one in the figure aforementioned) we need to be able to distinguish paths, and we can do that by increasing the number of gates to cross, by increasing the number of gates we pin further and further the path along the way until we completely prescribe the path. The Chapman Kolmogorov equation will be helpful in handling probability of such compound events

Definition 4.1.1: Chapman Kolmogorov equation

For a Markov process X_t , whose transition density is given by $\mathbb{P}\left(y,T|x,t\right)$ we have the following equality

$$\mathbb{P}\left(z,T|x,t\right)=\int\limits_{-\infty}^{+\infty}\mathbb{P}\left(z,T|y,s\right)\mathbb{P}\left(y,s|x,t\right)dy \tag{4.4}$$

Since non overlapping increments are independent, the probability of the compound events factorize in products of single event probability and we have

$$\begin{aligned}&\mathbb{P}\left(X_{t_1}\in\left[a_1,b_1\right],X_{t_2}\in\left[a_2,b_2\right],...,X_{t_n}\in\left[a_n,b_n\right]\right)\\&=\int\limits_{a_1}^{b_1}W\left(x_1,t_1\right)dx_1\int\limits_{a_2}^{b_2}W\left(|x_2-x_1|,t_2-t_1\right)dx_2\times...\times\int\limits_{a_n}^{b_n}W\left(|x_n-x_{n-1}|,t_n-t_{n-1}\right)dx_n\end{aligned} \tag{4.5}$$

If one increases the number of gates $[a_i, b_i]$ to be crossed by the Brownian particle and as a result let $\Delta t := t_{i+1}-t_i \to 0$, then we recover the stochastic process continuously depending on t we are interested in. The probability that the Brownian particle wanders through the infinite number of gates of "length" dx that we specify is then the probability of a specific trajectory, looking at the integrands

$$\begin{aligned}&\mathbb{P}\left[x\left(t\right)\in\left[A,B\right]\right]\\&=\int\limits_{\mathcal{C}\{0,0;[A,B],t\}}\lim_{\substack{\Delta t_i\to0\\N\to\infty}}\exp\left[-\sum_{i=1}^{N}\frac{\left(x_i-x_{i-1}\right)^2}{2\sigma^2\left(t_i-t_{i-1}\right)}\right]\prod_{i=1}^{N}\frac{dx_i}{\sqrt{2\sigma^2\left(t_i-t_{i-1}\right)}}\\&=\int\limits_{\mathcal{C}\{0,0;[A,B],t\}}\lim_{\substack{\Delta t_i\to0\\N\to\infty}}\exp\left[-\frac{1}{2\sigma^2}\sum_{i=1}^{N}\left(\frac{x_i-x_{i-1}}{t_i-t_{i-1}}\right)^2\Delta t_i\right]\prod_{i=1}^{N}\frac{dx_i}{\sqrt{2\sigma^2\Delta t_1}}\\&\equiv\int\limits_{\mathcal{C}\{0,0;[A,B],t\}}\exp\left[-\frac{1}{2\sigma^2}\int\limits_0^t\dot{x}^2\left(s\right)ds\right]\prod_{i=1}^{N}\frac{dx(s)}{\sqrt{2\sigma^2ds}}\\&=\frac{1}{\sqrt{2\pi\sigma^2t}}\int\limits_A^B\exp\left(-\frac{x^2}{2\sigma^2t}\right)dx\end{aligned} \tag{4.6}$$

the symbol $\mathcal{C}\{x,0;y,t\}$ is formally used to specify the set of paths that starting from x at time $t=0$, will reach y at time t. Equation 4.6 is the first example of a path integral, an infinite dimensional integral where the integrand represents short-time Gaussian-like transition. Practically the time interval $[0, t]$ between the start and end points is sliced $x = x_{t_0=0} < ... < x_{t_{N+1}=t} = y$ and intermediary points are integrated over using the Gaussian density. Alternatively and to echo concepts introduced in section 2.1, the path integral can be conceived as an integration using the Wiener measure over a subset of the functional infinite dimensional space of continuous non differentiable trajectories[MC01] obeying some prescribed conditions.

Definition 4.1.2: Wiener measure

The (pinned) wiener measure $dW_x(\tau)$ is formally defined as

$$dW_x(\tau) := \exp\left[-\frac{1}{2\sigma^2}\int_0^t \dot{x}^2(\tau)\,d\tau\right]\prod_{\tau=0}^{t}\frac{dx(\tau)}{\sqrt{2\pi\sigma^2 d\tau}} \tag{4.7}$$

where σ^2 is a constant related to the volatility or variance of the path. The integrand term in the exponential is referred to as *the Lagrangian*.

This measure is meant to be taken as the limit for N intermediate points X_i, $N \to \infty$ of the following finite dimensional integrals

$$\frac{1}{(2\pi\sigma^2\Delta t)^{(N+1)/2}}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}...\int_{-\infty}^{\infty}\exp\left[\frac{-1}{2\sigma^2\Delta t}\sum_{i=0}^{N}(x_{i+1}-x_i)^2\right]dx_N...dx_2dx_1 \tag{4.8}$$

where $\Delta t := \frac{t_N - t_0}{N+1} \to 0$, x_0 is the starting point at $t = t_0$, and x_{N+1} the end point at time t_N.

The transition probability is then the following path integral using the Wiener measure

$$W(|x_t - x_0|, t) \equiv \mathbb{P}(x_t, t|x_0, 0) = \int_{\mathcal{C}\{x_0,0;x_t,t\}} dW_x(\tau) \tag{4.9}$$

Needless to say that it is necessary to lay properly the theoretical foundation of the path integral, and present the wiener measure, however the reader should be advised that we will hardly need to go to such a low level and see later on that some important shortcuts or equivalences can be taken.

We are going to go through an example to see that the computation is not as cumbersome as it may seem, let us compute the probability that a brownian particle starting at 0, returns at 0 at time t, so here $x_0 = x_{N+1} = 0$, and $(t-0) = (N+1)\,\Delta t$.

$$
\begin{aligned}
\mathbb{P}\left(0,t|0,0\right) &= \int\limits_{\mathcal{C}\{0,0;0,t\}} dW_x\left(\tau\right) \\
&= \lim_{\Delta t\to 0} \frac{1}{\left(2\pi\sigma^2\Delta t\right)^{(N+1)/2}} \int\limits_{-\infty}^{\infty} ... \int\limits_{-\infty}^{\infty} \exp\left[\frac{-1}{2\sigma^2\Delta t}\sum_{i=0}^{N}\left(x_{i+1}-x_i\right)^2\right] dx_N ... dx_1 \\
&= \lim_{\Delta t\to 0} \frac{1}{\left(2\pi\sigma^2\Delta t\right)^{(N-1)/2}} \int\limits_{-\infty}^{\infty} ... \int\limits_{-\infty}^{\infty} \exp\left[-\sum_{i=0}^{N-2}\frac{\left(x_{i+1}-x_i\right)^2}{2\sigma^2\Delta t}\right] \Bigg\{ \\
&\times \int\limits_{-\infty}^{\infty} \frac{\exp\left[\frac{-\left(x_{N+1}-x_N\right)^2-\left(x_N-x_{N-1}\right)^2}{2\sigma^2\Delta t}\right]}{2\pi\sigma^2\Delta t} dx_N \Bigg\} dx_{N-1}...dx_1 \\
&= \lim_{\Delta t\to 0} \frac{1}{\left(2\pi\sigma^2\Delta t\right)^{(N-2)/2}} \int\limits_{-\infty}^{\infty} ... \int\limits_{-\infty}^{\infty} \exp\left[-\sum_{i=0}^{N-3}\frac{\left(x_{i+1}-x_i\right)^2}{2\sigma^2\Delta t}\right] \Bigg\{ \\
&\times \int\limits_{-\infty}^{\infty} \frac{\exp\left[-\frac{\left(x_{N+1}-x_{N-1}\right)^2}{2\sigma^2(2\Delta t)} - \frac{\left(x_{N-1}-x_{N-2}\right)^2}{2\sigma^2\Delta t}\right]}{\sqrt{2\pi\sigma^2\Delta t}\sqrt{2\pi\sigma^2\left(2\Delta t\right)}} dx_{N-1} \Bigg\} dx_{N-2}...dx_1
\end{aligned}
\tag{4.10}
$$

clearly a recursion process takes place: at the first integration over x_N, $\epsilon \to 2\epsilon$, after the second integration $2\epsilon \to 3\epsilon$...

After N integrations there is no longer any dependence on N, and we are left with

$$
\mathbb{P}\left(0,t|0,0\right) = \frac{\exp\left[-\frac{\left(x_{N+1}-x_0\right)^2}{2\sigma^2(N+1)\epsilon}\right]}{\sqrt{2\pi\sigma^2\left(N+1\right)\epsilon}}
$$

plugging in $\left(N+1\right)\epsilon = t$ and $x_0 = x_{N+1} = 0$

$$
\mathbb{P}\left(0,t|0,0\right) = \frac{1}{\sqrt{2\pi\sigma^2 t}} \tag{4.11}
$$

Obviously we did know that result and did not to carry on the task with such a heavy tool, but it helps to see that there is a proper definition that lays at the foundation of path integration.

Let us summarize our discussion to fix ideas. We started by describing a Brownian particle trajectory between 0 and t, first by recalling that the transition density describing Brownian motion obeyed a diffusion equation. From the solution to the diffusion equation, we delved deeper into the intermediate trajectories by time slicing $[0,t]$ and pinning its coordinate at some intermediate times t_i, relying on (4.4) to join the slices. In the end and by increasing the number of t_i until it *covers* the $[0,t]$ interval, we described a stochastic process in the

sense of definition 2.2.1: the Wiener process. It should become increasingly clearer to the reader, at least on an intuitive level, that there is a deep connection between stochastic processes, path integrals and partial differential equations (PDE).
Hence let us make that more explicit by pointing out the second equivalence relations between path integrals and PDE

Proposition 4.1.1: Fokker Planck equation

To each path integral of the form

$$\mathbb{P}\left(x_t, t|x_0, 0\right) = \int\limits_{\mathcal{C}\{x_0,0;x_t,t\}} dW_x\left(\tau\right) \tag{4.12}$$

describing a transition probability, corresponds a Fokker Planck PDE

$$\frac{\partial P}{\partial t} + \frac{\partial\left(\mu P\right)}{\partial x_t} - \frac{1}{2}\frac{\partial^2\left(\sigma^2 P\right)}{\partial x_t^2} = 0 \tag{4.13}$$

written on the Ito process X_t $(\equiv x_t)$

$$dXt = \mu\left(t, X_t\right)dt + \sigma\left(X_t, t\right)dW_t \tag{4.14}$$

Let us give an example of how to use this equivalence. Suppose we are interested in the probability that a Brownian particle with constant drift μ and volatility σ starting from 0 reaches the point y at time t. The SDE is

$$dX_t = \mu dt + \sigma dW_t \tag{4.15}$$

then according to proposition 4.1.1 the transition probability obeys

$$\frac{\partial P}{\partial t} = \frac{\sigma^2}{2}\frac{\partial^2 P}{\partial x_t^2} - \mu\frac{\partial P}{\partial x_t}, \tag{4.16}$$

which is just a convection diffusion PDE. To solve, introduce moving coordinates $\xi = x - \mu t$ and $\eta = t$, yielding the diffusion equation $P_\eta = \sigma^2/2P_{\xi\xi}$ whose solution we know to be (4.2). All that is left is to revert back to original variables, apply normalizing and initial conditions to get the solution

$$\mathbb{P}\left(y, t|0, 0\right) = \frac{\exp\left[\frac{-(y-\mu t)^2}{2\sigma^2 t}\right]}{\sqrt{2\pi\sigma^2 t}} \tag{4.17}$$

4.2 The Feynman-Kac formula

In the previous section we introduced the theoretical foundation upon which path integrals were built. And although it was necessary, the real potential of path integrals will be made more explicit in this section. Up until now, we did not really discuss Wiener integral of functionals, limiting ourselves to the mere integration of the Wiener measure.

What if we are interested in the following Wiener integral (describing the expectation w.r.t. the Wiener measure)

$$\int\limits_{\mathcal{C}\{x_0,0;x_t,t\}} F\left[x\left(\tau\right)\right] dW_x\left(\tau\right) \tag{4.18}$$

for a functional F on $x\left(\tau\right)$? We could follow the time-slicing method as introduced in the previous section and define simple functionals F_n that approximate in a piecewise constant fashion the functional F

$$\begin{aligned} F_n[x\left(\tau\right)] &:= \int\limits_{-\infty}^{+\infty} ... \int\limits_{-\infty}^{+\infty} F_n\left(x_1, ..., x_n\right) \mathbb{1}_{\{x(t_1)=x_1,...,x(t_n)=x_n\}} dx_1...dx_n \\ I_n &:= \int\limits_{\mathcal{C}\{x_0,0;x_t,t\}} F_n\left[x\left(\tau\right)\right] dW_x\left(\tau\right) \end{aligned} \tag{4.19}$$

then since those simple functionals form a vector space with a well defined norm [MC01][1], we can discuss their convergence and define the limit if the space is complete

$$\int\limits_{\mathcal{C}\{x_0,0;x_t,t\}} F\left[x\left(\tau\right)\right] dW_x\left(\tau\right) := \lim_{n\to\infty} I_n \tag{4.20}$$

For obvious reasons, this construction poses some integrability constraints on the type of functionals available.

For an important class of functionals we have the following theorem[Kac66] from Mark Kac(1914-1984) and Richard Feynman(1918-1988) that alleviates the need to perform the (infinitely many) integrations

[1] Although the cited text does not mention the completeness of such a space, it alludes to the fact that simple functionals are dense in the space of continuous function we are interested in

Theorem 4.2.1: The Feynman-Kac formula

For $V(x)$ lower bounded and C^2 on $\mathbb{R}$, its expectation rewritten under a path integral form $W(x_t, t|x_0, 0)$

$$\mathbb{E}\left[e^{-\int\limits_0^t V(x(s))ds} |X_0 = x_0\right] = \int\limits_{\mathcal{C}\{x_0,0;x_t,t\}} \exp\left[-\int\limits_0^t V(x(s))\,ds\right] dW_x(\tau) \qquad (4.21)$$

is the fundamental solution to the Bloch PDE

$$\frac{\partial W}{\partial t} = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial x_t^2} + \mu\frac{\partial W}{\partial x_t} - V(x_t)\,W \text{ , with } W(x_t, t|x_0, T) = \delta(x_T - x_0) \qquad (4.22)$$

written on the Ito process

$$dX_t = \mu dt + \sigma dW_t \qquad (4.23)$$

Proof for theorem 4.2.1 can be found in the original paper by Kac[Kac66]. The $V(x)$ function is often termed potential for reason made clearer when discussing path integrals in the quantum mechanics setting, another interpretation for $V(x)$ is that of a killing term for the diffusion X_t.

We will work now a simple example of the use of the Feynman-Kac formula: take $V(x) = \alpha^2 x^2$ for a driftless ($\mu = 0$) Ito process starting at 0, then the path integral

$$W(x_t, t|0, 0) := \int\limits_{\mathcal{C}\{0,0;x_t,t\}} \exp\left[-\alpha^2 \int\limits_0^t x_\tau^2 d\tau\right] dW_x(\tau) \qquad (4.24)$$

is the solution to

$$\frac{\partial W}{\partial t}(x_t, t) = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial x_t^2}(x_t, t) - \alpha^2 x_t^2 W(x_t, t) \text{ with i.c } W(x_t, t)|_{t=0} = \delta(x_t) \qquad (4.25)$$

From table of worked-out PDEs[Pol01], applying normalizing and initial conditions, we find

$$W(x_t, t|0, 0) = \sqrt{\frac{\alpha}{\pi \sinh(\alpha t)}} e^{-\alpha x_t^2 \coth(\alpha t)} \qquad (4.26)$$

If it were not for the Feynman-Kac formula, the evaluation of the path integral (4.24) would require a time-slicing approach and the use of the Gelfand-Yaglom method[GY60]. We will give in chapter 6 a more powerful example of the application of Feynman-Kac formula for the pricing of time occupation derivatives.

4.3 Path integrals in quantum mechanics

The path integral formulation of quantum mechanics proposed by Richard Feynman, describes the probability amplitude between two points in space time as a functional integral over all the possible paths, each one being weighted by an adequate measure. This is to be contrasted with the classical mechanics where the path that is followed by particle is completely determined by solving the Euler Lagrange equations, and is found to be the one that minimizes the classical action.

4.3.1 From the Schrodinger formulation

One of the most traditional way quantum mechanics is introduced is through the Schrodinger formulation of quantum mechanics. In this formulation, the main object of interest is the state function $\Psi\left(t\right)$, describing the state of a quantum system[2]. At our level, the state function will describe the existence of a quantum particle whose degree of freedom we are interested in, is its position x. The fundamental object to introduce next, when interested in the dynamics of our quantum particle, is the Hamiltonian operator H. The Hamiltonian operator is the one in charge of describing at a later time T a state function given at time $t < T$ through the linear Schrodinger equation (here time-dependent)

$$\frac{\partial\Psi}{\partial t} = \frac{-i}{\hbar}H\Psi \tag{4.27}$$

with formal solution

$$\Psi\left(T\right) = e^{-i(T-t)H/\hbar}\Psi\left(t\right) \tag{4.28}$$

The time independent version of the Schrodinger equation takes the eigenvalue equation form

$$E\Psi = H\Psi \tag{4.29}$$

with E the energy of the state Ψ. The importance of the time independent equation lies partially in the fact that solution to the time dependent equation can be built as an infinite sum of solution to the independent one [Gri05]. The Hamiltonian operator is customarily of the form $T+V$ with T a kinetic term of differential form and V a potential term, describing the force acting on the quantum particle. As an example the Hamiltonian for a free particle

[2] Mathematically the state function is a member of an Hilbert space, ensuring some nice property (e.g. existence of complete orthonormal basis) when working within. Note that solutions to the time independent Schrodinger equation form a complete orthonormal basis[Baa04], therefore solution to the time dependent Schrodinger equation since they live in the same space can be written w.r.t. those basis vectors

(absence of any potential term) with mass m is

$$H_{free} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} \tag{4.30}$$

and the solution to

$$-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} = E\Psi \tag{4.31}$$

with $E \in \mathbb{R}^+$ is found to be

$$\Psi_E = C_1 e^{i\sqrt{2mE/\hbar^2}x} + C_2 e^{-i\sqrt{2mE/\hbar^2}x} \tag{4.32}$$

where $C_{1,2}$ are constant to be found after normalization[3] . Another textbook example would be the Hamiltonian for a particle trapped inside a well with infinitely high wall at position $x = 0$ and $x = a$ is described by

$$H_{well} = -\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V(x) \tag{4.33}$$

with

$$V(x) = 0 \text{ if } 0 < x < a \text{ and } V(x) = +\infty \text{ otherwise} \tag{4.34}$$

the infinite potential effectively preventing the particle from ever crossing it the barrier. Even though it does not constitute any kind of challenge to solve it [4], we will not work out the solution here since it is outside of the scope of this thesis.

4.3.2 to the Feynman formulation

A fundamental object in the Schrodinger flavor of quantum mechanics was the Hamiltonian operator, describing how states evolve in time. However, in the Feynman formulation of quantum mechanics, it is the Lagrangian $L(\dot{x}, x, t)$ that occupies the center of attention. It is based on the Lagrangian that the so-called propagator is built. The propagator[5] is the object responsible for weighing possible paths in time for the quantum particle under study, when working in the path integral formulation of quantum mechanics. One is often not too far-off by thinking of a propagator as the transition function described in Wiener path integral.

[3] based on its interpretation as a probability, and integrated over all final states $|\Psi(x)|^2$ must evaluate to unity.

[4] Once we write down the continuity condition of Ψ everywhere, and $\partial\Psi$ everywhere except at the wall. Additionally $\Psi(0) = \Psi(a) = 0$...

[5] also called Kernel or Green function(for it is the fundamental solution to Schrodinger equation) in some texts

Definition 4.3.1: Propagator

For a quantum particle describing a path $x\left(t\right)$, with mass m and subject to a potential energy $V\left(x,t\right)$, the Lagrangian $L\left(\dot{x},x,t\right)$ and the classical action $\mathcal{S}\left[a,b\right]$ read

$$\begin{aligned} L\left(\dot{x},x,t\right) &= \frac{m}{2}\dot{x}^2 - V\left(x,t\right) \\ \mathcal{S}\left[a,b\right] &= \int\limits_{t_a}^{t_b} L\left(\dot{x},x,t\right)dt \end{aligned} \tag{4.35}$$

the action functional being taken on path such that $x\left(t_a\right) = x_a$ and $x\left(t_b\right) = x_b$. The propagator describing the probability *amplitude*[a] for this system $K\left(b,a\right)$ is defined by

$$K\left(b,a\right) := \int e^{\frac{i}{\hbar}\mathcal{S}[a,b]}\mathcal{D}x\left(t\right) \tag{4.36}$$

where $\mathcal{D}x\left(t\right)$ means the summation over all paths obeying the boundary conditions

$$x\left(t_a\right) = x_a \text{ and } x\left(t_b\right) = x_b \tag{4.37}$$

[a] In quantum mechanics, the propagator gives out complex results, and have therefore no interpretation as classic probability, they are usually called probability amplitude. It is the squared of the modulus of the propagator value that has an usual probabilistic interpretation.

Let us try right away the definition with the easiest example at our hand, the free particle $V = 0$. The probability amplitude between the endpoints $x\left(t_a\right) = x_a$ and $x\left(t_b\right) = x_b$ is given by the path integral

$$\begin{aligned} K\left(b,a\right) &:= \int \exp\left[\frac{i}{\hbar}\int\limits_{t_a}^{t_b}\frac{m}{2}\dot{x}^2 dt\right]\mathcal{D}x\left(t\right) \\ &= \lim_{\Delta t\to 0}\left(\frac{m}{2\pi i\Delta t}\right)^{N/2}\int\limits_{-\infty}^{\infty}...\int\limits_{-\infty}^{\infty}\exp\left[\frac{im}{2\hbar\Delta t}\sum_{i=0}^{N-1}\left(x_{i+1}-x_i\right)^2\right]dx_1...dx_{N-1} \end{aligned} \tag{4.38}$$

the second equality being the usual time-slicing approach to path integrals. Performing the Gaussian integrals in the same fashion as in (4.10), finally yields the result for the propagator of a free particle

$$K\left(b,a\right) = \sqrt{\frac{m}{2\pi i\hbar\left(t_b - t_a\right)}}\exp\left[\frac{im\left(x_b - x_a\right)^2}{2\hbar\left(t_b - t_a\right)}\right] \tag{4.39}$$

The final result bears striking resemblance with the transition probability for a Brownian particle with volatility $\propto m$. In effect let us apply the following substitution rules: $\hbar =$

$1, m = 1/\sigma^2, t \to -it$, then (4.39) becomes

$$\frac{1}{\sqrt{2\pi\sigma^2\left(t_b - t_a\right)}} \exp\left[-\frac{\left(x_b - x_a\right)^2}{2\sigma^2\left(t_b - t_a\right)}\right] \tag{4.40}$$

This is the well known result[FH12] that a propagator in imaginary time (Wick rotation) recovers the result we derived in the Wiener path integral theory. This is the reason why we referred to the functional in Feynman-Kac formula as a potential term, while the Wiener measure is related to the Kinetic term of a free particle with mass $1/\sigma^2$, both making a proxy Lagrangian. A deeper discussion of the resemblance of the Wiener measure to a Lagrangian can be found in [MC01] or [EBT99].

4.3.3 Connection between the diffusion equation and the Schrodinger equation

Starting with the Schrodinger equation as discussed in 4.3.1

$$i\hbar\frac{\partial\psi\left(x,t\right)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi\left(x,t\right)}{\partial x^2} + V\left(x,t\right)\psi\left(x,t\right) \equiv H\psi\left(x,t\right) \tag{4.41}$$

for an Hamiltonian operator H acting on wave function Ψ

$$H := \left[-\frac{\hbar^2}{2m}\frac{\partial^2}{\partial x^2} + V\left(x,t\right)\right] \tag{4.42}$$

Taken as a function of its final point b the propagator $K\left(b,a\right)$ satisfies (4.41)[FH12]. With the additional condition that $K\left(b,a\right) = 0$ for $t_b < t_a$ and $\lim\limits_{t_b \to t_a} K\left(b,a\right) = \delta\left(x_b - x_a\right)$, the propagator $K\left(b,a\right)$ is actually the Green function of (4.41).
Analogy between the Wiener integral/diffusion equation and Feynman integral/Schrodinger equation can be made even clearer by applying the same set of subsitution rules as before $\hbar = 1, m = 1/\sigma^2, t \to -it$ to a free particle $V = 0$, then

$$i\hbar\frac{\partial\psi\left(x,t\right)}{\partial t} = -\frac{\hbar^2}{2m}\frac{\partial^2\psi\left(x,t\right)}{\partial x^2} \Rightarrow \frac{\partial\psi\left(x,t\right)}{\partial t} = -\frac{\sigma^2}{2}\frac{\partial^2\psi\left(x,t\right)}{\partial x^2} \tag{4.43}$$

the Schrodinger equation becomes, at least formally, a diffusion equation.

4.4 Application to option pricing

The body of work inside the Econophysics field aimed to using path integral methods for option pricing can be traced back to seminal papers by John Dash[J.D88][J.D89]. The author motivates the recourse to path integrals by the prevalence of pricing models where diffusion equations arise, and the ease to rewrite them under path integral forms. Since as we saw all throughout this chapter, there is a close connection between diffusion-like PDE and path integrals. Thus it is rather easy to switch between the most convenient representations for one problem.To us, it is the many analogies between quantum mechanics probabilistic description of phenomenon and financial assets stochastic dynamics, whether expressed in the propagator form or in the Schrodinger equation, that constitute the most cogent argument for the use of path integrals methods in finance.

A non exhaustive overview of the available works targeting the use of path integrals in finance is

- Theoretical and introductions to the framework can be found in [Lin98],[EBT99],[Baa04].
- Exotic options pricing publications are discussed in [LLT11],[CCO14a],[CCO14b],[DLT10]
- Stochastic volatility models are studied in [Bel97],[BKS00]
- Computational aspects are discussed in [RCT02],[Mat00]

In the present thesis we are uniquely interested in the second matter: exotic option pricing. The path integrals methodology used to address such a challenge vary from problems to problems but we are going to give here of the outline, for a vanilla option

1. Write a SDE for the asset dynamics
2. Rewrite the SDE under path integral form
3. Solve the path integral to get the propagator
4. Use the propagator as a pricing kernel along with the payoff function

The general approach we follow for a more complex products remains however similar the difference between trivial vanilla product and exotic is the nature of the payoff. In the vanilla case it is merely a function of the final price and thus does not participate in the path integral while for exotic case the payoff is a functional and thus must be accounted for in the path integral Lagrangian term. An example of this approach will be given in chapter 5 while it will be shown in chapter 6 that for time-occupation derivatives it may be more sensible to start from the Feynman-Kac formula and solve the Bloch PDE.

Regarding our vanilla example, assume a stock S whose price obeys the following dynamic

$$dS_t = S_t(rdt + \sigma dW_t) \tag{4.44}$$

using Ito lemma on $f(x) = \ln(x)$ yields for the log price process $X_t := \ln(S_t/S_0)$

$$dX_t = \mu dt + \sigma dW_t \ , \ \left(\mu := r - \frac{\sigma^2}{2}\right) \tag{4.45}$$

Here the Lagrangian takes the following form

$$L\left(\dot{x},x,t\right)=\frac{\left(\dot{x}-\mu\right)^2}{2\sigma^2} \tag{4.46}$$

We previously derived the propagator for such a process in (4.16) where it was found to be

$$W\left(X_T,T|0,0\right)=\int\limits_{\mathcal{C}\{0,0;X_T,T\}} e^{-\int\limits_0^T L(\dot{x},x,t)dt} dW_x\left(t\right)=\frac{\exp\left[\frac{-(X_T-\mu t)^2}{2\sigma^2 T}\right]}{\sqrt{2\pi\sigma^2 T}} \tag{4.47}$$

We write a European call on this asset with the following parameters: maturity T, risk-free rate r, strike and initial price $K=S_0$, the payoff **function** is $\max\left(S_T-K,0\right)$. We can get the fair price C for this option by using the risk neutral formula and the propagator as a pricing kernel in the following manner

$$\begin{aligned}C&=e^{-rT}\mathbb{E}\left[\max\left(S_T-K,0\right)\right]\\&=e^{-rT}\int\limits_{-\infty}^{+\infty}\max\left(S_T-K,0\right)W\left(X_T,T|0,0\right)dX_T\\&=e^{-rT}\int\limits_{\ln K/S_0}^{+\infty}\left(S_T-K\right)W\left(X_T,T|0,0\right)dX_T\\&=e^{-rT}\left[S_0\int\limits_{\ln K/S_0}^{+\infty}e^{X_T}\frac{\exp\left[\frac{-(X_T-\mu t)^2}{2\sigma^2 T}\right]}{\sqrt{2\pi\sigma^2 T}}-K\int\limits_{\ln K/S_0}^{+\infty}\frac{\exp\left[\frac{-(X_T-\mu t)^2}{2\sigma^2 T}\right]}{\sqrt{2\pi\sigma^2 T}}dX_T\right]\end{aligned} \tag{4.48}$$

which after simplification can be rewritten

$$S_0\Phi\left(\frac{\ln\left(\frac{S_0}{K}\right)+T\left(\mu+\sigma^2\right)}{\sigma\sqrt{T}}\right)-e^{-rT}K\Phi\left(\frac{\ln\left(\frac{S_0}{K}\right)+\mu T}{\sigma\sqrt{T}}\right) \tag{4.49}$$

$\Phi\left(\cdot\right)$ being the normal distribution CDF.

The result derived just here is the same that could have been obtained from solving the Black-Scholes PDE, confirming the general validity of the path integral approach in the context of option pricing.

Chapter 5

Path Integral Pricing of Double Outside Barrier Asian Options

5.1 Introduction

The introductions chapters preceding this one have prepared the stage for the study of the exotic option pricing problem, by introducing the relevant tools and concepts. In this chapter we will study the pricing problem for a rather involved class of exotic options, Asian options with an added barrier restriction on an additional asset. As we hinted at in the introduction of this thesis there are different framework and toolbox that one can use when addressing the pricing of options derivatives, since there are many angles available to attack the option pricing problem: from rather abstract purely probabilistic consideration, or from formulation as partial differential equations[Pas11][Kwo08], then again by re-casting the problem in the framework of functional integration[J.D88][J.D89][Lin98][EBT99]. The powerful gateway acting as a pivotal tool to switch from and to those different representations of the same problem being the Feynman-Kac formula[Kac66], that permits to express the solution to a Bloch type partial differential equation (related to convection diffusion equation) as a path integral. The worked out solution will posses all characteristics of a transition probability density or propagator (e.g. its normalisation, its vanishing for negative time), which is a necessary element when considering risk-neutral pricing. In effect,the transition probability density of the system under consideration can broadly be regarded as a pricing kernel for the integration one solve when writing the expectation of a payoff.

When targeting derivatives so-called exotic options, the formulation of the pricing problem as a functional integral appears to be of particular appeal since we are interested in the particular path that will be followed by the price process. Path dependent options belong to the so-called class of exotics products, and exhibits a payoff structure that is connected to the entire history, the whole path, of the process. Hence, path integrals or functional integration is, at least in our opinion, a very fruitful and intuitive framework to work in. It is then only natural to see a growing body of literature dedicated to the application of path integrations techniques in finance. Although it is often referred to as Feynman path integrals in the literature[MNM02][FH12], it is then restricted to classical action, or analytically continued in imaginary time (Wick's rotation) to get rid of the oscillatory nature of the action in quantum mechanics. But then, one is actually considering Wiener functional integration[Wie21][Wie24]. Thus in this chapter, both terms "path integrals" and "Wiener integrals" are not meant to be semantically distinguished, and will be used interchangeably.

The specific type of exotic options that will be studied in this chapter, using path integrals of the Wiener type are termed Asian options[Hul08], and more precisely continuously monitored geometrically averaged Asian options. Since they propose payoffs that depend on the whole path of the risky asset process, they can effectively provide some protection against market manipulations close to the option maturity date. Using this process as our payoff asset, we will study different barrier condition imposed on a distinct but correlated stochastic process. Such a barrier condition is often called *outside barrier* in the financial literature[Zha95], and are rather less studied than the regular type (i.e. the payoff and barrier are imposed on the same asset). Barriers in derivatives are often used to limit exposure to risk and allow to lower the premium value; since a barrier conditions can only diminish the possible payoff of an option, it can not be worth more than a similar option without barrier condition. We will study the pricing problem when the barrier condition is of knock-out type, continuing the work in [DLT10] we will transform the system to an equivalent form and solve the pricing problem without partitioning paths. Then we will build complexity by imposing a double outside barrier condition and, to the best of our knowledge, derive an altogether completely original result for such a complex product.

5.2 Wiener's path integration

The Wiener path integrals theory was introduced to greater length in chapter 4, yet for ease of reading we will recall here the reader, briefly, of the original works of Norbert Wiener in defining path integrals. The so-called conditional Wiener measure[MC01], using the finite-dimensional approximations (time-slicing approach) $\epsilon = t/N$ is defined by

$$d_W x\left(\tau\right) := \exp\left[-\frac{1}{4D}\int\limits_0^t \dot{x}^2\left(\tau\right)d\tau\right]\prod_{\tau=0}^{t}\frac{dx\left(\tau\right)}{\sqrt{4\pi D d\tau}} \tag{5.1}$$

$$\prod_{\tau=0}^{t}\frac{dx\left(\tau\right)}{\sqrt{4\pi D d\tau}} := \frac{1}{\left(4\pi D\epsilon\right)^{(N+1)/2}}\int\limits_{-\infty}^{+\infty}dx_1\int\limits_{-\infty}^{+\infty}dx_2\cdots\int\limits_{-\infty}^{+\infty}dx_{N+1} \tag{5.2}$$

Then, the transition probability[1] $W\left(x_T|x_0\right)$ is given by the following conditional Wiener integral where $C\{x_0,0;x_T,T\}$ is the set of continuous but non differentiable paths with

[1] Also referred to as propagator, Green function or transition kernel.

fixed endpoints $x\left(0\right)=x_0$ and $x\left(T\right)=x_T$

$$W\left(x_T|x_0\right):=\int\limits_{C\{x_0,0;x_T,T\}}d_Wx\left(\tau\right) \tag{5.3}$$

It is worth pointing out here, that this transition density is also the solution to the diffusion equation. Directly relevant to this study on option pricing is the Wiener integral of functionals $F\left[x\left(\tau\right)\right]$ simply defined as

$$\int\limits_{C\{x_0,0;x_T,T\}}F\left[x\left(\tau\right)\right]d_Wx\left(\tau\right) \tag{5.4}$$

where the finite dimensional approximation is carried on with a Cauchy sequence of simple functions $\{F_n\}$ converging to F with respect to the norm

$$\|F_n-F\|=\int|F_n-F|d_Wx\left(\tau\right) \tag{5.5}$$

5.3 Average price put option with outside up-and-out barrier

5.3.1 System description

Working in the Black Scholes model [BS73], let S_X and S_Y be 2 correlated risky assets

$$\begin{aligned} S_{X,t}&=S_{X,0}\exp\left[\left(r-\frac{\sigma_X^2}{2}\right)t+\sigma_XW_{1,t}\right]\\ S_{Y,t}&=S_{Y,0}\exp\left[\left(r-\frac{\sigma_Y^2}{2}\right)t+\sigma_Y\left(\rho W_{1,t}+\sqrt{1-\rho^2}W_{2,t}\right)\right]\end{aligned} \tag{5.6}$$

where r is the risk-free rate, σ_X (resp. σ_Y) is the annualised volatility for the asset S_X (resp. S_Y), and ρ is the correlation between S_X and S_Y. We also define the following up-and-out barrier condition on $S_Y(t)$ at level B, while the payoff itself is written on the geometric average of $S_{X,t}$ with a strike level K. The payoff function can be written as such

$$H(S_{X,T},S_{Y,T})=\begin{cases}\left(K-S_{X,0}e^{\frac{1}{T}\int_0^t\ln\left(S_{X,s}\right)ds}\right)_+ & if S_{Y,t}<B,\forall t\in[0,T]\\ 0 & \text{otherwise}\end{cases}$$

Switching to log variables the two asset dynamics reads

$$\begin{aligned} X_t&:=\ln\left(\frac{S_{X,t}}{S_{X,0}}\right)=\mu_Xt+\sigma_XW_{1,t}\\ Y_t&:=\ln\left(\frac{S_{Y,t}}{S_{Y,0}}\right)=\mu_Yt+\sigma_Y\left(\rho W_{1,t}+\sqrt{1-\rho^2}W_{2,t}\right)\end{aligned} \tag{5.7}$$

where we introduce the following definitions

$$l_k := \ln\left(\frac{K}{S_{X,0}}\right), l_b := \ln\left(\frac{B}{S_{Y,0}}\right), \mu_X := \left(r - \frac{\sigma_X^2}{2}\right), \mu_Y := \left(r - \frac{\sigma_Y^2}{2}\right) \tag{5.8}$$

Introducing the continuously monitored geometric average of X between 0 and t, $\bar{X}_t$

$$\bar{X}_t := \int_0^t X_s ds \tag{5.9}$$

Now it is known that $\bar{X}_t$ is normally distributed with mean $\mu_{\bar{X}} t = \mu_X \frac{t}{2}$ and variance $\sigma_{\bar{X}}^2 t = \sigma_X^2 \frac{t}{3}$. Interested reader can find a brief derivation in appendix B. The correlation between $\bar{X}$ and Y is found to be $\tilde{\rho} = \rho\frac{\sqrt{3}}{2}$. Thus we can rewrite the system (5.7) where we use equality in law to justify the substitution of stochastically equivalent processes $X_t \Rightarrow \bar{X}_t$

$$\begin{aligned} \bar{X}_t &= \mu_{\bar{X}} t + \sigma_{\bar{X}} W_{1,t} \\ Y_t &= \mu_Y t + \sigma_Y \left(\tilde{\rho} W_{1,t} + \sqrt{1-\tilde{\rho}^2} W_{2,t}\right) \end{aligned} \tag{5.10}$$

5.3.2 System Propagator

The theory of path integration as developed by Norbert Wiener, the functional used in order to weigh respective paths is, at least formally, equivalent to the classical action of the system under consideration, where the volatility σ assumes in that case a role that is analogous to the mass of a particle under consideration. Since we will assume the Black Scholes assumptions, σ is assumed to be constant.

The Lagrangian for the system (5.10) under Ito interpretation is defined[Lin98] as

$$\mathcal{L} = \frac{1}{2}\sum_{\eta=\nu=1}^{2} \mathbf{G}_{\eta,\nu}(\mathbf{X},t)\left[\dot{\mathbf{X}}_t^\eta - \mathbf{A}(\mathbf{X},t)^\eta\right]\left[\dot{\mathbf{X}}_t^\nu - \mathbf{A}(\mathbf{X},t)^\nu\right] \tag{5.11}$$

where

$$\mathbf{X}_t := \begin{bmatrix} \bar{X}_t \\ \bar{Y}_t \end{bmatrix}, \mathbf{A} := \begin{bmatrix} \mu_{\bar{X}} \\ \mu_Y \end{bmatrix}, \mathbf{G} := \begin{bmatrix} \sigma_{\bar{X}}^2 & \sigma_{\bar{X}}\sigma_Y\rho \\ \sigma_{\bar{X}}\sigma_Y\rho & \sigma_Y^2 \end{bmatrix}^{-1}$$

Integrating (5.11) yields the classical action

$$A_T = \int_0^T \mathcal{L}dt = A_0 + \alpha T - \beta\bar{X}_T - \gamma Y_T \tag{5.12}$$

with

$$
\begin{aligned}
A_0 &:= \int_0^T \frac{1}{2(1-\tilde{\rho}^2)} \left(\frac{\dot{\bar{X}}_t^2}{\sigma_{\bar{X}}^2} + \frac{\dot{Y}_t^2}{\sigma_Y^2} - \frac{2\tilde{\rho}\dot{\bar{X}}_t \dot{Y}_t}{\sigma_{\bar{X}}\sigma_Y} \right) dt \\
\alpha &:= \frac{1}{2(1-\tilde{\rho}^2)} \left(\frac{\mu_{\bar{X}}^2}{\sigma_{\bar{X}}^2} + \frac{\mu_Y^2}{\sigma_Y^2} - 2\tilde{\rho}\frac{\mu_{\bar{X}}\mu_Y}{\sigma_{\bar{X}}\sigma_Y} \right) \\
\beta &:= \frac{1}{\sigma_{\bar{X}}^2(1-\tilde{\rho}^2)} \left(\mu_{\bar{X}} - \frac{\tilde{\rho}\sigma_{\bar{X}}\mu_Y}{\sigma_Y} \right) \\
\gamma &:= \frac{1}{\sigma_Y^2\,(1-\tilde{\rho}^2)} \left(\mu_Y - \frac{\tilde{\rho}\sigma_Y\mu_{\bar{X}}}{\sigma_{\bar{X}}} \right)
\end{aligned}
\tag{5.13}
$$

Terms that are linear in $\bar{X}_t$ and Y_t are pulled out of the free action A_0 for convenience, and we will see later in (5.18) that a straightforward algebraic manipulation pulls them back into the propagator. The correlation between $\bar{X}_t$ and Y_t makes the problem of finding the classical action more delicate than otherwise. Thus we employ the following transformation

$$
\begin{aligned}
\tilde{X} &\leftarrow \bar{X} - \tilde{\rho}\frac{Y\sigma_{\bar{X}}}{\sigma_Y} \\
\tilde{Y} &\leftarrow Y
\end{aligned}
\tag{5.14}
$$

to effectively uncouple A_0 terms, and factorize propagators. Using (5.1) and (5.3) after transformation, leads to the following path integrals for the barrier-restricted propagator $W\left(\tilde{X}_T, \tilde{Y}_T | X_0 = Y_0 = 0\right)$

$$
\begin{aligned}
W\left(\tilde{X}_T, \tilde{Y}_T | 0, 0\right) = \int\limits_{C\{0;x_T,T\}} & e^{-\int_0^T \frac{\dot{\tilde{X}}_t^2}{2\sigma_{\bar{X}}^2(1-\tilde{\rho}^2)}dt} d_W x(\tau) \\
& \times \int\limits_{\substack{C\{0;y_T,T\} \\ Y(t)<l_b,\forall t\in[0,T]}} e^{-\int_0^T \frac{\dot{Y}_t^2}{2\sigma_Y^2}dt} d_W y(\tau)
\end{aligned}
\tag{5.15}
$$

Since action terms are at most quadratic in each variable, path integration can be done exactly[MC01]. The computation of the restricted $[-\infty; l_b]$ path integral for Y is handled through reflection method[Goo81]. After reverting back to our original variables $\bar{X}_t$ and Y_t, the propagator takes the following form

$$
\begin{aligned}
W\left(\bar{X}_T, Y_T | 0, 0\right) = \frac{1}{2\pi\sigma_{\bar{X}}\sigma_Y T\,(1-\tilde{\rho}^2)} & \exp\left[-\frac{\left(\bar{X}_T - \tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}Y_T\right)^2}{2\sigma_{\bar{X}}^2\,(1-\tilde{\rho}^2)\,T} \right] \\
\times \left[\exp\left(-\frac{(Y_T)^2}{2\sigma_Y^2 T} \right) - \exp\left(-\frac{(2l_b - Y_T)^2}{2\sigma_Y^2 T} \right) \right] &
\end{aligned}
\tag{5.16}
$$

where the last difference of normal density term is characteristic of barrier discontinuity resolution through reflection, as can be seen on Fig. 5.1. If one is interested in measuring the Brownian paths that at any time in their lifetime, at one point crossed the barrier, one is also measuring the Brownian paths that never crossed the barrier. The problem of looking at the probability that a Brownian particle moves from X_0 to X_T without crossing the barrier is similar to the physic problem of the infinite potential. The problem is solved by subtracting to the set of all paths from X_0 to X_T (the term $e^{-Y_T^2}$) the set of all paths reflected around a first barrier crossing (the term $e^{(2l_b-Y_T)^2}$). Interested readers can find in appendix A a derivation of a similar result based on an original change of measure argument.

5.3.3 Option fair value

Under efficient market assumption the fair price P of our newly written option is given by the discounted expectation (w.r.t risk neutral measure) of the payoff functional, thus leading to the following integral of the form (5.4)

$$P = e^{-rT} \int_{-\infty}^{l_k} \int_{-\infty}^{l_b} e^{-\alpha T + \beta \bar{X}_T + \gamma Y_T} \left(K - S_{X,0} e^{\bar{X}_T} \right) W \left(\bar{X}_T, Y_T | 0, 0 \right) d\bar{X}_T dY_T \tag{5.17}$$

Combining (5.16) and (5.17) and the following identity to bring back linear terms in the bivariate Gaussian density $\psi_x(\mu, \Omega)$ with mean vector μ and covariance matrix Ω

$$\exp\left[w'x\right] \psi_x(\mu, \Omega) = \exp\left[w'\mu + \frac{1}{2} w' \Omega w\right] \psi_x(\mu + \Omega w, \Omega) \tag{5.18}$$

yields the following result for the single outside barrier Asian put option

$$\begin{aligned} P = e^{-T\left(r+\alpha-\frac{\sigma_{\bar{X}}^2}{2}\beta-\frac{\sigma_Y^2}{2}\gamma-\tilde{\rho}\sigma_{\bar{X}}\sigma_Y\beta\gamma\right)} \Bigg[& K\left(\Psi_1 - \left(\frac{B}{S_{Y,0}}\right)^{\frac{-\ln\left(\frac{B}{S_{Y,0}}\right)+\sigma_Y^2+2\tilde{\rho}\sigma_Y\sigma_{\bar{X}}\beta}{2\sigma_Y^2}} \Psi_2\right) \\ & - S_{X,0} e^{T\left(\frac{\sigma_{\bar{X}}^2}{2}+\tilde{\rho}\sigma_{\bar{X}}\sigma_Y\gamma\right)} \left(\Psi_3 - \left(\frac{B}{S_{Y,0}}\right)^{\frac{-\ln\left(\frac{B}{S_{Y,0}}\right)+\sigma_Y^2+2\tilde{\rho}\sigma_Y\sigma_{\bar{X}}\beta}{2\sigma_Y^2}} \Psi_4\right) \Bigg] \end{aligned} \tag{5.19}$$

where Ψ_i is the bivariate standard cumulative distribution function

$$\Psi_i = \frac{1}{2\pi\sqrt{(1-\rho^2)}} \int_{-\infty}^{a_i} \int_{-\infty}^{b_i} \exp\left[\frac{-1}{2(1-\rho^2)}\left(x^2 + y^2 - 2\rho x y\right)\right] dy dx \tag{5.20}$$

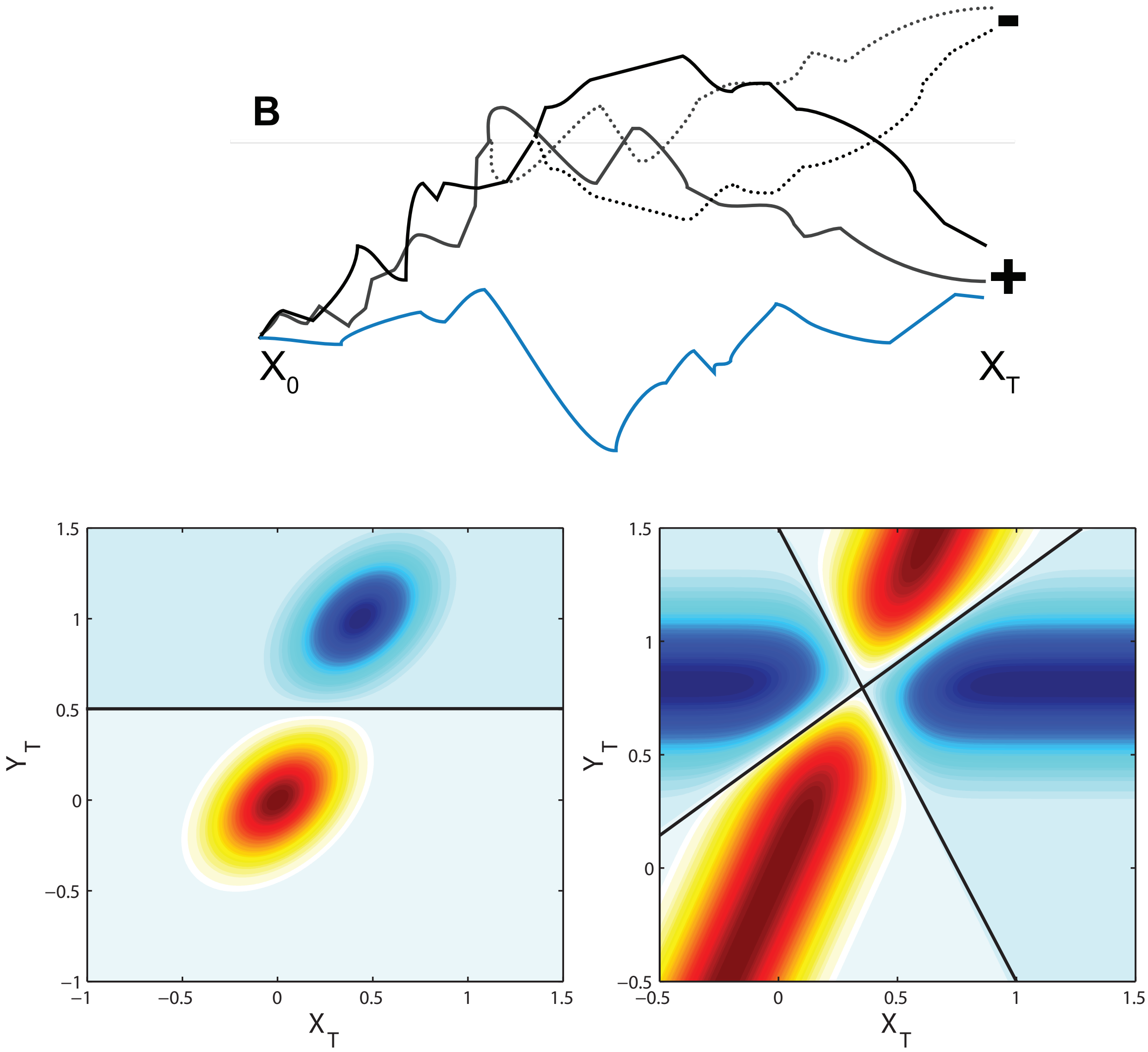


Figure 5.1: Top panel: Illustration of the correspondence between reflected paths and crossing paths. Only paths that crossed the barrier at one point can be put in a 1 to 1 pair with a path reflected around the barrier at the first crossing time (Black and grey paths); while the blue path does not have a reflected path and will not be discarded.
Bottom panel: Visual representation of the amplitude of barrier restricted propagator as a function of terminal price X_T, Y_T. Left panel: Single outside barrier placed at level $B = 150$, a line has been plotted to indicate where the amplitude falls to 0. As expected it coincides with the barrier absorption. Right panel: Double outside barrier placed at levels $B^+ = 140$ and $B^- = 60$, as previously the line shows where the amplitude falls to 0. Lines indicate approximately where the barrier absorption takes place.

evaluated at

$$a_1 := \frac{l_k - \left(\beta\sigma_{\bar{X}}^2 T + \gamma\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right)}{\sigma_{\bar{X}}\sqrt{T}} \qquad b_1 := \frac{l_b - \left(\gamma\sigma_Y^2 T + \beta\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right)}{\sigma_Y\sqrt{T}}$$

$$a_2 := \frac{l_k - \left[\beta\sigma_{\bar{X}}^2 T + \left(\gamma + \frac{l_b}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right]}{\sigma_{\bar{X}}\sqrt{T}} \qquad b_2 := \frac{l_b - \left(l_b + \gamma\sigma_Y^2 T + \beta\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right)}{\sigma_Y\sqrt{T}}$$

$$a_3 := \frac{l_k - \left[(\beta+1)\sigma_{\bar{X}}^2 T + \gamma\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right]}{\sigma_{\bar{X}}\sqrt{T}} \qquad b_3 := \frac{l_b - \left[\gamma\sigma_Y^2 T + (\beta+1)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right]}{\sigma_Y\sqrt{T}}$$

$$a_4 := \frac{l_k - \left[(\beta+1)\sigma_{\bar{X}}^2 T + \left(\gamma + \frac{l_b}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right]}{\sigma_{\bar{X}}\sqrt{T}} \qquad b_4 := \frac{l_b - \left[l_b + \gamma\sigma_Y^2 T + (\beta+1)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T\right]}{\sigma_Y\sqrt{T}}$$

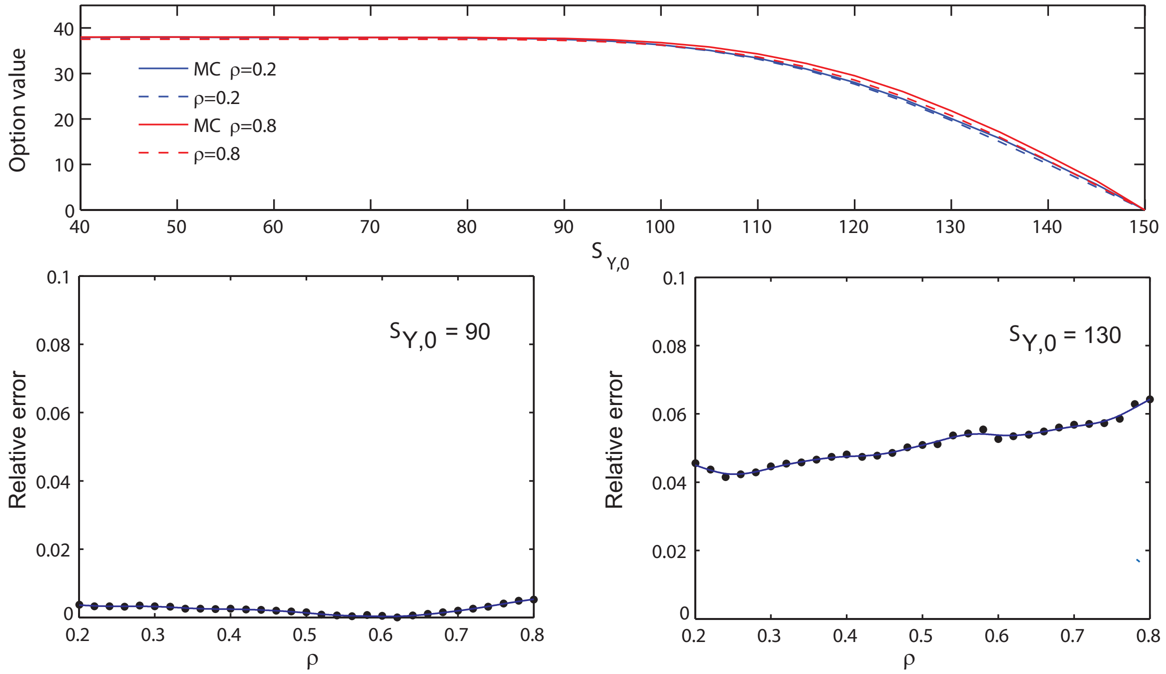


Figure 5.2: Top panel: Option value P with respect to initial control process value $S_{Y,0}$, with barrier level $B = 150$. **Bottom panel**: Relative error to Monte Carlo price with respect to correlation coefficient $\rho \in [0.2, 0.8]$ for $S_{Y,0} = 90$ and $S_{Y,0} = 130$. Both panels use arbitrary currency unit in which the asset is originally quoted.

5.3.4 Results

We benchmarked the accuracy of our analytic formula against a Monte Carlo simulation using 2^{16} generated paths, and 2^{10} timesteps. Results are plotted in Fig.5.2 for the following set of annualized parameters in their own usual units: risk-free rate $r = 0.03$, volatility is $\sigma_X = \sigma_Y = .2$ the strike level is set at $K = 100$, the barrier is at $B = 150$, the maturity is $T = 1$, the initial price for the risky asset X is $S_{X,0} = 60$. Results are plotted in Fig.5.2 for two values of the correlation parameter ρ: weakly correlated $\rho = 0.2$ and highly correlated $\rho = 0.8$.

We have the positive result that as correlation increase the maximum relative error between the Monte Carlo simulation and our result decreases in a quasi-monotonically fashion, with the precise value depending on the barrier proximity to $S_{Y,0}$. This result is an improvement over the formula available in [DLT10] that saw major degradation of result accuracy as correlation increased. Since in our work we did not follow the usual method of separating the sets of paths with respect to their average value, we also do not meet the caveat of the barrier canceling the wrong set of paths as explained in their paper. We also reproduce the expected result that when the probability of knocking out the barrier is negligible, i.e. $S_{Y,0} \ll B$, Monte Carlo and analytical outputs coincide almost perfectly.

Building on those satisfying results, we can move on to a more involved problem that to best of our knowledge has not been explored so far, the pricing of Asian option with double outside knock-out barrier.

5.4 Average price put option with outside double knock-out barrier

5.4.1 System description

We will work here with the two dimensional assets system as described in (5.10) where we now impose a double knock-out barrier condition on $S_{Y,t}$ at level B^+ and B^-. Now the payoff takes the following form

$$H(S_{X,T}, S_{Y,T}) = \begin{cases} \left(K - S_{X,0} e^{\frac{1}{T}\int_0^t \ln\left(S_{X,s}\right) ds}\right)_+ & if B^- < S_{Y,t} < B^+, \forall t \in [0,T] \\ 0 & \text{otherwise} \end{cases}$$

and much in the trail of the first section we introduce the log-barrier levels

$$l_{b^+} := \ln\left(\frac{B^+}{S_{Y,0}}\right), l_{b^-} := \ln\left(\frac{B^-}{S_{Y,0}}\right), l_b := \ln\left(\frac{B^+}{B^-}\right) \tag{5.21}$$

5.4.2 System Propagator

Using the same transformation strategy to uncorrelate both variables as the one described in section 5.3.2, the doubly restricted path integral now takes the following form

$$
\begin{aligned}
W\left(\tilde{X}_T, \tilde{Y}_T|0,0\right) = & \int\limits_{C\{0;X_T,T\}} e^{-\int_0^T \frac{\dot{\tilde{X}}_t^2}{2\sigma_{\bar{X}}^2(1-\tilde{\rho}^2)}dt} d_W x\left(\tau\right) \\
& \times \int\limits_{\substack{C\{0;Y_T,T\} \\ l_{b^-}<Y(t)<l_{b^+},\forall t\in[0,T]}} e^{-\int_0^T \frac{\dot{Y}_t^2}{2\sigma_Y^2}dt} d_W y\left(\tau\right)
\end{aligned}
\tag{5.22}
$$

As before the path integral for X_t is one of a free particle, and so it shall yield a propagator that is simply a Gaussian distribution. Now Y_t presents a double barrier, a situation that is analog to one of a particle confined in an infinite square well. The strategy to handle in the context of path integrals a double infinite potential corresponding here to our double barriers situation, was put forward in [Goo81], and the propagator is found from repeated use of reflection principle

$$
\begin{aligned}
W\left(\bar{X}_T, Y_T|0,0\right) = & \frac{1}{2\pi\sigma_{\bar{X}}\sigma_Y T\left(1-\tilde{\rho}^2\right)} \exp\left[-\frac{\left(\bar{X}_T - \tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}Y_T\right)^2}{2\sigma_{\bar{X}}^2\left(1-\tilde{\rho}^2\right)T}\right] \\
& \sum_{n=-\infty}^{+\infty}\left[\exp\left(-\frac{(Y_T-2nl_b)^2}{2\sigma_Y^2 T}\right) - \exp\left(-\frac{(Y_T-2(l_{b^-}+nl_b))^2}{2\sigma_Y^2 T}\right)\right]
\end{aligned}
\tag{5.23}
$$

Let us note here that the second density term in the infinite sum determines the magnitude of the un-canceled terms when truncating said sum. This is especially important for assets that are about to knock out. If $S_{Y,0} = B^-$ then the sum vanishes completely as $l_{b^-} = 0$, which is not the case if the asset knocks out the lower barrier B^-. A similar series could be written where the last exponential term in (5.23) is changed $l_{b^-} \to l_{b^+}$ which would lead to the opposite situation.

5.4.3 Option fair value

With the greeks' shorthands defined in (5.13), the fair price P of our newly written option is given by

$$
P = e^{-rT}\int_{-\infty}^{l_k}\int_{l_b^-}^{l_b^+} e^{-\alpha T+\beta\bar{X}_T+\gamma Y_T}\left(K - S_{X,0}e^{\bar{X}_T}\right) W\left(\bar{X}_T, Y_T|0,0\right) d\bar{X}_T dY_T \tag{5.24}
$$

Since the series converge absolutely, we are allowed to interchange summation and integration, and fall back to a path integral similar to (5.19). Define the parameterized solution to

(5.24)

$$P(\Lambda) := \exp\left[-T\left(r+\alpha-\frac{\sigma_{\bar{X}}^2}{2}\beta-\frac{\sigma_Y^2}{2}\gamma-\tilde{\rho}\sigma_{\bar{X}}\sigma_Y\beta\gamma\right)\right]$$

$$\times\sum_{n=-\infty}^{+\infty} Ke^{2nl_b\left(\tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}\beta+\frac{1}{2}-\frac{nl_b}{\sigma_Y^2 T}\right)}\left(\Psi_1(\Lambda)-e^{\Lambda(2\tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}\beta+1+2\frac{2nl_b-\Lambda}{\sigma_Y^2 T})}\Psi_2(\Lambda)\right) \tag{5.25}$$

$$-S_{X,0}e^{2nl_b\left(\tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}(\beta+1)+\frac{1}{2}-\frac{nl_b}{\sigma_Y^2 T}\right)}\left(\Psi_3(\Lambda)-e^{\Lambda(2\tilde{\rho}\frac{\sigma_{\bar{X}}}{\sigma_Y}(\beta+1)+1+2\frac{2nl_b-\Lambda}{\sigma_Y^2 T})}\Psi_4(\Lambda)\right)$$

where $\Psi_i(\Lambda)$ is the CDF of a bivariate normal distribution with unit variance and zero mean evaluated over the infinite strip parametrized by $\Lambda \in \{l_{b^-}, l_{b^+}\}$

$$\Psi_i(\Lambda) = \frac{1}{2\pi\sqrt{1-\tilde{\rho}^2}}\int_{-\infty}^{\frac{l_k-a_i}{\sigma_{\bar{X}}\sqrt{T}}}\int_{\frac{l_b^- -b_i}{\sigma_Y\sqrt{T}}}^{\frac{l_b^+ -b_i}{\sigma_Y\sqrt{T}}}\exp\left[\frac{-1}{2(1-\tilde{\rho}^2)}\left(x^2+y^2-2\tilde{\rho}xy\right)\right]dxdy \tag{5.26}$$

$$a_1 := \beta\sigma_{\bar{X}}^2 T+\left(\gamma+\tfrac{2nl_b}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$
$$b_1 := \left(\gamma+\tfrac{2nl_b}{\sigma_Y^2 T}\right)\sigma_Y^2 T+\beta\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$

$$a_2 := \beta\sigma_{\bar{X}}^2 T+\left(\gamma+\tfrac{2nl_b+2\Lambda}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$
$$b_2 := \left(\gamma+\tfrac{2nl_b+2\Lambda}{\sigma_Y^2 T}\right)\sigma_Y^2 T+\beta\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$

$$a_3 := (\beta+1)\sigma_{\bar{X}}^2 T+\left(\gamma+\tfrac{2nl_b}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$
$$b_3 := \left(\gamma+\tfrac{2nl_b}{\sigma_Y^2 T}\right)\sigma_Y^2 T+(\beta+1)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$

$$a_4 := (\beta+1)\sigma_{\bar{X}}^2 T+\left(\gamma+\tfrac{2nl_b+2\Lambda}{\sigma_Y^2 T}\right)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$
$$b_4 := \left(\gamma+\tfrac{2nl_b+2\Lambda}{\sigma_Y^2 T}\right)\sigma_Y^2 T+(\beta+1)\tilde{\rho}\sigma_{\bar{X}}\sigma_Y T$$

then the option no-arbitrage value is approximated by a sigmoid-shaped interpolation between $P(l_{b^-})$ and $P(l_{b^+})$ where we weigh each contribution based on the knock-out barrier that is closest to $S_{Y,0}$. The motivation behind this interpolation is to compensate the uncanceled terms in the infinite series (5.23) when $S_{Y,0} \approxeq B^+$ or $S_{Y,0} \approxeq B^-$.

$$\tilde{P} \approx \frac{P\left(l_{b^-}\right)}{1+\exp\left[-\alpha\frac{\left(S_{Y,0}-B^-\right)}{B^+-B^-}+\frac{\alpha}{2}\right]} + \left(1-\frac{1}{1+\exp\left[-\alpha\frac{\left(S_{Y,0}-B^-\right)}{B^+-B^-}+\frac{\alpha}{2}\right]}\right) P\left(l_{b^+}\right) \tag{5.27}$$

The α parameter is used in casting $[B^-; B^+]$ into $[-\alpha; +\alpha]$ yielding the well known S-shaped logistic curve, usually $\alpha = 5$ is adequate enough in retrieving said curve. Fig.5.3 shows the very basic logistic curve we used.

5.4.4 Results

We tested the accuracy of our analytical result against a Monte-Carlo simulation, generating 2^{16} paths for the 2 risky assets, and 2^{10} timesteps. The set of parameters is the same as found in 5.3.4, where we set the double barrier levels to $B^- = 60, B^+ = 140$.

The results derived from our formula are found to be relatively close to the one delivered by the Monte-Carlo simulation. Relative error is still found to be $\ll 0.1$ for a wide range of $S_{Y,0}$. However as seen from empirical evidence we can not longer claim that relative error is monotonically decreasing with respect to increased correlation ρ. That is, unless we disregard assets system about to knock-out upon being signed, i.e. $S_{Y,0} \approxeq B^+$ or $S_{Y,0} \approxeq B^-$, as seen on figure 5.4 for $|S_{Y,0} - B^+| = 10$. Explanation for this new behavior can be found in the impossibility to now exactly cancel the contribution from the infinitely many image sources and sinks. The infinite sum do converge, however when implementing said sum, we have to truncate after a reasonable number of terms, which lead to uncanceled residuals. Those residuals grow in magnitude as $S_{Y,0} \approxeq B^+$ or $S_{Y,0} \approxeq B^-$.

The smooth interpolation in (5.27) helps to mitigate such residuals impact regardless of the initial closest discontinuity, but can not remove it completely.

5.5 Conclusion

In this study we have derived two formula for the pricing of simple and double outside knock-out barrier options with Asian type payoff. By employing a transformation strategy we derived a stochastically equivalent system. We built a propagator for this system and did the functional integration without using the usual trick of partitioning the paths. Doing so, we reduced the error bound to canceling the wrong set of paths when correlation increase. We saw significant improvement over the hardships faced in previous study. Then we showed that the same steps could be taken to challenge the double outside barrier problem.

Future study should be aimed at a more detailed theoretical explanation for the different error profiles as seen in Fig. 5.4, then improve the relative error for options about to knock-

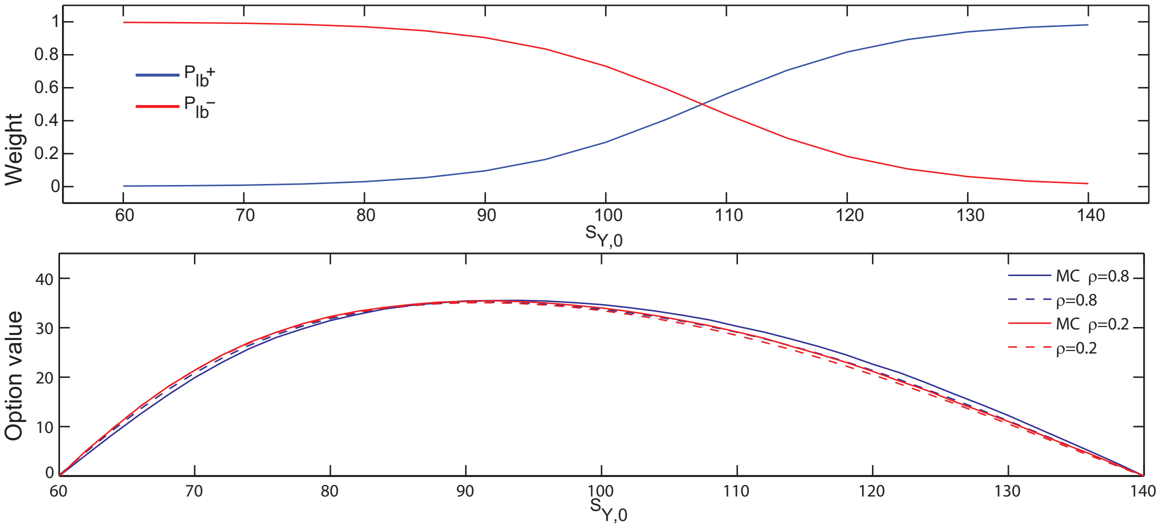


Figure 5.3: Top panel: Logistic curve used to weigh the contribution of $P\left(l_{b^+}\right)$ and $P\left(l_{b^-}\right)$ depending on the proximity to the up or down barrier. **Bottom panel**: Option fair price as given by a Monte Carlo simulation, versus the analytical result for $\tilde{P}$, with respect to the initial price for the barrier asset $S_{Y,0}$. The fair price is computed for weak correlation ($\rho = 0.2$), and highly correlated assets ($\rho = 0.8$).

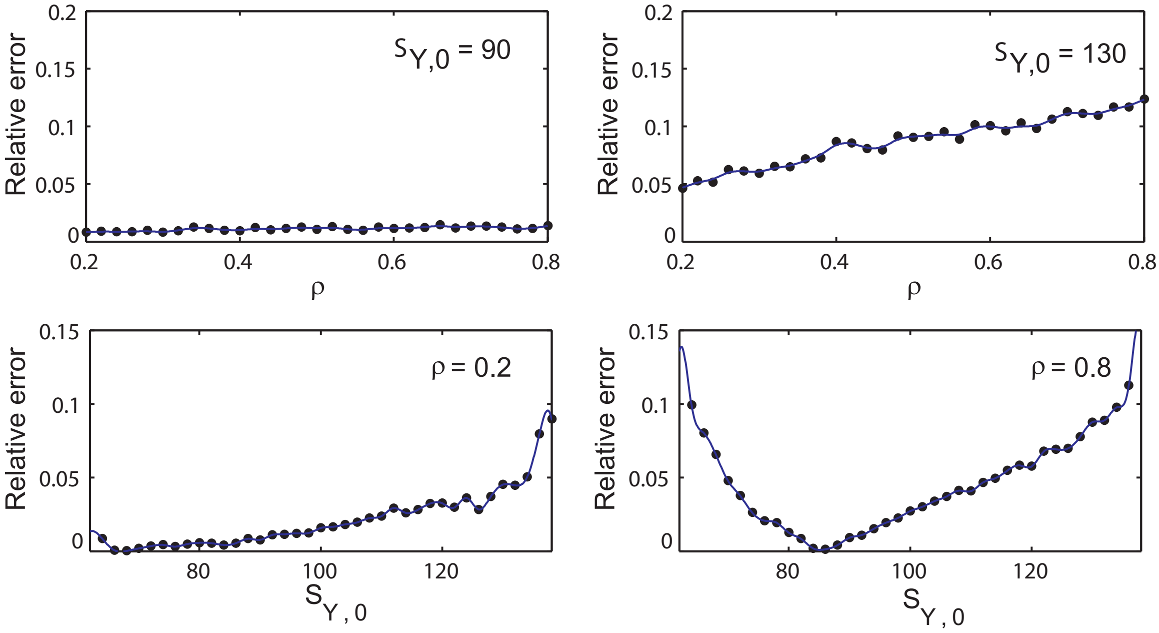


Figure 5.4: Top panel: Relative error between prices obtained through Monte-Carlo simulation and analytical result in (5.27) with respect to increasing correlation $\rho \in [0.2, 0.8]$.**Bottom panel**: Same relative error now with respect to increasing initial price $S_{Y,0} \in [60, 140]$.For each panel, five terms were used in the approximation of the infinite sum of (5.23).

out upon writing.

Chapter 6

Path Integral Pricing Of Wasabi Options

6.1 Introduction

In this chapter we will use the path integral to study the problem of a subclass of exotic products, namely the "cumulative Parisian option". The previous chapter used path integrals technique for a class of exotic products whose complexity is connected to so-called "topological constraint". The exotic product in the current chapter has a payoff structure that is connected to the average amount of time a risky asset price process spends in a prescribed domain. For this reason this class of exotics is often termed, "occupation time derivative". The model of the market used in this study is as was introduced in chapter 3 the Black-Scholes model[BS73]. The application of path integral techniques to the study of occupation time derivatives has to the best of our knowledge, so far been rather sparse. By using the Feynman-Kac formula we will derive a closed form for the propagator and use it to derive the option fair price. After comparison with a Monte Carlo simulation to ensure the accuracy of our results we will derive an approximation formula for an original product. The accuracy of our approximation will then be challenged against a Monte Carlo simulation.

The structure of this chapter is the following. In Section 6.2 we will study the case of the cumulative Parisian. In Section 6.3 the new "Wasabi option" is introduced and studied. Finally we draw our conclusion and present some perspectives in Section 6.4.

6.2 Up-and-in Parisian option

The cumulative Parisian option [CJPY97] is an exotic derivative, and yet more specifically an occupation time derivative. It allows to restrict the condition under which an option yields a profit. In the cumulative knock-in Parisian case, the option is worthless unless the option spends a certain ratio of the time to maturity over a specified constant level. We will hereafter refer to this ratio of time as "Parisian time" and the level of excursion as "Parisian level". It generalizes the so-called barrier option which nullifies the contract once the barrier level is touched. Therefore, one can see that a Barrier option is nothing more than a Parisian option where the Parisian time → 0. The vanilla option, is also a generalization where the Parisian time tends to the time to maturity.

6.2.1 System description

Let S_X be a risky asset, which under the Black-Scholes assumptions takes the following form

$$S_{X_t} = S_{X_0} \exp\left[\left(r - \frac{\sigma^2}{2}\right) t + \sigma W_t\right] \tag{6.1}$$

where r is the risk-free rate, σ is the annualized volatility for S_X, and W_t a Wiener process. The up-and-in Parisian barrier condition on S_X is set at level B and Parisian time d. The payoff is at strike level K. The Parisian up-and-in call option payoff function takes the following form

$$\left[S_{X_T} - K\right]_+ \mathbb{1}\left(\int\limits_0^T \Theta\left(S_{X_t} - B\right) dt > d\right) \tag{6.2}$$

Passing to log variables and under the Ito interpretation the asset is rewritten

$$X_t := \ln\left(\frac{S_{X_t}}{S_{X_0}}\right) = \mu t + \sigma W_t \tag{6.3}$$

where we introduce the following definitions

$$k := \ln\left(\frac{K}{S_{X_0}}\right), b := \ln\left(\frac{B}{S_{X_0}}\right), \mu := \left(r - \frac{\sigma^2}{2}\right) \tag{6.4}$$

6.2.2 Parisian propagator

In this chapter we will be concerned with the following path integral for a functional of exponential form and additive

$$W_B\left(x_T, T|x_0, 0\right) = \int\limits_{C\{x_0,0;x_T,T\}} e^{-\int\limits_0^T V(x_t)dt} d_W x\left(t\right) \tag{6.5}$$

with $V(x) \geq 0$. This equation defines a path integral whose solution $W_B\left(x_T, T|x_0, 0\right)$ admits an equivalent PDE form, the so-called *Bloch equation* [MC01]

$$\left[\frac{\partial}{\partial T} - D\frac{\partial^2}{\partial x_T^2} + V\left(x_T, T\right)\right] W_B\left(x_T, T|x_0, 0\right) = \delta\left(T\right) \delta\left(x_T - x_0\right) \tag{6.6}$$

That the solution to the path integral (6.5) is also the fundamental solution to (6.6) is basically the content of the celebrated Feynman-Kac theorem [Kac49].

Define the propagator $W_P\left(X_T,T,\lambda|X_0,0\right)$ for a driftless risky asset X_t that spends λ unit of time over the constant level b

$$\begin{aligned} W_P\left(X_T,T,\lambda|X_0,0\right) &:= \int\limits_{\substack{X_0\to X_T \\ \int_0^T \Theta(X_t-b)\,dt=\lambda}} d_W x\left(t\right) \\ &= \int\limits_{X_0\to X_T} \delta\left(\int\limits_0^T \Theta\left(X_t-b\right)dt-\lambda\right) d_W x\left(t\right) \\ &= \int\limits_{X_0\to X_T} \frac{1}{2i\pi}\int\limits_{\epsilon-i\infty}^{\epsilon+i\infty} e^{p\left(\lambda-\int_0^T \Theta(X_t-b)\,dt\right)}\,dp\,d_W x\left(t\right) \\ &= \frac{1}{2i\pi}\int\limits_{\epsilon-i\infty}^{\epsilon+i\infty} e^{p\lambda}\Big[\int\limits_{X_0\to X_T} e^{-p\int_0^T \Theta(X_t-b)\,dt}\,d_W x\left(t\right)\Big]\,dp \end{aligned} \tag{6.7}$$

As can be seen in the last line, the derivation of the Parisian propagator leads to the description of a particle going through a finite potential of height p. Denoting, for brevity sake, by W the Laplace transform of the transition density in the last line, and since by design $X_0 = 0$. We have, based on the Feynman-Kac formula, the following equivalent PDE form with $p > 0$, $b > 0$ and $\Theta\left(\cdot\right)$ being the step function

$$\frac{\partial W}{\partial T} = \frac{\sigma^2}{2}\frac{\partial^2 W}{\partial X_T^2} - p\Theta\left(X_T-b\right)W \tag{6.8}$$

Taking $\widehat{W}$ to be the Laplace transform of W with respect to T, the time to maturity

$$\left(s+p\Theta\left(X_T-b\right)\right)\widehat{W} - \frac{\sigma^2}{2}\widehat{W}'' = \delta\left(X_T\right) \tag{6.9}$$

splitting the real line in 3 regions: $R_1 := X_T < 0, R_2 := 0 < X_T < b, R_3 := X_T > b$, the bounded (W vanishing at infinity) continuous solutions in R_1, R_2, R_3 are respectively

$$\begin{aligned} &Ae^{\frac{\sqrt{2s}}{\sigma}X_T} \\ &Be^{\frac{\sqrt{2s}}{\sigma}X_T} + Ce^{-\frac{\sqrt{2s}}{\sigma}X_T} \\ &De^{-\frac{\sqrt{2(p+s)}}{\sigma}X_T} \end{aligned} \tag{6.10}$$

Requiring the continuity of the solution at $X_T = 0$, $X_T = b$ and of its derivative at $X_T = b$, and based on discontinuity magnitude at $X_T = 0$, we have the following conditions to enforce

$$\begin{aligned} A &= B + C \\ Be^{\frac{\sqrt{2s}}{\sigma}b} + Ce^{-\frac{\sqrt{2s}}{\sigma}b} &= De^{-\frac{\sqrt{2(p+s)}}{\sigma}b} \\ A + C - B &= \sigma\sqrt{\frac{2}{s}} \\ \sqrt{s}Be^{\frac{\sqrt{2s}}{\sigma}b} - \sqrt{s}Ce^{-\frac{\sqrt{2s}}{\sigma}b} &= -\sqrt{p+s}De^{-\frac{\sqrt{2(p+s)}}{\sigma}b} \end{aligned} \tag{6.11}$$

Yielding the results for the constants

$$\begin{aligned} A &= \frac{\left(2\sqrt{s\left(p+s\right)} - p - 2s\right)e^{-\frac{2b\sqrt{2s}}{\sigma}} + p}{\sigma p\sqrt{2s}} \\ B &= -\frac{\left(-2\sqrt{s\left(p+s\right)} + p + 2s\right)e^{-\frac{2b\sqrt{2s}}{\sigma}}}{\sigma p\sqrt{2s}} \\ C &= \frac{1}{\sigma\sqrt{2s}} \\ D &= \frac{\sqrt{2}e^{\frac{b\sqrt{2}\left(\sqrt{p+s}-\sqrt{s}\right)}{\sigma}}}{\sigma\left(\sqrt{p+s}+\sqrt{s}\right)} \end{aligned} \tag{6.12}$$

Finally we get the results for the double Laplace transform of the transition density

$$\widehat{W}(X_T, s) = \begin{cases} \frac{e^{-\sqrt{2s}\frac{|X_T|}{\sigma}} - e^{-\sqrt{2s}\frac{(2b-X_T)}{\sigma}}}{\sigma\sqrt{2s}} + \frac{\sqrt{2}e^{\sqrt{2s}\frac{(X_T-2b)}{\sigma}}}{\sigma\left(\sqrt{p+s}+\sqrt{s}\right)} & X_T < b \\ \frac{\sqrt{2}e^{-\sqrt{2s}\frac{b}{\sigma} - \sqrt{2(p+s)}\frac{(X_T-b)}{\sigma}}}{\sigma\left(\sqrt{p+s}+\sqrt{s}\right)} & X_T > b \end{cases}$$

Rewrite $\widehat{W}(X_T, s)$ for $X_T < b$ as

$$\frac{e^{-\sqrt{2s}\frac{|X_T|}{\sigma}} - e^{-\sqrt{2s}\frac{(2b-X_T)}{\sigma}}}{\sigma\sqrt{2s}} + \frac{2}{\sigma}\int_0^\infty \exp\left[-\sqrt{2(p+s)}y\right]\exp\left[-\sqrt{2s}\left(\frac{2b-X_T}{\sigma}+y\right)\right]dy \tag{6.13}$$

The double inversion on the first two terms is straightforward. For the last integral term using Fubini's theorem and the following double Laplace transform property (note that by design the transition vanishes for $\lambda > T$)

$$\mathcal{L}^{-1}\mathcal{L}^{-1}\{F_1\left(p+s\right)F_2\left(s\right)\} = f_1\left(\lambda\right)f_2\left(T-\lambda\right)\mathbb{1}_{0,T}\left(\lambda\right) \tag{6.14}$$

Then the double inversion for $X_T < b$ is

$$\delta(\lambda)\left[\frac{e^{-\frac{|X_T|^2}{2\sigma^2 T}} - e^{-\frac{(2b-X_T)^2}{2\sigma^2 T}}}{\sigma\sqrt{2\pi T}}\right] + \int_0^{\infty} \frac{y\left(\frac{2b-X_T}{\sigma}+y\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left[-\frac{y^2}{2(T-\lambda)} - \frac{\left(\frac{2b-X_T}{\sigma}+y\right)^2}{2\lambda}\right] dy \tag{6.15}$$

Rewrite $\widehat{W}(X_T, s)$ for $X_T > b$

$$\frac{2}{\sigma}\int_0^{\infty} \exp\left[-\sqrt{2s}\left(\frac{b}{\sigma}+y\right)\right] \exp\left[-\sqrt{2(p+s)}\left(\frac{x-b}{\sigma}+y\right)\right] dy \tag{6.16}$$

Using the same property for the double Laplace transform as before leads to

$$\int_0^{\infty} \frac{(y+\frac{b}{\sigma})\left(\frac{X_T-b}{\sigma}+y\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left(-\frac{(y+\frac{b}{\sigma})^2}{2(T-\lambda)} - \frac{\left(y+\frac{X_T-b}{\sigma}\right)^2}{2\lambda}\right) dy \tag{6.17}$$

Finally we have the following result for the propagator (6.7)

$$\begin{aligned} W_P\left(X_T, T, \lambda|0,0\right) &= \Theta\left(b - X_T\right)\left[\delta(\lambda)\left(\frac{e^{-\frac{X_T^2}{2\sigma^2 T}} - e^{-\frac{(2b-X_T)^2}{2\sigma^2 T}}}{\sigma\sqrt{2\pi T}}\right)\right. \\ &+ \left.\int_0^{\infty} \frac{y\left(y+\frac{2b-X_T}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left[-\frac{y^2}{2(T-\lambda)} - \frac{\left(y+\frac{2b-X_T}{\sigma}\right)^2}{2\lambda}\right] dy\right] \\ &+ \Theta\left(X_T - b\right)\left[\int_0^{\infty} \frac{(y+\frac{b}{\sigma})\left(y+\frac{X_T-b}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left[-\frac{(y+\frac{b}{\sigma})^2}{2(T-\lambda)} - \frac{\left(y+\frac{X_T-b}{\sigma}\right)^2}{2\lambda}\right] dy\right] \\ &:= \Theta\left(b - X_T\right) W_P^-\left(X_T, T, \lambda|0,0\right) + \Theta\left(X_T - b\right) W_P^+\left(X_T, T, \lambda|0,0\right) \end{aligned} \tag{6.18}$$

both integrals can be worked out and their complete expression is given in appendix C.

6.2.3 Option pricing

Define $C\left(S_{X_0}, T, K, B, d\right)$ to be the fair price of a call option with initial price S_{X_0}, maturity T, strike K, Parisian level B and Parisian time d, then based on the risk neutral pricing formula [Shr04], where $\mathbb{E}_\mu$ means that the expectation is to be taken for a Wiener process undergoing a drift μ, the second equality being the result of an application of the Cameron-

Martin-Girsanov theorem [Pas11]

$$\begin{aligned} C\left(S_{X_0},T,K,B,d\right) &= \mathbb{E}_\mu\left[e^{-rT}\left[S_{X_T}-K\right]_+ \mathbb{1}\left[\Theta\left(S_{X_T}-B\right)>d\right]\right] \\ &= e^{-T\left(r+\frac{\mu^2}{2\sigma^2}\right)} \int_{-\infty}^{+\infty} \int_{X_0\to X_T} e^{\frac{\mu X_T}{\sigma^2}} \left[S_{X_0}e^{X_T}-K\right]_+ \mathbb{1}\left[\Theta\left(X_T-b\right)>d\right] d_W x\left(\tau\right) dX_T \end{aligned} \tag{6.19}$$

Introducing the propagator derived in (6.18)

$$\begin{aligned} C\left(S_{X_0},T,K,B,d\right) &= e^{-T\left(r+\frac{\mu^2}{2\sigma^2}\right)} \int_k^{+\infty}\int_d^T \left(S_{X_0}e^{X_T}-K\right) W_P\left(X_T,T,\lambda|0,0\right) d\lambda\, dX_T \\ &= e^{-T\left(r+\frac{\mu^2}{2\sigma^2}\right)} \Bigg[S_{X_0}\int_d^T\Bigg(\int_{k\wedge b}^{b} e^{X_T\left(\frac{\mu}{\sigma^2}+1\right)} W_P^-\left(X_T,T,\lambda|0,0\right) dX_T \\ &+ \int_{k\vee b}^{+\infty} e^{X_T\left(\frac{\mu}{\sigma^2}+1\right)} W_P^+\left(X_T,T,\lambda|0,0\right) dX_T\Bigg) d\lambda - K\int_d^T\Bigg(\int_{k\wedge b}^{b} e^{\frac{\mu X_T}{\sigma^2}} W_P^-\left(X_T,T,\lambda|0,0\right) dX_T \\ &+ \int_{k\vee b}^{+\infty} e^{\frac{\mu X_T}{\sigma^2}} W_P^+\left(X_T,T,\lambda|0,0\right) dX_T\Bigg) d\lambda\Bigg] \end{aligned} \tag{6.20}$$

For completeness sake we want to point out the following obvious relation

$$C^-\left(S_{X_0},T,K,B,T-d\right) = C\left(S_{X_0},T,K\right) - C\left(S_{X_0},T,K,B,d\right) \tag{6.21}$$

with $C^-\left(S_{X_0},T,K,B,T-d\right)$ a variant where the time *under* the Parisian level is counted, and $C\left(S_{X_0},T,K\right)$ is the fair price for a Vanilla option written on the same parameters.

6.2.4 Results

We benchmarked our results against a Monte Carlo simulation, generating 2^{16} paths , with 2^{14} timesteps. We use the following set of parameters: $T=1, \sigma=.2, r=.03, B=110$ in an arbitrary currency unit, and we let the other parameters, i.e Parisian time d and the initial asset price S_{X_0}, vary. As can be seen on the Fig. 6.1, the results obtained through Monte Carlo simulation and the ones derived through our analytical formula agree perfectly.
To no one surprise maybe, the option value is decreasing with respect to an increasing d, since the measure of paths that trigger the payoff itself decreases. Otherwise put:

$$\left\{C\{x_0,0;x_T,T\} : \left(\int_0^T \Theta\left(X_t-b\right)dt > d_1\right)\right\} \subseteq \left\{C\{x_0,0;x_T,T\} : \left(\int_0^T \Theta\left(X_t-b\right)dt > d_2\right)\right\}$$

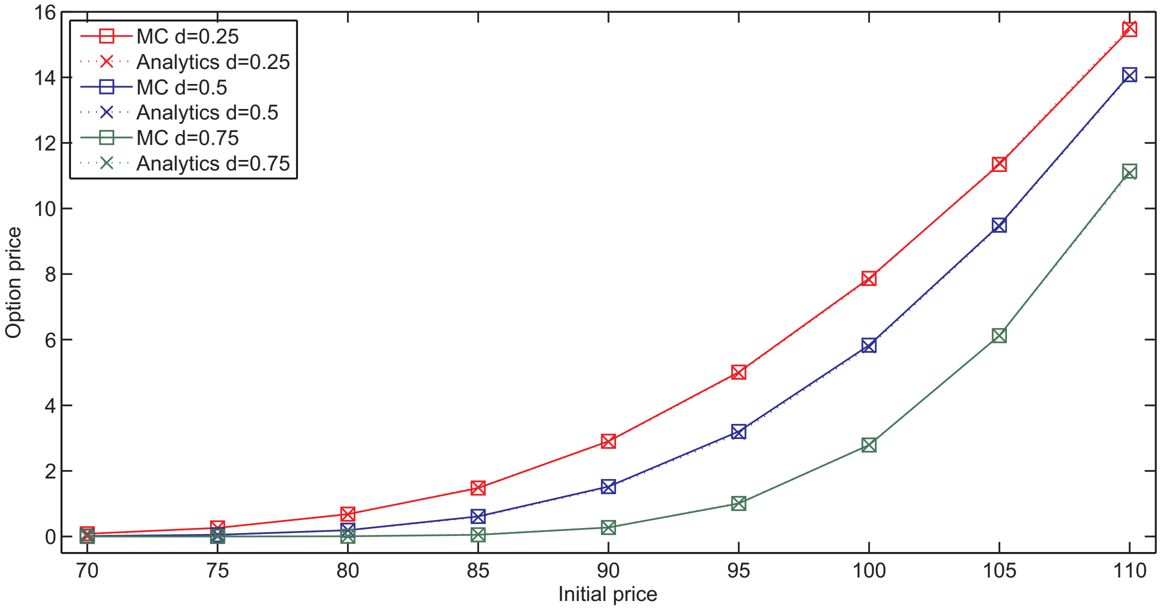


Figure 6.1: The Parisian option fair price (in arbitrary currency unit) with regards to both the initial asset price S_{X_0}, and the Parisian time d. MC stands for the results obtained through a Monte Carlo simulation, while analytics are the results obtained using our exact formula derived in Section 6.2.3.

for $d_1 \leq d_2$.

The results we derive using our analytic formula match the ones obtained with the Monte Carlo simulation, which was expected since in our derivation no approximation was made at any point. Building on those encouraging results we move on to an original product, not studied so far to the best of our knowledge.

6.3 Wasabi option

We introduce now an original exotic occupation time derivative. In addition to the Parisian condition, we let the payoff be a function of the geometric average of the asset price. This payoff feature is usually referred to as “Asian option”. Our original product embedding both Parisian and Asian features will thereafter be called “Wasabi[1] option” for short. Past its theoretical interest, the appeal of this original derivative is to both exhibit the positive features of each separate option, yet to be usually cheaper than both of them.

[1] From the eponymous 2001 French-Japanese movie

6.3.1 System description

Let us introduce the continuously monitored geometric average of S_{X_T} between $t = 0$ and $t = T$, and its log-variable counterpart $\bar{X_T}$

$$\begin{aligned} \bar{S}_{X_T} &:= S_{X_0} \exp\left(\frac{1}{T}\int_0^T \ln\left(S_{X_s}\right) ds\right) \\ \bar{X_T} &:= \ln\left(\frac{\bar{S}_{X_T}}{S_{X_0}}\right) \end{aligned} \tag{6.22}$$

The Wasabi payoff, with level B and time d is then of the following form

$$\left[\bar{S}_{X_T} - K\right]_+ \mathbb{1}\left(\int\limits_0^T \Theta\left(S_{X_t} - B\right) dt > d\right) \tag{6.23}$$

Computation of the option price would entail to derive an exact expression for the following propagator, where the paths have been partitioned according to the final value for the average $\bar{X_T}$. Note that since $\bar{X_T}$ can be negative, the Fourier transform representation for the Dirac Delta is required, leading to an imaginary term in the potential.

$$W_W\left(X_T, \bar{X}_T, T, \lambda|0,0,0\right) := \int\limits_{-\infty}^{+\infty} \frac{e^{iq\bar{X}_T}}{2\pi} \int\limits_{\epsilon-i\infty}^{\epsilon+i\infty} \frac{e^{p\lambda}}{2i\pi}\Big[\int\limits_{X_0\to X_T} e^{-\int\limits_0^T p\Theta(X_t-b)+iq\frac{X_t}{T}\,dt} d_{Wx(t)}\Big]\, dp\, dq \tag{6.24}$$

because the path integral has an equivalent Airy equation form we feel that the computational cost for an exact expression would be prohibitively high. Costly enough that it would discourage its adoption by financial practitioners. Therefore in the following section we propose an approximation and study its accuracy in section 6.3.4.

6.3.2 Wasabi propagator

It is known that the first two moments for the geometric average of the log-price can be exactly derived [KV90]. More explicitly, with r the risk-free rate, σ the annualized volatility for the asset S_X, we can write the following equations regarding the dynamics of our two processes

$$\begin{aligned} X_t &= \left(r - \frac{\sigma^2}{2}\right) t + \sigma W_{X,t} := \mu t + \sigma W_{X,t} \\ \bar{X}_t &= \left(r - \frac{\sigma^2}{6}\right) \frac{t}{2} + \frac{\sigma}{\sqrt{3}}\left(\rho W_{X,t} + \sqrt{1-\rho^2} W_{\bar{X},t}\right) := \bar{\mu} t + \bar{\sigma}\left(\rho W_{X,t} + \sqrt{1-\rho^2} W_{\bar{X},t}\right) \end{aligned} \tag{6.25}$$

where ρ the correlation between X_t and its average $\bar{X}_t$ is known to be equal to $\sqrt{3}/2$. We proceed now to uncorrelate both processes

$$\begin{aligned}\widetilde{X}_t &\leftarrow X_t \\ \bar{\widetilde{X}}_t &\leftarrow \bar{X}_t - \rho\frac{\bar{\sigma}}{\sigma}X_t\end{aligned} \tag{6.26}$$

In the transformed system the Wasabi propagator factorizes into a Parisian propagator for the process $\widetilde{X}_t$ and a free propagator for $\bar{\widetilde{X}}_t$

$$W_W\left(X_T, \bar{X}_T, T, \lambda|0,0\right) = W_P\left(\widetilde{X}_T, T, \lambda|0,0\right) W\left(\bar{\widetilde{X}}_T, |0\right) \tag{6.27}$$

which after reverting to our original variables lead to the following expression for our Wasabi propagator

$$\begin{aligned}W_W\left(X_T, \bar{X}_T, T, \lambda|0,0\right) = \frac{e^{-\frac{\left(\bar{X}_T - \frac{\bar{\sigma}\rho X_T}{\sigma}\right)^2}{2\bar{\sigma}^2(1-\rho^2)T}}}{\sqrt{2\pi\bar{\sigma}^2\left(1-\rho^2\right)T}}\Bigg(\Theta\left(b - X_T\right)\left[\delta(\lambda)\left(\frac{e^{-\frac{X_T^2}{2\sigma^2 T}} - e^{-\frac{(2b-X_T)^2}{2\sigma^2 T}}}{\sigma\sqrt{2\pi T}}\right)\right. \\ \left. + \int_0^\infty \frac{y\left(y + \frac{2b-X_T}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left(-\frac{y^2}{2(T-\lambda)} - \frac{\left(y + \frac{2b-X_T}{\sigma}\right)^2}{2\lambda}\right) dy\right] \\ + \Theta\left(X_T - b\right)\left[\int_0^\infty \frac{\left(y + \frac{b}{\sigma}\right)\left(y + \frac{X_T-b}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left(-\frac{\left(y + \frac{b}{\sigma}\right)^2}{2(T-\lambda)} - \frac{\left(y + \frac{X_T-b}{\sigma}\right)^2}{2\lambda}\right) dy\right]\Bigg)\end{aligned} \tag{6.28}$$

The integrals can yet again be translated to a simpler form available in appendix C.

6.3.3 Option pricing

We derive here a fair price for the Wasabi call option. Following a path similar to the one paved in section 6.2.3.Define $C\left(S_{X_0}, T, K, B, d\right)$ to be the fair price of a Wasabi call option with initial price S_{X_0}, maturity T, strike K, Wasabi level B and time d, then based on the risk neutral pricing formula [Shr04]

$$
\begin{aligned}
&C\left(S_{X_0},T,K,B,d\right)=\mathbb{E}_{\mu}\left[e^{-rT}\left[\bar{S}_{X_T}-K\right]_{+}\mathbb{1}\left[\Theta\left(S_{X_T}-B\right)>d\right]\right]\\
&=e^{-rT}\int\limits_{-\infty}^{+\infty}\int\limits_{X_0\to X_T}e^{-\alpha T+\gamma\bar{X}_T+\beta X_T}\left[S_{X_0}e^{\bar{X}_T}-K\right]_{+}\mathbb{1}\left[\Theta\left(X_T-b\right)>d\right]d_W x\left(\tau\right)dX_T\\
&=e^{-T(r+\alpha)}\left[S_{X_0}\int\limits_{d}^{T}\left(\int\limits_{k}^{+\infty}\int\limits_{k\wedge b}^{b}e^{\bar{X}_T(1+\gamma)+\beta X_T}\frac{e^{-\frac{\left(\bar{X}_T-\frac{\bar{\sigma}\rho X_T}{\sigma}\right)^2}{2\bar{\sigma}^2(1-\rho^2)T}}}{\sqrt{2\pi\bar{\sigma}^2\left(1-\rho^2\right)T}}W_P^{-}\left(X_T,T,\lambda|0,0\right)d\bar{X}_T\,dX_T\right.\right.\\
&+\left.\int\limits_{k}^{+\infty}\int\limits_{k\vee b}^{+\infty}e^{\bar{X}_T(1+\gamma)+\beta X_T}\frac{e^{-\frac{\left(\bar{X}_T-\frac{\bar{\sigma}\rho X_T}{\sigma}\right)^2}{2\bar{\sigma}^2(1-\rho^2)T}}}{\sqrt{2\pi\bar{\sigma}^2\left(1-\rho^2\right)T}}W_P^{+}\left(X_T,T,\lambda|0,0\right)d\bar{X}_T\,dX_T\right)d\lambda\\
&-K\int\limits_{d}^{T}\left(\int\limits_{k}^{+\infty}\int\limits_{k\wedge b}^{b}e^{\gamma\bar{X}_T+\beta X_T}\frac{e^{-\frac{\left(\bar{X}_T-\frac{\bar{\sigma}\rho X_T}{\sigma}\right)^2}{2\bar{\sigma}^2(1-\rho^2)T}}}{\sqrt{2\pi\bar{\sigma}^2\left(1-\rho^2\right)T}}W_P^{-}\left(X_T,T,\lambda|0,0\right)d\bar{X}_T\,dX_T\right.\\
&+\left.\left.\int\limits_{k}^{+\infty}\int\limits_{k\vee b}^{+\infty}e^{\gamma\bar{X}_T+\beta X_T}\frac{e^{-\frac{\left(\bar{X}_T-\frac{\bar{\sigma}\rho X_T}{\sigma}\right)^2}{2\bar{\sigma}^2(1-\rho^2)T}}}{\sqrt{2\pi\bar{\sigma}^2\left(1-\rho^2\right)T}}W_P^{+}\left(X_T,T,\lambda|0,0,0\right)d\bar{X}_T\,dX_T\right)d\lambda\right]
\end{aligned}
\tag{6.29}
$$

with the following constants definition

$$
\alpha:=\frac{\frac{\mu^2}{\sigma^2}+\frac{\bar{\mu}^2}{\bar{\sigma}^2}-\frac{2(\mu\bar{\mu}\rho)}{\bar{\sigma}\sigma}}{2\left(1-\rho^2\right)},\beta:=\frac{\mu-\frac{\bar{\mu}\rho\sigma}{\bar{\sigma}}}{\left(1-\rho^2\right)\sigma^2},\gamma:=\frac{\bar{\mu}-\frac{\mu\bar{\sigma}\rho}{\sigma}}{\bar{\sigma}^2\left(1-\rho^2\right)}
\tag{6.30}
$$

As demonstrated in C, the integrals in the Parisian propagator reduce to erfc $(\cdot)$ evaluation, the remaining integrals converge quickly enough that it takes less than 5 seconds to numerically evaluate on a Xeon 3.20GHz.

6.3.4 Results

We benchmarked our results against a Monte Carlo simulation, generating 2^{16} paths , with 2^{14} timesteps. We use the following set of parameters: $T = 1, \sigma = .2, r = .03, B = 110$ in an arbitrary currency unit, and we let the other parameters, i.e Wasabi time d and the initial asset price S_{X_0}, vary. We also plot the option values for a plain Vanilla option written on the same maturity and strike price. The results can be seen on Fig. 6.2
The approximation leads to an error resulting from the joint use of path partitioning and decorrelation step. Since the paths that trigger the knock-in condition are bound to spend an important share of the time to maturity over an upward level, they ultimately possess a distribution that is skewed compared to the paths normally distributed according to the

distribution of the average. The shift in the distribution of the subset of triggering paths relatively to the distribution of $\bar{X}_T$ can be seen on Fig. 6.3. Therefore the payoff when triggered is taken over an asset whose distribution is off. On Fig. 6.3, the discrepancy in distributions overall location parameters arising as a result of the decorrelation is made obvious.

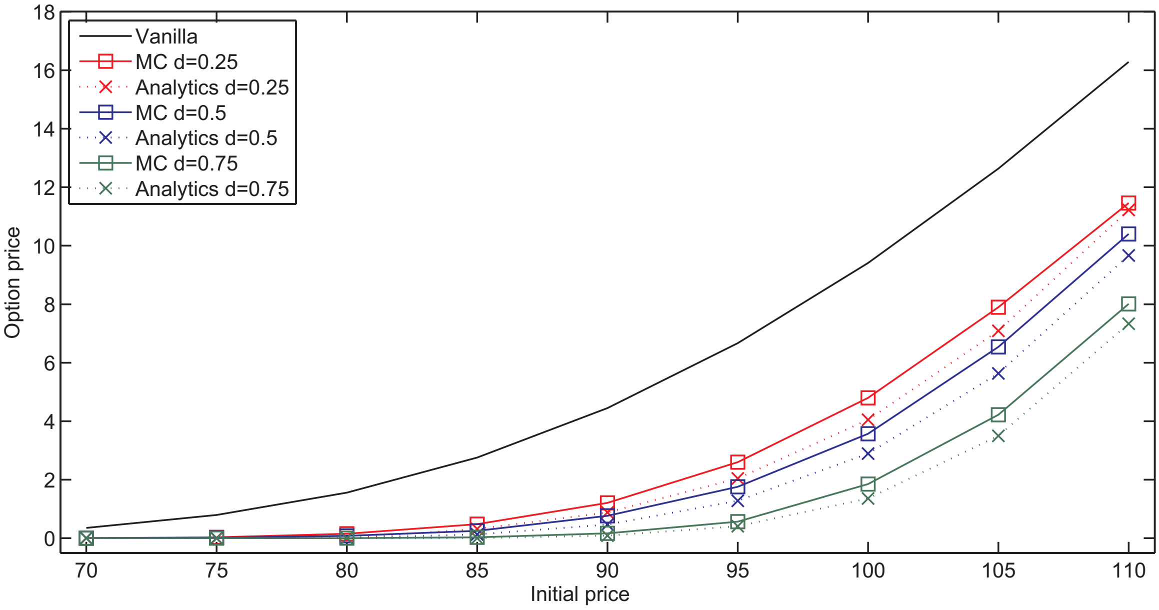


Figure 6.2: The Wasabi option fair price (in arbitrary currency unit) with regards to both the initial asset price S_{X_0}, and the Wasabi time d. MC stands for the results obtained through a Monte Carlo simulation, while analytics are the results obtained using our approximation formula derived in section 6.3.3.

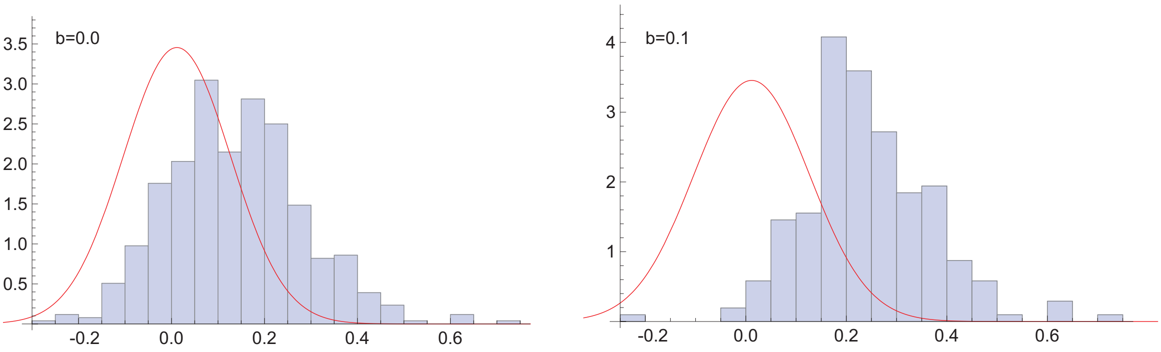


Figure 6.3: Comparing the final price distribution for the subset of paths of X_T triggering the payoff (therein displayed as bins) and for $\bar{X}_T$ (red line), when the Wasabi time is taken to be $d = 0.5$. The left panel is for a Wasabi level at $b = 0.0$, while the right panel is for $b = 0.1$.

It would be trivial to account for this offset by manually widening the distribution of $\bar{X}_T$, for example by doing the substitution $\bar{\sigma} = \sigma/\sqrt{2}$ instead of $\bar{\sigma} = \sigma/\sqrt{3}$ and doing that we would almost recover the same quality of results than seen for the cumulative Parisian case, as visible on Fig. 6.4. However, until it can be more rigorously justified, it should not be elevated to anything more than a "quick fix".

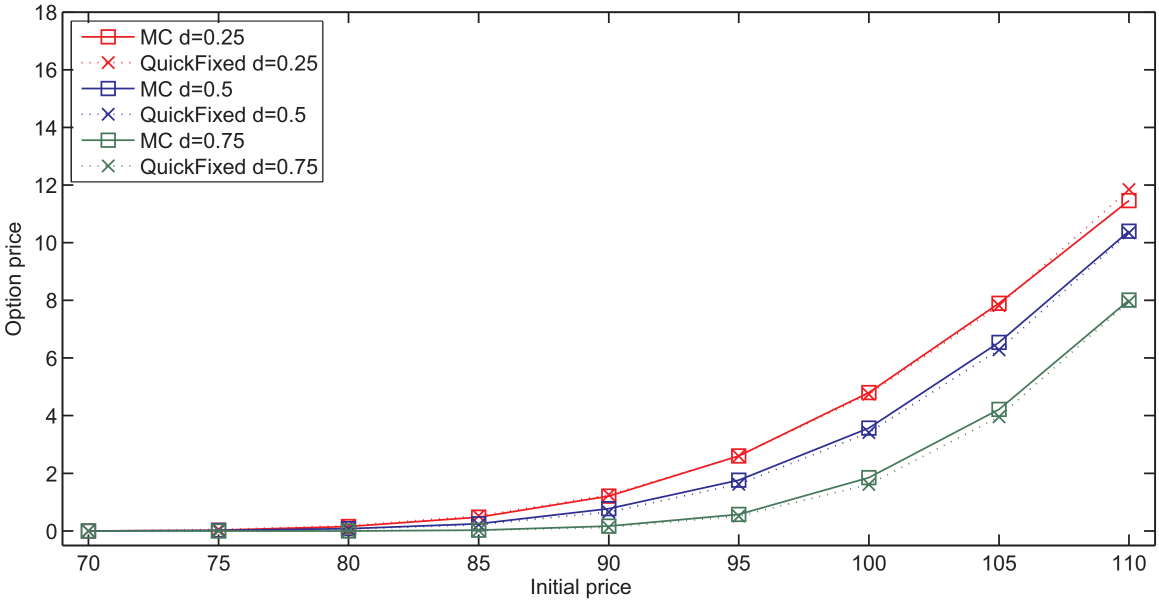


Figure 6.4: The Wasabi option fair price (in arbitrary currency unit) with regards to both the initial asset price S_{X_0}, and the Wasabi time d, when considering a change of volatility from $\bar{\sigma} = \sigma/\sqrt{2}$ to $\bar{\sigma} = \sigma/\sqrt{3}$. MC stands for the results obtained through a Monte Carlo simulation, while "QuickFixed" are the results obtained using our approximation formula derived in section 6.3.3 and the aforementioned substitution.

6.4 Conclusion

In this paper, we show how the path integral framework allows to reformulate the problem of occupation time derivatives into a form that is easily solvable through the Feynman-Kac theorem. Using this gateway, we derived the propagator and used it to solve the cumulative Parisian option. After confirming against a Monte Carlo simulation that the results were exact, we proposed to study a completely original product, that we called Wasabi option.

Since the propagator complex form hinted at computationally prohibitive exact solution, we proposed an approximation using the previously derived Parisian propagator. We checked the relative error of our approximation against a Monte Carlo simulation and proposed an explanation to the origin of results discrepancy, while at the same time demonstrating the efficiency of an easy workaround. Future works should be aimed at improving the current approximation while keeping in check the computational burden.

Part II

Implementation

Chapter 7

General Purpose computing on Graphic Processing Units

In the first part of this thesis we demonstrated the relevance of Econophysics methods when it comes to the theoretical component of the option pricing problem. Moving past the point of theory and looking at the implementation task, we are faced with a different challenge: Given a pricing formula, how to *efficiently* compute the price of an exotic option ? Our metrics to judge efficiency are rather common: accuracy, speed, ease of implementation.
In this thesis we study the distribution to a Graphic Processing Unit (GPU) device and propose some improvements. The choice of a GPU accelerated simulation is a choice that has seen some consideration in the Econophysics community[Pre11a], for its readier availability when compared to a CPU-cluster and the major performance improvement it can exhibit over a single CPU architecture.
General-purpose Computing on Graphic Processing Units (GPGPU) is a term that encompasses techniques and concepts relevant to the use of GPU in simulations or computations. It makes use of the massively parallel architecture of modern GPUs to alleviate CPUs load on tasks where data-parallelism is important[DW10], and provides rather customarily speed up tenfolds when compared with a traditional CPU implementation[1]. To be able to leverage this computational power, a programming environment is needed. NVIDIA impulsed a massive interest for GPGPU when they released CUDA[NVI] in 2007 as a programming framework for their G80 GPU. They released libraries, compilers, debugging tools, etc. A complete development chain, based on C, that makes programming for a GPU a relatively affordable task for one already acquainted with distributed challenges. Each release of a CUDA platform brings new functionality that are then sorted in Compute Capability (CC) from the original 1.0 to currently 5.0. In our study we will be using the CUDA terminology and concepts, since we target NVIDIA GPUs and our implementations are exclusively done using CUDA.
Our presentation of GPGPU will be organized as such: we will first introduce the philosophy behind parallel computation , then we will describe the architecture of a GPU device. Then

[1] The 100x performance gap is rather humorous, and hardly demonstrate anything more than one reluctance with doing proper CPU optimization[LKC$^+$10]

we will finish our tour by discussing some common optimizations and the bottlenecks they address.

7.1 Parallel computing

The formula derived to price exotic options range from straightforward closed form to multiple integrations, and for benchmarking purpose, Monte Carlo simulations. Therefore for the less trivial computation the motivation to decrease the pricing computation time is fully justified. An alternative to running a computing task on a faster hardware is to run a subtask on **more** hardwares with the requirement that subtasks combine in the end to give a similar result. This approach, that is splitting a complex task in computation sub entities and distribute them onto many computing devices is known as parallel computing.
The axis along which the splitting occurs introduce a first level of differentiation in the design space of parallel implementations. Parallelism can be introduced along the data axis or the functional axis: the first leads to data parallelism , while the second is known as task parallelism. **Data parallelism** is a type of code that is best suited when the larger problem shows one task that is iterated over a set of data. As an example, let us think about the task that is to increase brightness in a picture. What it boils down to for the first pixel is then the same processing is done on the next pixel, and this goes on until the last pixel. Therefore, if we have to process a million pixels, running alg. 7.2 sequentially would take a million times the cost of calling the *IncreaseBrightness* procedure. If we had a million computing unit however, each unit could run a single iteration and be done with the task. To drive the point futher, task that involves a set of operation being carried on repeteadly and independently over a data space is said to exhibit data parallelism and can be parallelized quite simply: split the data to be processed in $N \times k$ slices, require N computing devices to each process k slices.

Algorithm 7.1 Pixel brightness modification procedure

```
procedure IncreaseBrightness($i$)                    ▷ Increase brightness of pixel $i$
    $NewBrightnessValue$ =Read(Pixel[i],BrightnessValue)
    $NewBrightnessValue = NewBrightnessValue + 1$
    Save(Pixel[i],NewBrightnessValue)
end procedure
```

Algorithm 7.2 Pixel brightness modification program

```
$i = 0$
for $i < NbPixels$ do
    IncreaseBrightness($i$)
    $i = i + 1$
end for
```

A different approach can be taken in parallelizing a task, when said task involves a sequence of subtasks that can be done independently from one another. Let us take an example: a Monte Carlo simulation to price exotic options. For one path the computation is merely building a random trajectory and finally returning the payoff on this particular path as described in alg. 7.3, and the whole simulation is the average of many calls to this procedure, as available in alg. 7.4.

Algorithm 7.3 Monte Carlo option pricing procedure

```
procedure TrajectoryPricing
    $i = 0$
    for i<NbTimesteps do
        UpdateTrajectory(CurrentUnderlierPriceValue)
        UpdateStateVariables(StateVariables[ ])  ▷ Excursion time, Average price, etc.
        $i = i + 1$
    end for
end procedure
return CalculatePayoff(CurrentUnderlierPriceValue,StateVariables[ ])
```

Algorithm 7.4 Monte Carlo option pricing program

```
$i = OptionValue = 0$
for $i < NbPaths$ do
    $OptionValue = OptionValue +$ TrajectoryPricing
    $i = i + 1$
end for
$OptionValue = OptionValue/NbPaths$
```

Since in this example any of the task of generating a path and computing its payoff can be executed independently without any temporal dependency, we can describe the problem as exhibiting a high level of **task parallelism**. In fact the only part of the program that is not parallelized is the sum and averaging. Distribution is straightforward, require each computing resources to take in charge the pricing for a ratio of the total paths to compute. Task parallelism is less straightforward than data parallelism in term of implementation when thinking about our GPU target hardware. Among other reasons, if tasks execute different code from one another then the code grow rapidly in complexity; also the different tasks may be highly imbalanced in term of computational complexity and some resources may be left idle while more complex tasks complete. This problem is less critical in data parallelism design since all units execute the same job, on different slice of data, thus we can expect every resource to return roughly at the same time.

7.2 GPU architecture

The evolution of CPU has been mainly the story of higher clock rate for a long time, each generation promising better performance through a faster clock. Yet, as CPU reached a 4GHz rate, the heat requires special cooling solution to address the so-called "power wall". Rather than increasing the clock rate, Intel and others opted to increase the number of cores. However to profit from this architecture, the software designers have to take into account the multiple threads that are now able to execute simultaneously. Modern GPU architectures follow the *many-core* trajectory of microprocessor design, where the focus is mainly on the execution throughput of parallel applications. The always increasing number of "cores", and embedded special units at each generation allow for transparent scalability and reduced execution time.

The difference[2] in CPU and GPU architectures arise in part from the difference in the functions they serve. CPU must allow for the seemingly parallel execution of many heavy and complex applications, keeping the apparent sequentiality even when executing out-of-order instructions. Such features are possible only at the expense of an important share of available resources dedicated to control logic. Control logic being the part of the program that is in charge of automatically performing tasks in an execution sequence, statically or dynamically decided. More importantly, since CPU are expected to perform large and complex instructions, eg. Very Long Instruction Word (VLIW), an important amount of cache memory[3] is provided in order to reduce latency attached to data and instruction access. Current GPUs even if they start to provide some amount of cache memory, still do not reach the size available on CPUs.

Streaming processors and multiprocessors A typical NVIDIA GPU card consists of a number of SM (Streaming Multiprocessors), and each SM[4] is comprised of a number of SP (Streaming processors). The SPs execute code simultaneously in steps of (currently) 32 SPs. Streaming processors inside a SM shared a set of on-chip resources such as shared memory, registers, etc. The warp scheduler and dispatcher in each SM is in charge of scheduling and dispatching threads to SP. More precisely threads are scheduled for execution as bundle of threads[5] typically set to 32 threads. A typical and simplified Fermi GPU architecture is

[2] Following Flynn's taxonomy, CPU in majority follows MIMD model while NVIDIA GPU follows the SPMD approach

[3] We recall here that caching is the act of keeping a local copy of a work variable, typically in a very fast type of memory, in order to limit requests to distant slower memory. After the first read request of variable X from global memory, it is cached and subsequent requests concerning X are then processed at cache level. This is not different from the use of cache on CPU architecture.

[4] called SMX on Maxwell architecture and SMM on Kepler architecture

[5] called **warps** in the CUDA world

shown on figure 7.1, while numerical values and how it compares with previous generation is displayed in tbl. 7.1.

Table 7.1: A Comparison of Maxwell GM107 to Kepler GK107

GPU	GK107 (Kepler)	GM107 (Maxwell)
CUDA Cores	384	640
Base Clock	1058 MHz	1020 MHz
GPU Boost Clock	N/A	1085 MHz
GFLOP/s	812.5	1305.6
Compute Capability	3	5
Shared Memory / SM	16KB / 48 KB	64 KB
Register File Size / SM	256 KB	256 KB
Active Blocks / SM	16	32
Memory Clock	5000 MHz	5400 MHz
Memory Bandwidth	80 GB/s	86.4 GB/s
L2 Cache Size	256 KB	2048 KB
TDP	64W	60W
Transistors	1.3 Billion	1.87 Billion
Die Size	118 mm2	148 mm2
Manufactoring Process	28 nm	28 nm
Memory Clock	5000 MHz	5400 MHz
Memory Bandwidth	80 GB/s	86.4 GB/s
L2 Cache Size	256 KB	2048 KB
TDP	64W	60W
Transistors	1.3 Billion	1.87 Billion
Die Size	118 mm2	148 mm2
Manufactoring Process	28 nm	28 nm

Memory model Memory hierarchy on a NVIDIA GPU card relies on a classical tradeoff size versus speed: the fastest memory can only hold a small amount of data, while slower memory hold in the gigabyte order of data. The fastest type of memory available is the **register**, currently 4x64kb on a Maxwell 750ti . *Registers* are typically used to hold variables that are local to a thread. Each thread has its own allocated set of registers. The next fastest type of memory is **shared memory**. As indicated by its name this memory is shared

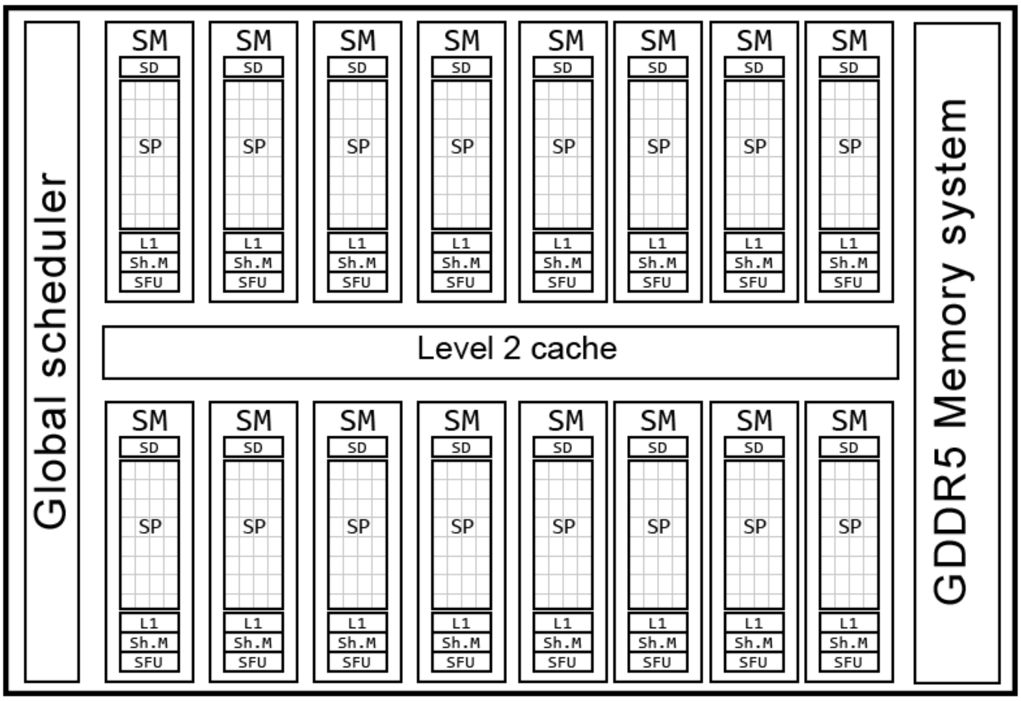


Figure 7.1: A simplified Fermi architecture, where **L1** stands for "Level 1 cache", **Sh.M** for "Shared memory", **SD** for "Scheduler and dispatcher unit" and **SFU** for "Special Function Unit".

among all threads in a block. It is the most efficient way to share data among cooperating threads. Some tuning allows for a partitioning of the shared memory in prefered ratio of L1 cache/shared memory, which may come in handy for special optimization needs. Registers and shared memory are called "on-chip" memories, physically integrated to the SP, which explains their speed.

Slower than shared memory is **local memory**, where local means "local to a thread". It is used when the amount of registers is insufficient (so-called register spilling), and to resolve some indexing issues. It is not fast with any regards since it is not cached and is actually a part of the global memory. Finally the slowest but biggest memory available on GPU cards, is the *global memory* implemented using DRAM. It is available for read and write by any threads in any blocks on the GPU, and also available to the CPU. This is typically where the input data are written by the CPU, read by the GPU, then saved after processing by the GPU and read back from the CPU. Input parameters can also be placed in **constant memory** which is a cached read-only memory, also read from constant memory can be efficiently broadcasted to other requesting threads. Hence it is a low latency high bandwidth type of memory useful for pricing model parameters.

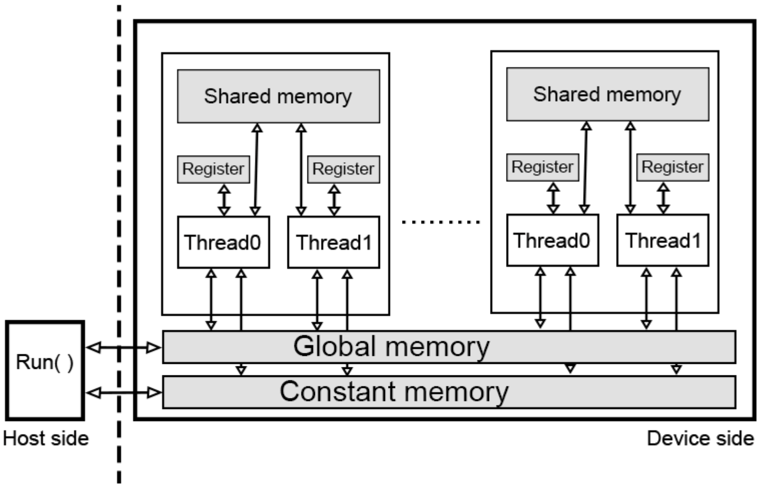


Figure 7.2: Stylised memory architecture of a GPU device, arrows indicate read or write access. Host side is typically the CPU while the device side is the GPU.

Grid, block and thread We now take a higher point of view to describe in a more conceptual fashion how work load is assigned to a GPU and its SM/SP. The code that is effectively running on a GPU is usually self contained in a procedure, that the CPU will ask the GPU to execute. This procedure once running on the GPU can obviously call other sub-procedures if needed[6], but let us consider the case of a single procedure being called. This entry point is called a **kernel**. When the CPU requests a kernel launch, a configuration of grid/block/thread must be specified. We will describe from the bottom to the top what it means. Each instance of a kernel run is assigned to a **thread**, and threads run (locally to a SP) in locked step as a warp of 32 threads[7]. Usually the mapping is one thread handling the processing of one input data. In the case of alg. 7.2, the cpu will map one thread to one instance of the kernel in alg. 7.1. Kernel launch of hundreds of thousands threads is not a rare number in practical applications. Threads are assigned for execution on a SP, and do not migrate until completion. At a higher level threads are bundled in **blocks**. It is the block level that is assigned to a SM and the number of blocks that can be executed on a SM depends on resources availability. At the highest level blocks are bundled in **grid**.

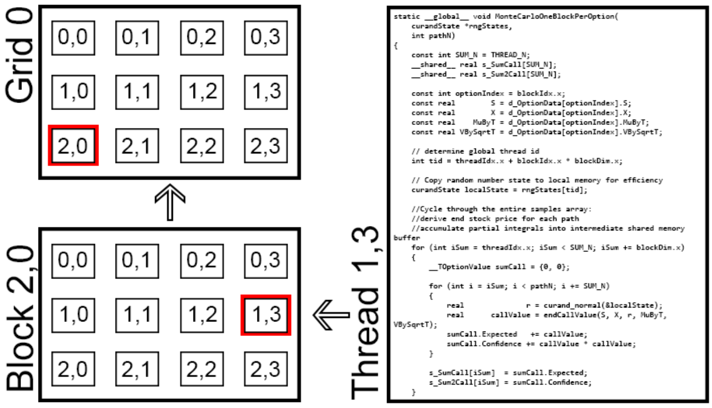
To summarize: kernels are executed as a multidimensional grid of blocks, where blocks are themselves multidimensional arrays of threads. Each of this thread represents an instance of the kernel, working on its own share of the input data. The relevance for more than a single dimension at each level can be demonstrated for computation where the data map more

[6] However recursion is wildly unsupported

[7] The need for warp arise from the fact that there is usually one control unit (for cost reduction and power consumption purpose) decoding the instruction and feeding the result to the SPs. Each of those SPs are controled by the same instruction yet each of them execute differently only because the operands in their registers are different. Close to SIMD in Flynn's taxonomy

elegantly to a 2D or 3D space, such as matrix processing or computer graphics. Figure 7.3 shows a simplified overview of this hierarchy.

An important performance metric that is directly influenced by the block/threads configuration of the kernel launch is the so-called *occupancy.* It is defined as the number of warps running concurrently on a SM, divided by the maximum number of concurrent warps allowed on a SM. A high occupancy rate means that the schedulers mostly find warps to schedule for execution when the current one hits a stall point (e.g. load request, dependent operands not available, etc.). It can also be impacted by resource limits: in the limit case, if a single warp need the full amount of registers available on the SM, then at any time only a single warp can be executing (and frequently *not* stalling). Since it may impact the block/threads configuration one chose, we should point here that by design, synchronization between threads is only allowed inside a block. One reason to constrain locality of synchronization, constraint that goes along with locality of resource assignment, is to allow blocks to execute in any order relatively to each other. The beneficial outcome is that hardware with different set of resources can execute the same code without change (referred to as *transparent scalability*).



```
static __global__ void MonteCarloOneBlockPerOption(
    curandState *rngStates,
    int pathN)
{
    const int SUM_N = THREAD_N;
    __shared__ real s_SumCall[SUM_N];
    __shared__ real s_Sum2Call[SUM_N];

    const int optionIndex = blockIdx.x;
    const real        S = d_OptionData[optionIndex].S;
    const real        X = d_OptionData[optionIndex].X;
    const real    MuByT = d_OptionData[optionIndex].MuByT;
    const real VBySqrtT = d_OptionData[optionIndex].VBySqrtT;

    // determine global thread id
    int tid = threadIdx.x + blockIdx.x * blockDim.x;

    // Copy random number state to local memory for efficiency
    curandState localState = rngStates[tid];

    //Cycle through the entire samples array:
    //derive end stock price for each path
    //accumulate partial integrals into intermediate shared memory
buffer
    for (int iSum = threadIdx.x; iSum < SUM_N; iSum += blockDim.x)
    {
        __TOptionValue sumCall = {0, 0};

        for (int i = iSum; i < pathN; i += SUM_N)
        {
            real                r = curand_normal(&localState);
            real      callValue = endCallValue(S, X, r, MuByT,
VBySqrtT);
            sumCall.Expected   += callValue;
            sumCall.Confidence += callValue * callValue;
        }

        s_SumCall[iSum]  = sumCall.Expected;
        s_Sum2Call[iSum] = sumCall.Confidence;
    }
```

Figure 7.3: Two-level hierarchy between blocks and threads created at grid launch time. Here we restrained ourselves to a two-dimensional example for ease of reading.

One may wonder what is the point of creating more threads that there are actual computing resources ? A major reason is the latency hiding that it may procures. Since load request from global memory are slow to complete, the scheduler once the request has been issued will swap out the threads that issued the request and schedule for execution other available threads (by bundle of 32, a warp). It is a far better strategy than simply waiting idle that the long-latency request complete and continuing with the original threads. Obviously we need a high number of threads in order to achieve latency hiding, and the cost for "context" switching should be low. The first point is why we argued that a large number of threads is a favorable option, the second is ensured by NVIDIA so-called *zero overhead thread scheduling.*

7.3 Optimization

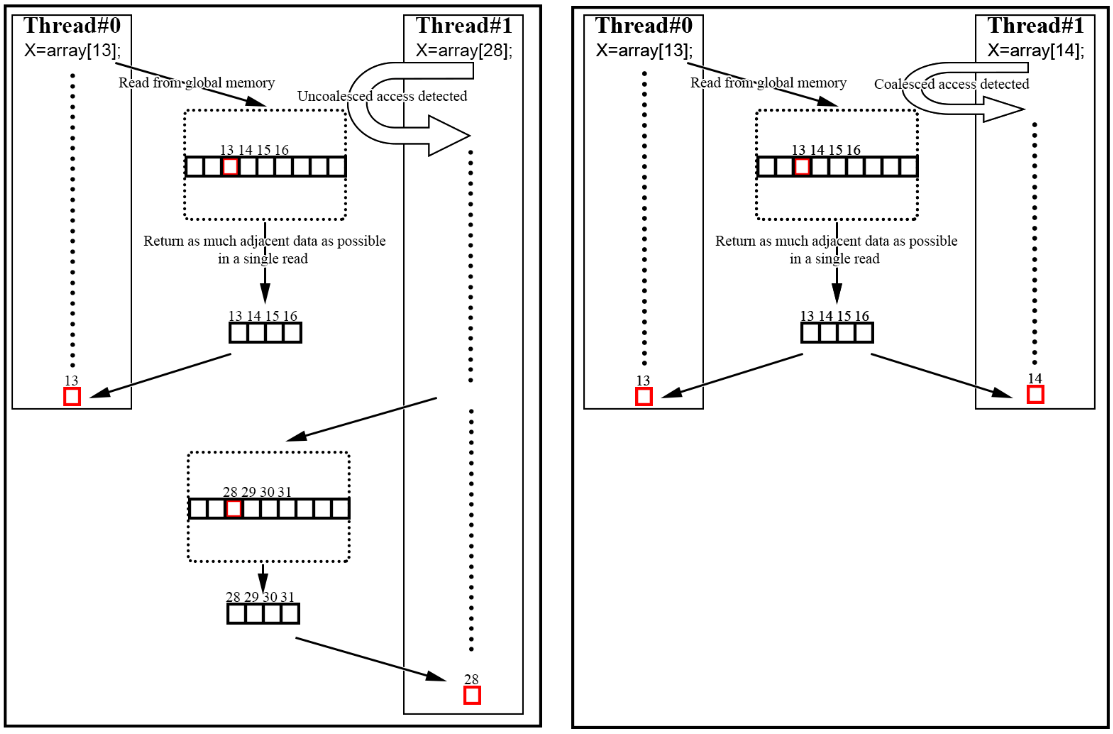


Figure 7.4: Uncoalesced versus Coalesced memory access schematic depiction. On the L.H.S (Uncoalesced memory access): Both threads request data from global memory, since both datum are far away from each other, two read request are generated and only 25% of each returned data are used, effectively wasting a major share of the bandwidth. On the R.H.S (Coalesced memory access): When neighbor threads request datum that sits nearby in global memory, GPUs detect a coalesced memory access and issue a single read request. After returning the whole line of data, each thread can now process their data in half the time that would take two read requests.

Reducing execution time is a topic that asks for a deep grasp of low-level architecture concept. However and even though we do not pretend to exhaustivity here, since optimization is a major measure of GPGPU relevance, we will provide the reader with some *good-practice* that help in reducing execution time.

One of the main bottleneck in GPU computation is global memory access, it is intrinsically a high latency request[DW10], and as such that should be the first place one should try to look for improvements. To see how we can optimize access to global memory, we must first explain how a read request from global memory is processed. We will do so by discarding some technical points irrelevant to our discussion, and redirect interested readers to [DW10]. We said earlier that accessing global memory was a slow process, then it makes sense that in order to improve access rate, modern GPUs typically load not just the requested data, but also the data directly adjacent to it in global memory space. This builds on a major concept in computer architecture called **spatial and sequential locality** [Den]. This concept ar-

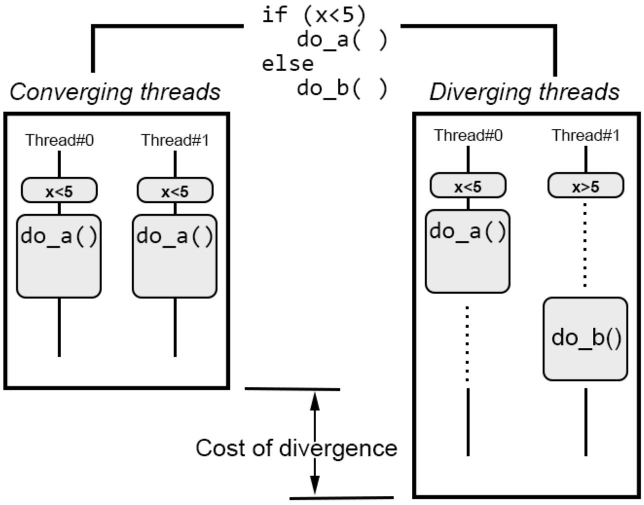


Figure 7.5: Simplified depiction of warp divergence, with a warp made of 2 threads. On the R.H.S, we see threads diverging on their evaluation of the predicate $x < 5$

gues that typically, input elements are accessed in a linear sweep-like fashion. Since GPUs are capable of detecting when some nearby threads (executing in a same warp) are requesting nearby data from memory (so-called *coalesced memory access*), they are also capable of issuing a single read request instead of 4 if 4 threads are working on 4 adjacent datum in memory. Here we see that there lies an important room for optimization in insuring that close-by threads access close-by data in global memory. We provide a simplified picture of coalesced versus uncoalesced memory access, available in fig. 7.4.

Another source of performance cripple is the so-called *warp divergence*. We said earlier that threads were executing inside a SP in pack of threads called warp. The exact number of threads that constitute a warp may vary in the future, it is usually 32. For our discussion, and ease of exposure we will just picture warp of 2 threads. Threads in a warp are supposed to be executing in locked step, the same kernel instructions. Such a model is called **SIMD** (Single Instruction Multiple Data). If two threads in a warp are forced to take different execution paths (e.g a conditional statement *if* evaluate to different results for different threads), the execution is sequentialized. As a result, the total cost of execution for this particular part is the sum of each diverging paths. It is then of critical importance for performance consideration that threads in a warp do not diverge. To go back to a warp of 32 threads, if all threads diverge then the degree of parallelism is null and does not improve one bit over a fully sequential code. Finally, threads that sit in different warps can execute alternative statements without performance penalty. We give a graphical depiction of this phenomenon in figure 7.5.

Chapter 8

Heterogeneous Computation of Rainbow Option Prices Using Fourier Cosine Series Expansion

8.1 Introduction

The option pricing problem is connected to the estimation of a no-arbitrage price for financial derivatives. For simple enough products such as European Vanilla Options, closed-form solutions exist and are mostly the results of the celebrated Black-Scholes formula [BS73]. More involved problems such as the pricing of American, Exotic or high dimensional derivatives, often rely on transform methods such as the Fourier, Laplace or Gauss to handle their increasing complexity[CMS99][DGM09][BY03]. Readers interested in a more complete and technical introduction to the financial asset pricing theory including American options pricing using Markov Chain approximation are redirected to [Pro01][Sim11]. Because of increasingly complex products that are handled on a daily basis by the financial industry, tangentially occurring with the rise of big data problematics, requirements in sheer computational force are expected to increase over time.

General-purpose computing on graphic processing units (GPGPU) is a term that encompasses techniques and concepts relevant to the use of GPU in distribution of computational load. GPUs traditionally used for the rendering of graphics on home computers are, by design, many-cores architectures fit for distributed computing. As such they proved to be viable hardware platforms for computationally heavy tasks with significant data-parallelism. It has been shown to provide speed-up factor customarily tenfolds when compared with a traditional standalone CPU implementation [DW10]. For computational finance it can mean for example, splitting the generation of Monte Carlo paths among idle cores in the hope to divide by the same factor the required execution time [TB10]. It is a popular alternative to the very costly grid and cluster or supercomputer solutions, an alternative that proved viable enough to generate a vast literature in fields as diverse as econophysics[Pre11b], computational chemistry[PDY10], pattern recognition[LSLM10], etc.

While a lot of the available literature so far has focused on comparing "pure CPU" with "pure GPU" implementation [ZO09][BZ12][PGLD11], in this paper we expand the solution perimeter in order to leverage over all available resources. We will propose several heterogeneous distributed GPU/CPU designs for the Fourier-cosine series expansion (COS) pricing of Rainbow options[RO12]. Said derivatives take into account more than a single asset into their dynamics, yet we can maintain a rather vanilla taste to the payoff structure. Therefore its non-trivial dimensionality and tractability of computations proves to be appealing enough for benchmarking load-balancing scheme. We point out to the reader that analytic solution is available for both the option price and the various coefficients that we will build. As we can easily check results against said closed forms, its suitability as a benchmark tool is further justified. By leaving the decision of an optimal load balancing ratio at runtime to the system, we will see major improvements in pricing speed and ease of portability.

The remainder of this paper is organized as follows: Section 8.2 describes the Fourier cosine series expansion method in the context of option pricing. We present our heterogeneous designs and various optimizations in section 8.3. Section 8.4 presents the results and finally in section 8.5 we give the conclusion of our study and discuss some future directions.

8.2 Method

8.2.1 Option pricing

Options are financial derivative instruments, contracts, that give their holders the right (but not the obligation) to buy or sell the underlying asset(s) at a prescribed date, at a determined price.
Therefore the contract need to specify at least: an underlying asset (also called underlier), an exercise date (so-called maturity) T, and the exercise price (so-called strike price) K. An option is said to be European if it is exercisable only on maturity date, Bermudan if there is a discrete set of exercisable dates, and called American if it can be exercised at any time before maturity. Call options give a right to buy, while Put options give a right to sell the underliers both at the prescribed strike price [Pas11][Hul08]. Such derivatives are called Vanilla options since they represent the most simple instruments one can think of. Rainbow (also called Basket) options discussed in this paper belong to the subclass of Exotic options, since they represent more complex and structured products. Rainbow options are written on a basket of more than one underlier, where the dynamics of underlying assets are allowed to be correlated[Hul08]. We will deal in this paper exclusively with European Rainbow option. The dimensionality of the process aggregated dynamics makes it a more complex task to determine a fair price for such a type of option.

The calculation of a fair (also called no-arbitrage) price for an option is a problem which is tightly bound to the type of the options and its payoff (i.e the value of the option at expiry). Obviously increasingly structured derivatives allow for richer payoff functions. For simple enough products (e.g European Put/Call following geometric Brownian motion), the determination of a no-arbitrage price amounts to solving the following Black-Scholes equation with prescribed boundary conditions [BS73][Pas11]

$$\frac{\sigma^2 S^2}{2} \partial_{SS} C(t,S) + rS\partial_S C(t,S) + \partial_t C(t,S) = rC(t,S) \tag{8.1}$$

where σ is the volatility of the asset price process $S(t)$, r the risk-free rate and $C(t,S)$ the fair price of the option. Closed form solution are available for relatively vanilla products and interested readers can find it in any textbooks or in the seminal paper of Black and Scholes [BS73]. American options pricing is a problem that belongs to the class of "free boundary problem" and as such it is a far more complex task than European option pricing [BA05]. Popular techniques to solve the optimal stopping problem that is connected to the pricing of American options are (among others) binomial methods [CRR79], Monte Carlo methods [LS01] and Markov Chain methods [Sim11]. Transform methods applicable to Rainbow option pricing will be introduced in this section, other methods based on approximation of moments are more thoroughly reviewed in [KdKKM04].

8.2.2 The Fourier cosine series expansion method

The main idea behind the COS method and more generally of all methods relying on Fourier Transform, is that the characteristic function of the underlier dynamics for a wide class of assets is explicitly known while the probability density function (hereafter referred to as density function) itself might not be available. This is true for exponential Levy processes such as CGMY (Carr, Geman, Madan, and Yor) processes, Merton model, and for other affine-diffusion processes of non-Levy types. Models which do not fulfill Black-Scholes model requirements (e.g the Heston model has a non constant volatility) no-arbitrage price can not be simply obtained as the solution to Black-Scholes equation.

COS method introduced in the series of papers [FO08][RO12] relies on similar arguments as the ones motivating use of Fourier transform methods, but use a cosine series expansion instead of the traditional Fourier transform. The first motivation for using a series expansion is mainly in the speed of convergence to exact results, the order of which is at least geometric for entire functions[Boy01], i.e. without singularities (poles or branch points) in the complex plane except possibly at ∞ . The more well-behaved the function is, the faster the series converge and the less terms are required in the series expansion. The second motivation that particularly interests us here is the flexibility regarding distribution of computation.

We derive first here an expression for the COS for arbitrarily high dimension n, which, to the best of our knowledge was not available so far in the literature. Analytical results were derived so far up to dimension three in [RO12]. The no-arbitrage price at time t_0 with maturity T for a European Rainbow option over a n-dimensional log-asset process

$\mathbf{X}_t = (X_t^1, \cdots, X_t^n)$ is given by the discounted expectation of the payoff function under risk-neutral measure, the so-called discounted Feynman-Kac formula [Shr04], where we chose without loss of generality $t_0 = 0$,

$$\begin{aligned} C(\mathbf{x}, T) &= e^{-rT} \widetilde{E}[g(\mathbf{X}_T) \mid \mathbf{x}] \\ &= e^{-rT} \int \cdots \int_{R^n} g(\mathbf{y}) f(\mathbf{y} \mid \mathbf{x}) \, d\mathbf{y} \end{aligned} \tag{8.2}$$

Here we used a tilde to denote that we work under risk-neutral measure $\widetilde{\Pr}$, $g(.)$ is the payoff function, C is the option fair price, $\mathbf{x} = \left(x_t^1, \cdots, x_t^n\right)$ is the current log asset price,f is the conditional density function of $\mathbf{X}$ and r is the risk-free rate. For simplicity purpose and without loss of generality we use here a hypercube integration domain $[a, b]^n$, where a and b are chosen according to the shape of the density function available through the cumulants. For a more detailed discussion on the choice of a and b see for example [FO08] or [Pas11]. Truncating the integration range and expanding the density function in cosine series we rewrite the option price S_0 as

$$\begin{aligned} C(\mathbf{x}, T) &\approxeq e^{-rT} \int \cdots \int_{[a,b]^n} g(\mathbf{y}) f(\mathbf{y} \mid \mathbf{x}) \, d\mathbf{y} \\ &= e^{-rT} \int \cdots \int_{[a,b]^n} g(\mathbf{y}) \left[{\sum_{k_1=0}^{\infty}}' \cdots {\sum_{k_n=0}^{\infty}}' A_{k_1, \cdots, k_n}(\mathbf{x}) \cos(\phi_1(y_1)) \cdots \cos(\phi_n(y_n)) \right] dy_{1 \cdots n} \end{aligned} \tag{8.3}$$

with the following definitions

$$\begin{aligned} \phi_i(x) &:= k_i \pi \frac{x-a}{b-a} \\ dy_{1 \cdots n} &:= dy_1 \cdots dy_n \\ {\sum_{i=0}^{\infty}}' u_i &:= \frac{u_0}{2} + \sum_{i=1}^{\infty} u_i \end{aligned}$$

We point out here that $\{A_{k_1, \cdots, k_n}\}$ are the coefficients of the Fourier cosine expansions of $f(\mathbf{y} \mid \mathbf{x})$. Now exchanging the integrations and sums operations, we rewrite $C(\mathbf{x}, T)$ and define $\{V_{k_1, \cdots, k_n}\}$

$$C(\mathbf{x}, T) = e^{-rT} \left(\frac{b-a}{2}\right)^n {\sum_{k_1=0}^{\infty}}' \cdots {\sum_{k_n=0}^{\infty}}' A_{k_1, \cdots, k_n}(\mathbf{x}) V_{k_1, \cdots, k_n}(T) \tag{8.4}$$

$$V_{k_1, \cdots, k_n}(T) := \left(\frac{2}{b-a}\right)^n \int \cdots \int_{[a,b]^n} g(\mathbf{y}) \cos(\phi_1(y_1)) \cdots \cos(\phi_n(y_n)) \, dy_{1 \cdots n} \tag{8.5}$$

We now approximate $\{A_{k_1,\cdots,k_n}\}$

$$A_{k_1,\cdots,k_n}(\mathbf{x}) = \left(\frac{2}{b-a}\right)^n \int \cdots \int_{[a,b]^n} f(\mathbf{y}|\mathbf{x}) \cos(\phi_1(y_1)) \cdots \cos(\phi_n(y_n))\, dy_{1\cdots n} \tag{8.6}$$

$$\approxeq \left(\frac{2}{b-a}\right)^n \int \cdots \int_{R^n} f(\mathbf{y}|\mathbf{x}) \cos(\phi_1(y_1)) \cdots \cos(\phi_n(y_n))\, dy_{1\cdots n} \tag{8.7}$$

Now repeated use of the following two equalities

$$\cos(a)\cos(b) = \frac{\cos(a+b)+\cos(a-b)}{2}$$
$$\cos(\theta) = \Re\left[e^{i\theta}\right]$$

yields

$$A_{k_1,\cdots,k_n}(\mathbf{x}) \approxeq \left(\frac{1}{b-a}\right)^n \sum_{i_1=1}^{2} \cdots \sum_{i_n=1}^{2} \Re\Bigg\{\varphi_{levy}\left(\frac{k_1\pi}{b-a}, \frac{(-1)^{i_1} k_2\pi}{b-a}, \cdots, \frac{(-1)^{i_n} k_n\pi}{b-a}\right) \times \exp\Big[i\phi_1(x_1) + (-1)^{i_1} i\phi_2(x_2) \cdots + (-1)^{i_n} i\phi_n(x_n)\Big]\Bigg\} \tag{8.8}$$

with $\varphi(\mathbf{u} \mid \mathbf{x}) = e^{i\mathbf{x}'\mathbf{u}}\varphi_{levy}(\mathbf{u})$ and $\varphi_{levy} = \exp\left(i\mu'\mathbf{u} - \frac{1}{2}\mathbf{u}'\Sigma\mathbf{u}\right)$ is the characteristic function for a n-variate normal distribution, with Σ the covariance matrix, μ the mean vector and $\mathbf{u}'$ stands for the transpose of $\mathbf{u}$. Finally after truncating the n infinite sums after m terms, the price of an option at t_0 with maturity T is

$$C(\mathbf{x}, T) \approxeq \left(\frac{e^{-rT}}{2^{m-1}}\right) \sum_{k_1=0}^{m-1}{}' \cdots \sum_{k_n=0}^{m-1}{}' A_{k_1,\cdots,k_n}(\mathbf{x})\, V_{k_1,\cdots,k_n}(T) \tag{8.9}$$

It is then immediate that results in [RO12] are obtained by setting $n = 2$ or $n = 3$. Since we will deal in this paper exclusively with a two dimensional rainbow call option of European type exercise, we shall give here a more tractable expression for the option price in that particular case

$$\begin{aligned} C(\mathbf{x}, T) &= \frac{e^{-rT}}{2} \sum_{k_1=0}^{n-1}{}' \sum_{k_2=0}^{n-1}{}' A_{k_1,k_2}(\mathbf{x})\, V_{k_1,k_2}(T) \\ &= \frac{e^{-rT}}{2} \sum_{k_1=0}^{n-1}{}' \sum_{k_2=0}^{n-1}{}' \Bigg(\Re\Bigg[\varphi\left(\frac{k_1\pi}{b-a}, \frac{+k_2\pi}{b-a} \mid \mathbf{x}\right) \exp\left(i\phi_1(x_1) + i\phi_2(x_2)\right) \\ &+ \varphi\left(\frac{k_1\pi}{b-a}, \frac{-k_2\pi}{b-a} \mid \mathbf{x}\right) \times \exp\left(i\phi_1(x_1) - i\phi_2(x_2)\right)\Bigg]\Bigg) V_{k_1,k_2}(T) \end{aligned} \tag{8.10}$$

with

$$V_{k_1,k_2}(T) \equiv \frac{4}{(b-a)^2} \int\int_{[a,b]^2} \max\left(\sqrt{e^{y_1}}\sqrt{e^{y_2}} - K{,}0\right) \cos\left(k_1\pi\frac{y_1-a}{b-a}\right) \times \cos\left(k_2\pi\frac{y_2-a}{b-a}\right) dy_1 dy_2 \tag{8.11}$$

8.3 GPGPU Implementation

We will now look at different strategies regarding computational load balancing between GPU and CPU, and give some directions concerning optimization. In the different heterogeneous implementation designs of the model, we focus on two key indicators: efficiency (with respect to speed) and accuracy of the result. Those two indicators will be evaluated with respect to the number N of terms in each dimension of the COS and with respect to the different integration schemes. We defined efficiency here as $\frac{\min(TimeToCompletion)}{TimeToCompletion}$.

8.3.1 Computational load distribution

The first choice regarding implementation design is with regard to which axis (e.g. among functions or data) we wish to split the problem, keeping in mind that the computational load is mainly located in the V_{k_1,k_2} coefficients computation where numerical approximation of a double integral is involved. For fairness of comparison and efficiency concern, every major loop on CPU side is parallelized using OpenMP [Boa08]. We present here four original heterogeneous CPU/GPU designs and two purely homogeneous ones (i.e full CPU and full GPU) that will be used later-on for comparisons.

- D_0: Computation of V_{k_1,k_2} is done by the GPU while C_{k_1,k_2} are computed on host side.
- D_1: Computation of V_{k_1,k_2} is done by the CPU while C_{k_1,k_2} are computed on device side. It is obviously the symmetric design of D_0.
- D_2: The GPU is in charge of the computation of $[RN, N] \times [1, N]$ coefficients in the 2D space $[V_{k_1,k_2} \times C_{k_1,k_2}]$, with $0 < R < 1$ the GPU load ratio, while the CPU handles the computation of the remaining $[1, (1-R)\,N] \times [1, N]$ coefficient.
- D_3: This design takes the same distribution approach as D_2 but reduces the domain of integration from a square domain to a triangular one as will be explained in section 3.3
- D_4: Full CPU (all coefficients are computed by the CPU).
- D_5: Full GPU (all coefficients are computed by the GPU).

The severe imbalance that we can foresee in computation of V_{k_1,k_2} and C_{k_1,k_2} lead us to think that the functional distribution embedded in D_0 and D_1 may have scalability issues. Indeed as we increase the number of coefficients, computational load bound to V_{k_1,k_2} will outweigh the one bound to C_{k_1,k_2}. However, what we plan to achieve is to balance out CPU and GPU computation time, thus minimizing idle time on both devices simultaneously.

The GPU load ratio R that controls how much of the total computational load should go to the GPU is ultimately a to-be-tuned parameter, though we can predict that optimal value should ultimately be found around

$$\frac{GPU\ peak\ throughput}{(GPU + CPU)\ peak\ throughput}$$

This prediction holds for kernel that achieve peak performance and when both host and device sides perform the same computation, or comparable with respect to required time. Building on that idea, we here implement a simple performance calibration phase during which a small chunk of identical computation is sent to both CPU and GPU. On return, respective times to completion are compared and a split ratio is automatically decided. This flexible way to decide the suitable load balancing ratio should allow us to achieve best performance when targeting different hardware platform, without the need to re-tune the software.

The only global memory accesses are done at the beginning of the kernel execution to read the constant model parameters, then the final access to global memory is to write back final results on completion. There is no need to communicate between work-items during computation and no need for synchronization between work-items either. As we see the problem is a typically embarrassingly parallel problem.

8.3.2 Numerical approximation of V_{k_1,k_2}

In the original paper [RO12] the authors proposed to use the two-dimensional discrete cosine transform(2D-DCT) to evaluate the coefficients V_{k_1,k_2} for its ease of implementation and mainstream availability regarding APIs. We make here the choice to write an implementation of two-dimensional composite Simpson rule (2D-CSR) for its higher order accuracy. We will discuss a way to efficiently implement it on GPU and then propose a domain reduction to speed up the computation of V_{k_1,k_2}.

The CSR approximation relies on a spatially discretized grid with equidistant points where the integrand is evaluated, as such it belongs to the class of Newton-Cotes formula. The integral is approximated by a weighted sum of the integrand value at those points

$$I \approx \frac{(b-a)^2}{9M^2} \sum_{i=0}^{i=M} \sum_{j=0}^{j=M} w_{i,j} f\left(x_i, y_j\right) \tag{8.12}$$

where I is the exact value of V_{k_1,k_2}, $w_{i,j}$ is the $[i, j]$ entry in the $v' \times v$ matrix with $v = [1, 2, 4, 2, \cdots, 4, 2, 1]$ and $v[i]$ is the weight assigned to the i^{th} point in a 1D-CSR. It is well known that the 2D-CSR is of order $O(4)$ whereas a 2D-DCT method is only order $O(2)$ [VSR06]. From an implementation-oriented point of view the trade-off is mainly located in the increased complexity due to selecting the correct sampling weights based on the point position in the domain. A trivial implementation on CPU would run a nested loop to cover the domain and discriminate between weights based on the index of sampled points. Then, using modulo operation on the 2D index the correct weighting factor can be easily retrieved. We want to avoid both conditional statements and the modulo operator since it is well known that they tend to hurt computational performance on GPU. Conditional statements because they lead easily to warp divergence, and modulo operations because they are known to be a very costly arithmetic operation on early generation GPUs [GHK+11]. Without pretending here to exhaustivity, we feel compelled to briefly describe how warp divergence arises since it is one of the main contribution of this paper to propose an implementation free of any divergence.

GPUs implement an execution model called *SIMD* (Single Instruction Multiple Data). In this particular model, sets of threads called warps are supposed to be executing, in locked step, the same instructions of the GPU code. If two threads in a warp are forced to take different execution paths (e.g a conditional statement *if* evaluates to different results for different threads), the flow of execution is sequentialized. As a result, the total cost of execution for this particular part is the sum of each diverging paths. It is then of critical importance for performance consideration that threads in a warp do not diverge.

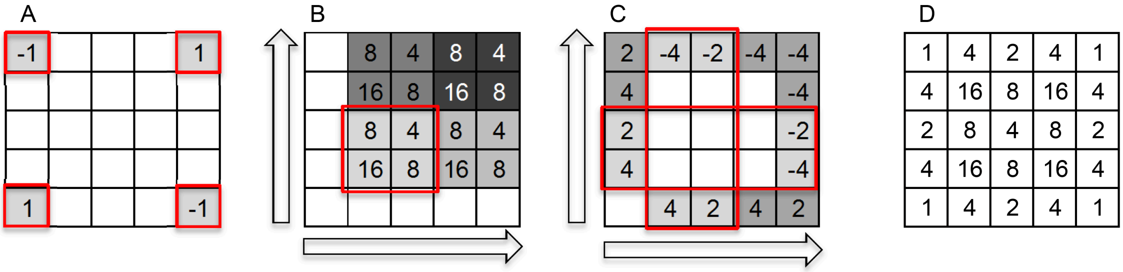
The algorithm we proposed for the 2D-CSR on GPU is free of any conditional statements, and as such do not let warp divergence cripple execution time. Profiling of our application with the tool “Compute Visual Profiler” by NVIDIA confirmed a 0% divergent branches as expected.

Our implementation on GPU of the 2D-CSR in pseudo-code is given in Alg. 8.1, a graphic overview of the process is given by Fig. 8.1.

It is worthwhile to note that in alg. 8.1, as we go around the domain in a block-wise fashion, we unroll the loop by a factor 2 in each direction, thus reducing the necessary number of comparison and loop increment by the same factor. Loop unroll is a well known source of speed up. By reducing the number of iteration of the loop we reduce the logical and arithmetical operations embedded in loop.

Algorithm 8.1 GPGPU implementation of CSR on a square domain

```
Require: $[a,b]^2$: spatial domain of integration.
  $M$: odd number of discretizing points in each dimension.
  $f(\cdot)$: integrand defined in $V_{k_1,k_2}$
Ensure: 2D-CSR approximation of $V_{k_1,k_2}$ over a square area
  $Result \leftarrow f(a,a) - f(a,b) - f(b,a) + f(b,b)$
  for $i = 1$ to $M-1$ step 2 do
      $Result+ = 4\big[f(a_i,a) + f(a,a_i) - f(a_i,b) - f(b,a_i)\big] + 2\big[f(a_{i+1},a) + f(a,a_{i+1}) - f(a_{i+1},b) - f(b,a_{i+1})\big]$
      for $j = 1$ to $M-1$ step 2 do
          $Result+ = 16f(a_j,a_i) + 8\big[f(a_{j+1},a_i) + f(a_j,a_{i+1})\big] + 4f(a_{j+1},a_{i+1})$
      end for
  end for
  $Result\times = \frac{(b-a)^2}{9M^2}$
```



-1				1
1				-1

	8	4	8	4
	16	8	16	8
	8	4	8	4
	16	8	16	8

2	-4	-2	-4	-4
4				-4
2				-2
4				-4
	4	2	4	2

1	4	2	4	1
4	16	8	16	4
2	8	4	8	2
4	16	8	16	4
1	4	2	4	1

Figure 8.1: A graphic overview of two-dimensional composite Simpson's rule on a toy (5×5)domain. Numbers correspond to weights affected to the sampled point in this particular step. Square in the same grey scale are sampled in the same loop iteration. For efficiency purpose steps B and C are merged together in the loop iterations.***A***: We first proceed by sampling the edge points of the square domain, outside of any loop. ***B***: We then proceed inside the domain sampling 4 adjacent points in each iteration of the inner loop, going up to and including the right and top border. ***C***: We finally proceed on the border strip sampling 8 points at a time on the left and bottom border, while correcting the weights affected to points on the top and right border. ***D***: The combination of steps A,B and C yield the correct weights for each sampled points.

8.3.3 Domain reduction

It is direct that by rewriting V_{k_1,k_2}, the integrand is vanishing over a major part of the domain

$$\begin{aligned} V_{k_1,k_2} &= \frac{4}{(b-a)^2}\int\int_{[a,b]^2} \max\left(\sqrt{e^{y_1+y_2}}-K,0\right)\cos\left(k_1\pi\frac{y_1-a}{b-a}\right)\cos\left(k_2\pi\frac{y_2-a}{b-a}\right)dy_1dy_2 \\ &= \frac{4}{(b-a)^2}\int_a^b\int_{2\ln(K)-y_2}^b \left(\sqrt{e^{y_1+y_2}}-K\right)\cos\left(k_1\pi\frac{y_1-a}{b-a}\right)\cos\left(k_2\pi\frac{y_2-a}{b-a}\right)dy_1dy_2 \end{aligned} \tag{8.13}$$

This is basically just a consequence of the all-or-nothing nature of the payoff function. Same conclusion can be intuitively drawn by plotting level curves of the absolute value of the integrand in the square domain $[a,b]\times[a,b]$, as can be seen in Fig. 8.4. Therefore we can increase speed even further if we avoid evaluating the integrand over the whole area where it is known to be vanishing; the integration domain is then no longer a square and can be accurately approximated by a triangular domain with vertices in Cartesian coordinates $(a,b)\,,(b,a)\,,(b,b)$. We also point out here that by using this modified domain, the accuracy of results should be improved for the following reason. We are going to skip over part of the domain where the integrand is vanishing, but also the part where it is extremely small $(\approx< 1e-6)$ by neglecting a small strip around the vanishing border. The minimum border is now $2\log(K)-(a+ih+\epsilon)$, where ϵ can be tuned on demand by offline analysis of the payoff magnitude structure, or as a first step, by visually plotting the integrand as in Fig. 8.4. Now, it is known from [DW10] that when working with simple precision arithmetics, adding up a large number of small values may lead to round off errors that turn out to be prohibitively large. Our modified triangular domain should lessen this artifact too.

Since Alg. 8.1 was written to run efficiently over a square domain, it can no longer be used without adjustments. We propose here a simple modification of 1D-CSR and use iterated quadrature over 1D for evaluating the 2D integral in V_{k_1,k_2}. Rigorous argument for the validity of this approach relies on Fubini's theorem.

Algorithm 8.1 is now modified and takes the form of Alg. 8.2 where we find it easier to work with the spatially discretizing step h rather than the number M of integration points

8.4 Results

The CPU side code is written in C++ using the OpenCL1.1 C++ wrapper, GPU side code is written using OpenCL C99. The code running on the host side is parallelized further using OpenMP spreading to 8 threads. The GPU we used is a NVIDIA TeslaC2070 and the CPU we used is an Intel Xeon X5647 2.93GHZ. Performance results are averaged over 10

Algorithm 8.2 GPGPU implementation of CSR on a triangular domain

```
Require: $[a,b]^2$ is the spatial domain of integration.
  $h$ is assumed to be the same for each dimension.
  $f(\cdot)$ is the integrand defined in $V_{k_1,k_2}$.
  $K$ is the strike price
Ensure: 2D-CSR approximation of $V_{k_1,k_2}$ over a triangular area
  for $i \leftarrow 1$ to $\frac{b-a}{h}$ do
    Compute coordinate of left boundary
    $LeftBnd = 2\log(K) - (a + ih)$
    $InnerResult += f(LeftBnd, b - ih) + f(b, b - ih) + 2f(b - h, b - ih)$
    for $j = 1$ to $\frac{b-a}{h} - 1$ step 2 do
      $InnerResult += 4f(LeftBnd + jh, b - ih) + 2f(LeftBnd + (j + 1)h, b - ih)$
    end for
  end for
  $Result += h\left(\frac{h}{3} \times InnerResult\right)$
```

runs with the same set of parameters. We use here the parameters set I in [RO12] for which analytical results exist and yield a fair price ≈ 8.88 in arbitrary domestic currency unit.

8.4.1 Scalability

The imbalance between the load on the computations of V_{k_1,k_2} and computations of C_{k_1,k_2} render a functional distribution to CPU and GPU of those coefficients (D_1) inefficient in regard to scalability. Evidence of this imbalance is made obvious on Fig. 8.2 when looking at the scalability curve with respect to N. This imbalance prevents D_1 to be of any use past $N = 16$, and it actually does not scale any better than the pure CPU design D_4. This can be explained empirically by the fact that since the major load is located in computing V_{k_1,k_2} there is small to no gains in “distributing” this load to the CPU. We can also see clear empirical evidence that every heterogeneous designs we proposed (excluding D_1), not only run faster than the pure GPU design D_5, but also scale strictly better. The speed-up factor between D_2 and D_3 as can be seen on Fig. 8.2 is not as big as one could imagine first, since we must take into account that it is no longer possible to manually unroll the loop in each dimension, but only in one as can be seen in the inner-loop of Alg. 8.2.

All curves from Fig. 8.2 can be fitted as cN^k with c a constant $\ll 1$ and $k \approxeq 2$ illustrating that the code is roughly scaling as $O\left(N^2\right)$. For illustration purpose, we give here the fitting curves for D_1 and D_3 (Fitted curves and residues are plotted on Figure 8.2)

$$D_1 \approxeq 2.6 \times 10^{-3} N^{2.022}, D_3 \approxeq 4 \times 10^{-4} N^{1.961}$$

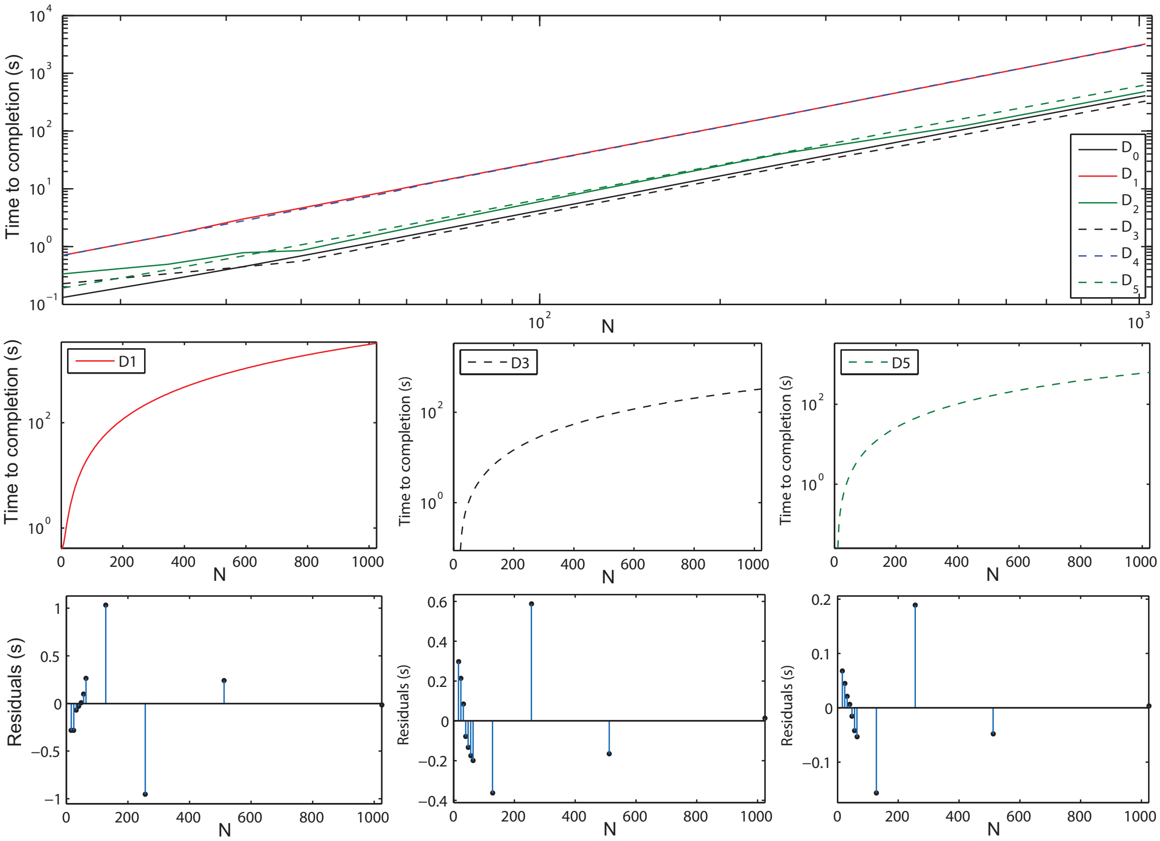


Figure 8.2: Top panel:Speed performance and scalability behavior of heterogeneous computational designs with respect to N, the number of terms in each dimension of the series expansion. **Middle panel**: Fitted quadratic scalability curves for D_1, D_3 and D_5 where points $N = 2^i, i \in [1..10]$ were used. **Bottom panel**: Residual errors of the quadratic fitting as represented on the middle panel for the same designs. For all panels $R = R^* = 0.82$ was used as a load-balancing value.

Figure 8.3 shows that performance degraded rather rapidly when diverging from an optimal GPU load ratio $R^* \approx 0.82$, this is under the ratio derived from technical considerations only

$$\frac{GPU\ peak\ throughput}{(CPU + GPU)\ peak\ throughput} \approxeq .9 \tag{8.14}$$

though as we said earlier, this optimal value is theoretically achievable only under perfect condition rarely found in hardware at use. However, R as automatically decided during the performance calibration phase by comparing the relative time to completion (T.T.C) of small computation chunk sent to the GPU and to the CPU was found to be extremely close ($|R - R^*| \approx 1\%$) to the optimal one

$$\frac{GPU\ T.T.C\ computation\ chunk}{(CPU + GPU)\ T.T.C\ computation\ chunk} \approxeq .835 \tag{8.15}$$

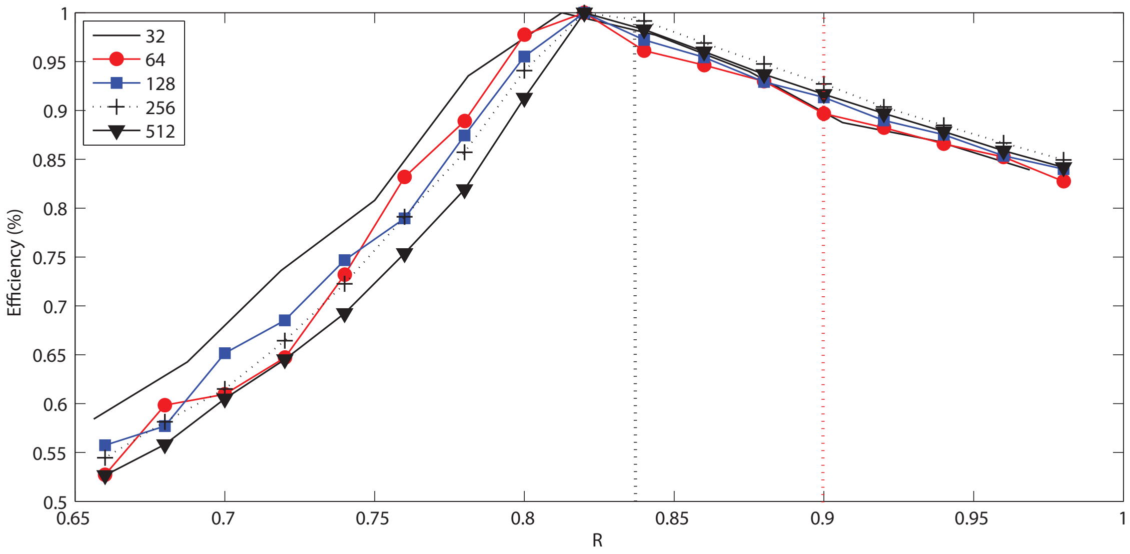


Figure 8.3: Efficiency of design D_3 as a function of the GPU load ratio R. Vertical dashed black line indicates R value automatically selected by our calibration phase whereas the red one stands for ideal value derived from technical specifications.

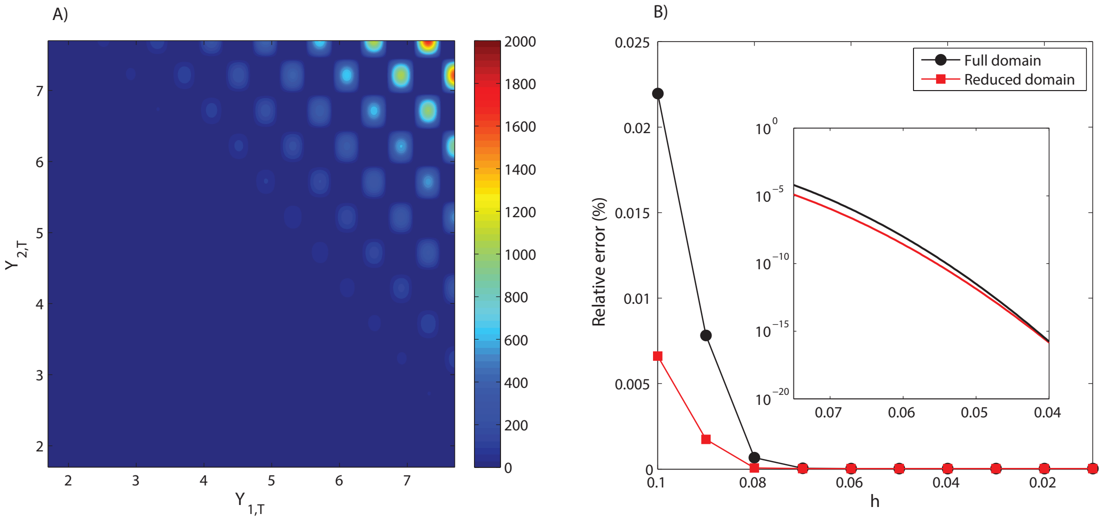


Figure 8.4: A:Magnitude of integrand for $[Y_{1,T}, Y_{2,T}] \in [a, b]$ in arbitrary currency unit. **B**: Relative error of option price using a full square domain versus using the reduced domain with respect to the discretization step h. 64 terms in each dimension of the series expansion have been used

Figure 8.3 confirms that automatic decision of the load balancing ratio R is not only simple enough to implement, but also yields performance close to the optimal ones. Performance decreased faster when under-using the GPU capacity than when over-using it, as expected from the tolerance of GPU architecture to massive load [DW10]. For small N (eg. $N \leq 32$), it is best to have $(1 - R)\,N = 8$ the physical numbers of threads available on our CPU. Let us also point out that for very small N, such as $N \leq 8$ it becomes relevant to compile the kernel in an off-line manner since the cost of compilation now start to weigh too much in an otherwise quick computation.

To summarize we can say that the design D_3 that offer a balanced distribution of computational load while reducing the costly task of integration is the one that performed best. Unsurprisingly the scheme that achieved the worst performance (over ten times slower) was the pure CPU design D_4.

8.4.2 Relative error

Numerical experiments results as displayed on Fig. 8.4.B show that accuracy is improved by using a reduced domain, when comparing with the full domain. Relative errors are given as the normalized distance from the computed price to the correct one $= |\frac{\widetilde{C}-C}{C}|$ where $\widetilde{C}$ is the calculated price and C is the exact one (i.e 8.88077 in arbitrary currency unit), also available in [RO12]. The explanation for the observed accuracy improvement stems from the reduction that we achieved on the required number of integrand evaluations on the domain; and as such, the slower accumulation of round-off errors. It is well known that single precision calculation is prone to fast accumulation of round-off errors and as such, it is a critical optimization that as few calculations as possible are done.

8.5 Conclusion

Building on previous works, we first derived a general formula for n-dimensional COS. Focusing on two dimensional options we achieved a major speed-up from a pure CPU implementation to a pure GPU one, and improved the speed even further by designing an heterogeneous CPU/GPU implementation that leverages over all available computational resources. We extracted the best design from the proposed ones. The ratio of computational load to be distributed to CPU and to GPU was showed to be easily decidable at runtime, and ultimately very close to the best ratio as found from experimentation. We also proposed a simple enough way to reduce the integration domain from a square to a triangle and as such reduced the burden for a kernel which is already Computation-bound on GPU. A two-dimensional CSR algorithm designed to execute rapidly on GPU has also been presented that exhibits no warp divergence and embeds some degree of loop unroll by-design. Further works should be directed to find a more flexible way to balance the computational load between available resources in an online fashion.

Chapter 9

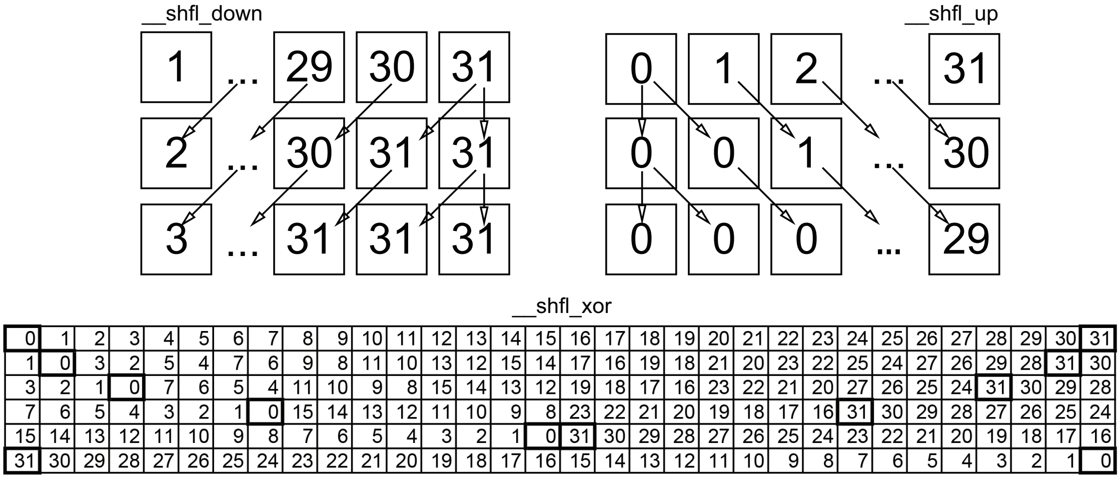
Shuffle up and deal : accelerating GPGPU Monte Carlo simulation through recycling

9.1 Introduction

When concerned with deriving a fair price for an option contract, in the non-trivial cases, one often have to resort to numerical techniques such as Monte Carlo simulations or finite element methods. The Monte-Carlo (MC) methods is especially suited for products deemed path-dependents such as Asian options, barrier options, etc. Also, since at its heart, MC methods require to build a massive amount of paths independent from one another, it is a technique that lend itself fairly well to parallelization. Techniques to accelerate MC convergence are usually based on variance reduction consideration, while the computational time can itself be reduced by accelerating the very generation of random numbers. The technique we propose here does not belong to either of those optimizations, and relies more on exploiting a computational gap between the creation and the recycling of random numbers. Put simply, if it is significantly easier to recycle a number than create a new one, there is a potential opportunity for major speed-up.

General-purpose computing on graphic processing units (GPGPU) stands for techniques and concepts relevant to the use of GPU as a platform for heavy computation. GPUs, since they were designed to handle the parallel workloads associated with graphics rendering, exhibit many-cores architectures where each core itself hosts numerous computation engines. As such they proved to be an especially fit hardware target for computationally heavy tasks exhibiting significant data-parallelism.

The remainder of this chapter is organized as follows. In section 9.2, we will briefly introduce the technical background necessary to our study: the path dependent product that we will be working with, so called “Wasabi option”. The application of Monte Carlo integration to the option pricing problem, then the Granger causality test that we use to help us in our selection step. We will also introduce the CUDA framework. In the following section 9.3 we will present our implementations on GPU, discussing pattern and period of shuffling. Section 9.4 presents the results evaluated with respect to speed and accuracy and finally we



0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	0	3	2	5	4	7	6	9	8	11	10	13	12	15	14	17	16	19	18	21	20	23	22	25	24	27	26	29	28	31	30
3	2	1	0	7	6	5	4	11	10	9	8	15	14	13	12	19	18	17	16	23	22	21	20	27	26	25	24	31	30	29	28
7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Figure 9.1: Top panel: Depiction of the effect of __shfl_up on the threads with low ID. The squares stand for the content to be swapped in each thread in a warp, different rows represent the evolution of the content in each thread registers after a call to __shfl_up. Bottom panel: Illustration of a butterfly swap pattern. In this example, the content being shuffled around is the thread ID and the different calls to __shfl_xor occur from top to bottom, with the xor mask being powers of 2 between 2^0 and 2^4.

will conclude and introduce our future work in the last part 9.5.

9.2 Background

9.2.1 The Wasabi option

Working in the Black-Scholes market model [BS73], the Wasabi option is a product whose payoff involves two path-dependent state variables: the geometric average of the asset price and the excursion length over a predetermined level. More explicitly, for an underlier with value S_t at time t, the payoff function $H(\cdot)$ of the Wasabi option is

$$\left[S_0 e^{\left(\frac{1}{T}\int_0^T \ln(S_t)\,dt\right)} - K\right]_+ \mathbf{1}_{\left\{(\int\limits_0^T \Theta(S_t-B)\,dt>d\right\}} \tag{9.1}$$

where K is the strike price, T the maturity date, B the so-called Wasabi level and d the Wasabi time. It knocks in under the condition that the asset spends an amount of time over d, in the region $S_t > B$.If the condition is realized it delivers a call on the Geometric average of the asset. Therefore it can be said that it merges the Asian payoff, the first factor in (9.1), and the Parisian condition,the second factor in (9.1), into a single derivative product.
In [?], the authors propose a pricing formula that requires the computation of double integrals, which even without posing analytical difficulties may be better handled with the use

of Monte Carlo methods to simulate asset random paths.

9.2.2 Monte-Carlo simulation

The use of Monte-Carlo methods for option pricing is well documented and further details can be found in [Gla04], we will just give here an overview of the philosophy. Starting with the following geometric Brownian motion modeling the asset dynamics

$$S_t = S_0 \exp\left[\left(r - \frac{\sigma^2}{2}\right) t + \sigma W_t\right] \tag{9.2}$$

whose discretized $0 = t_0 < ... < t_i < ... < t_N = T$ version is

$$S_{t_{i+1}} = S_{t_i} \exp\left[\left(r - \frac{\sigma^2}{2}\right) \Delta t + \sigma\sqrt{\Delta t} Z_{i+1}\right] \tag{9.3}$$

with $\Delta t \ll 1$ the timestep, r the risk-free rate, σ the annual volatility, and Z_i a standard normal variable independent of Z_j for $i \neq j$. Using the recursion formula (9.3), we can generate a sample paths for the underlying asset, updating the state variables along the way. The payoff H_i is then calculated on this i^{th} paths using (9.1) and finally the option value $\tilde{H}$ is approximated based on the law of large numbers with

$$\tilde{H} = \frac{1}{M} \sum_{i=0}^{M} H_i \tag{9.4}$$

with M the number of simulated paths. As is well known from MC theory, the error scales as

$$\frac{1}{\sqrt{M}} \tag{9.5}$$

(independently of the problem dimensionality), therefore a high number of paths simulations may be required for problems where accuracy is critical.
Particularly attractive to us is the fact that each payoff can be computed independently from each others which leads to particularly straightforward parallelization onto many cores architectures.

9.2.3 CUDA

CUDA is a now mainstream framework developed by CUDA [NVI] mainly in C/C++, to handle the programming for GPU for non-graphic related computation. The GPU originally destined for the gaming industry have seen a major interest as a platform for general computation. For all intents and purpose, we can consider the GPU as a stream processors running a Kernel in parallel on many SMM each of them holding a large number of cores. For SIMD problems, the straightforward mapping is one thread handling the processing of an input/output element. Here, one thread is responsible for the generation of one trajectory. A technical aspect particularly pertaining to our analysis is the granularity in which threads are executed in-step once elected for execution. Threads are bundled in group of 32 called a warp in CUDA terminology, and a warp is the limit in which we can use the shuffle instruction. Paraphrasing again, data can not be shuffled between threads that belongs to different warps. This limitation has the positive side-effect that we can assure that threads/paths with ID further than 32 away will be as uncorrelated as they would be in a traditional MC simulation.

The shuffle instructions we keep referring to are "__shfl, __shfl_up, __shfl_down, __shfl_xor" Cuda instructions, available for platforms with cc 3.x. They allow data to be shuffled between registers, without any trip to local memory and as such exhibit satisfying speed. They are mainly used to improve speed for reduce and broadcast operations. The different shuffle types distinguish themselves by the way the source thread ID is retrieved: __shfl asks explicitly for a thread ID inside the lane, __shfl_up(resp. down) retrieves the ID just over (resp. under) the caller ID, and __shfl_xor does a bitwise XOR between the caller and the given parameter ID.
As we stated in the introduction, our motivation in proposing the shuffled Monte-Carlo technique is to exploit a computational gap between generating random numbers and merely shuffling them. In our benchmarks, generating 2^{16} random numbers using the MRG32k3a generator takes 16.3319s while shuffling the same volume of number using __shfl_up takes only 0.356s.

9.2.4 Granger causality test

One way to quantify the causal relationship between time series is the Granger causality test introduced in [Gra69]. This statistical test allows one to measure how past values of a time series X add to the predictive power of an univariate regression of the time series, where lagged values are deemed meaningful based on F-statistics. It is intuitively based on the idea that 'X (Granger)causes Y if knowing the past of X helps predict the future of Y'. For illustration, let us write X and Y under the following bivariate (for generality) autoregressive form

$$\begin{aligned} X(t) &= \sum_{j=1}^{p} A_{XX,j} X(t-j) + \sum_{j=1}^{p} A_{XY,j} Y(t-j) + \xi_X(t) \\ Y(t) &= \sum_{j=1}^{p} A_{YY,j} X(t-j) + \sum_{j=1}^{p} A_{YX,j} Y(t-j) + \xi_Y(t) \end{aligned} \tag{9.6}$$

with $\xi_X(t), \xi_Y(t)$ the residual prediction errors for time series X and Y, and p is the order of the model. X causes Y if the variance of $\xi_X(t)$ is reduced by keeping in some Y terms in the first equation. The same argument holds to quantify causality in the reverse direction, looking now at the variance of $\xi_Y(t)$. Finally, the magnitude of the causal influence, $F_{X\to Y}$ of X on Y is measured by

$$F_{X\to Y} = \ln \frac{\mathrm{var}\left(\xi_{XR(XY)}\right)}{\mathrm{var}\left(\xi_{XU}\right)} \tag{9.7}$$

where $\xi_{XR(XY)}$ is derived from a restricted model setting $A_{XY,j} = 0, \forall j$ in the first line of (9.6) and ξ_{XU} originates from the full model. As before the same derivation holds to quantify in the reverse direction, setting $A_{YX,j} = 0, \forall j$ in (9.6), etc.

9.3 Shuffled Monte Carlo

As was briefly touched upon in the previous section, simulating a single path (among the N total paths)requires the generation of M random numbers (more precisely *pseudo* random numbers), where M is traditionally very large. Then the total number of random numbers generated for a simulation is $N \times M$. We propose here to interject some shuffling of random numbers among threads currently running on a SM instead of generating fresh ones (we will thereafter refer to this implementation as Shuffled Monte Carlo, or SMC). Since there are no access to global memory after Kernel launch and before reading the results back, we point out that ultimately the kernel will be compute-bound.
The motivation for SMC lies behind the speed of a shuffle operation relatively to the creation of a new random numbers. The downside being that in doing so, a correlation to be measured is introduced between neighboring (w.r.t. SM) paths, therefore the shuffle should remain marginal. We define the shuffling period (SP) to be the number of random numbers actually generated before a shuffling sequence occurs. Hereafter and for brevity sake, when we write *SMC27*, we actually refer to the shuffled Monte Carlo implementation with shuffling period $SP = 27$.

9.3.1 Shuffling pattern

Along with the period of shuffling, the other critical point is the pattern of said shuffling. In section 9.2.3 we introduced the 4 built-in flavors available in Cuda, obviously we should base our choice on the pattern that minimize the correlation or causal dependence introduced between paths. Since the number of possible patterns is in the order of $(32!)^M$, we must reduce our empirical study to something we can study and implement in a reasonable fashion. We present in figure 2 the results on the Granger causality test[Gra69] for a sequence of 5 shuffles interjected after generating 27 random numbers, repeated over 2^{16} timesteps. As can be seen on figure 2, the test correctly picks up the directionality component for the various patterns in which the shuffle occurs. The Xor exhibits the prototypical hierarchical dependency pattern that we expected, while the Down and Up shuffles demonstrate an

obvious limitation of such a trivial implementation, that we explain right after.
The threads with an ID equal to 31, upon calling __shfl_down can not receive anything from a superior thread, since as we pointed earlier, shuffles occur within a warp that host 32 threads. Therefore when the thread 31 calls __shfl_down, it actually recovers the content of its own variable. The thread with ID 30 which received the content of ID 31 on the first call, will receive once again the same content on the second call, since thread 31 recovered its own register. Therefore we see that there is a strip (whose width is connected to the shuffle length SL) of threads on the superior side of the warp lane that is bound to exhibit a high correlation in the direction $ID_i \to ID_{i-1}$. The same argument holds for thread with low ID and __shfl_up. To fix ideas, a visual representation is available regarding __shfl_up and __shfl_down in figure 9.1.

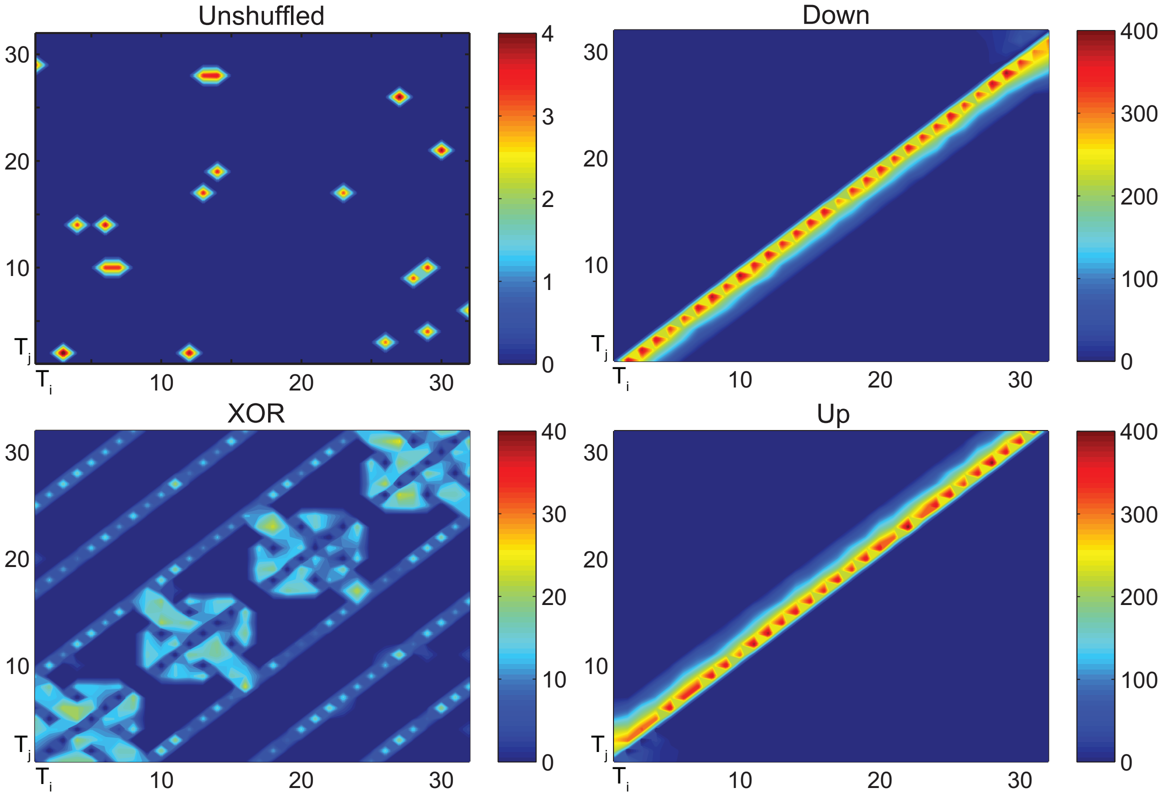


Figure 9.2: F-statistic value on the Granger Causality test for the unshuffled and the various style of shuffling. T_i (resp. T_j) stand for paths generated by thread with ID i (resp. j). The F-statistic value in cell $[T_i, T_j]$ is for the test $F_{T_i \to T_j}$ where only the results that have a p-value < 0.01 are kept.

9.3.2 Implementation

Regarding the generation of random number itself, relying on CURAND, we have the choice to use the following generators

- XORWOW: Xor-shift added, Marsaglia sequences [Mar03]
- MRG32k3a: 32-bit combined multiple recursive generator with two components of order 3
- MTGP32: Mersenne twister for graphic processors [SM13]
- Philox_4x32_10: Counter based parallel RNG[SMDS11]

where the trade-off lies between speed, memory footprint, and quality of pseudo random numbers.We will use for the scope of this study the “MRG32k3a” generator.

Finally the SMC implementation looks as such

Algorithm 9.1 SMC pseudo code

```
Require:
  $N$: Number of timesteps.
  $SP$: Generating period.
  $SL$: Shuffling length
Ensure: $Result \leftarrow H(S_T)$
  for $i = 1$ to $N/(SL + SP)$ do
    for $j = 1$ to $SP$ do
      $RNb = curand\_normal(\&localState)$
      $x* = exp(Drift + RNb * Vol)$
      $AvgX+ = log(x)$
      if $x > b$ then
        $ExcursionTime+ = dt$
      end if
    end for
    for $j = 1$ to $SL$ do
      $RNb = shfl\_\_xor(RNb, 1 << j)$
      $x* = exp(Drift + RNb * Vol)$
      $AvgX+ = log(x)$
      if $x > b$ then
        $ExcursionTime+ = dt$
      end if
    end for
  end for
  if $ExcursionTime > d$ and $exp(AvgX/N) > k$ then
    $H \leftarrow exp(AvgX/N) - k$
  else
    $H \leftarrow 0$
  end if
```

where b and k are the log-variables counterpart to B and K. Since both SL and SP are hard-coded *#define* values, the compiler has no problem unrolling the loops on *#pragma unroll*.

9.4 Results

The GPU used is the GeForce GTX 750Ti, a relatively cheap gaming GPU targeting personal desktop, hosting 5 SMs(640 cores) and 2GB of GDDR5. The code is compiled using the CUDA 6.0 [NVI] and Visual Studio 2012 build chain for Windows, we turned on the highest level of optimization and did the computation with single precision fast maths. We build our unidimensional blocks out of 512 threads, thus the number of blocks is $N\%512$.
We present our results for two main implementations, and some variations. The first is the traditional Monte Carlo (MC) simulation where every random numbers has been freshly generated with the CURAND library using a curandStateMRG32k3a generator, all simulations run over $M = 2^{18}$ timesteps. The traditional MC will serve as a baseline to measure the performance of our shuffling technique, both in terms of accuracy and speed. Unsurprisingly, the second set of measures displayed thereafter belongs to our SMC implementation where the shuffling pattern is a XOR butterfly swap, for reasons explained in section 9.3.
We benchmarked SMC for varying period of shuffling

- SMC3: SP=3, SL=5
- SMC11: SP=11, SL=5
- SMC27: SP=27, SL=5

The limit $SL = 5$ is a result both of the XOR choice as a shuffling pattern, and the limit of 32 for the size of threads participating in a swap.

9.4.1 Accuracy

First and foremost, we need to check our scheme accuracy against two problems admitting an analytical solution: the Vanilla and the Geometric Asian option. This will help up to put forward enough empirical evidence that SMC does not introduce any artifacts in the MC convergence property. Therefore, the first set of error-checking will involve the pricing of a Vanilla option. For this simpler case than the Wasabi evoked before, the payoff is just $[S_T - K]_+$ and admits a fairly straightforward difference of Gaussian CDFs.
The parameters are

- maturity T: 1 year
- annual risk-free rate r: 0.03
- annualized volatility σ: .2
- strike price K: 100
- initial price S_0: 100

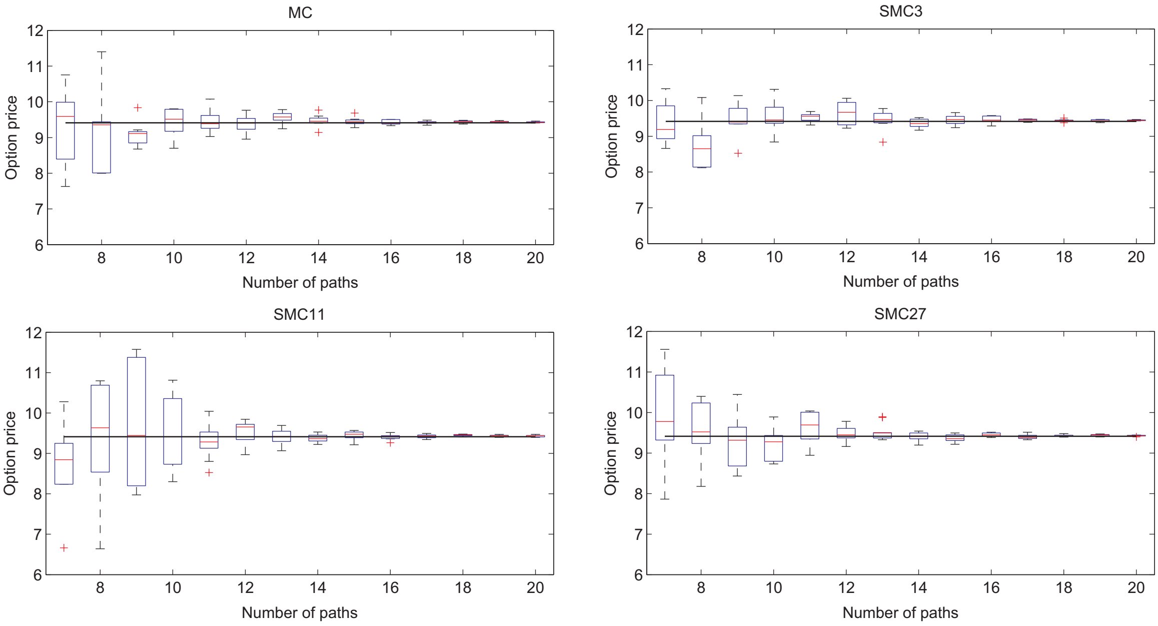


Figure 9.3: Box plots for the MC and SMC implemementation of the Vanilla option pricing problem. The analytical solution (the black line) is 9.4134 in arbitrary currency unit.

We present in figure 9.3 a set of box plots summarizing 10 runs for a number of generated paths between 2^7 and 2^{20}, for both MC and SMC with different periods. We do not find any non accidental, and qualitative shift between the distributions of a traditional MC implementation and our SMC. If any, the MC3 which is the implementation using the shuffling in the most intensive fashion actually looks as close to the analytical solution as the MC implementation. One can see from Fig. 9.3 that the period of shuffling has the biggest impact on the distance to the correct price for simulations that involve a low number of simulated trajectories. For simulation involving 2^{14} paths, SMC with a period 3 exhibit a relative error defined as $\frac{|SMC-MC|}{MC}$ that is ≈ 0.03 while the SMC with a period of 27 is off by approximatively 0.02 for the same number of trajectories. Once we move past 2^{17} paths the error does not exceed 0.01, and this for any shuffling period. Therefore, depending on the precision required by the target simulation a large number of paths can be required, but then again, this is a feature built in MC theory itself and is not a limitation introduced by our shuffling technique.

We continue our proof of concept, by running the same tests on a Geometric Asian option. This option, while it is path dependent and therefore fully justify using MC, still admits an analytical solution that take the same form as the Vanilla case. The payoff this time is $\left[S_0 e^{\left(\frac{1}{T}\int_0^T \ln(S_t)\,dt\right)} - K\right]_+$ and the parameters set remains the same as in the Vanilla simulation. The results are available in the various box plots of figure 9.4 for MC and the various SMC. As for the previous Vanilla case, but this time on a path dependent product,

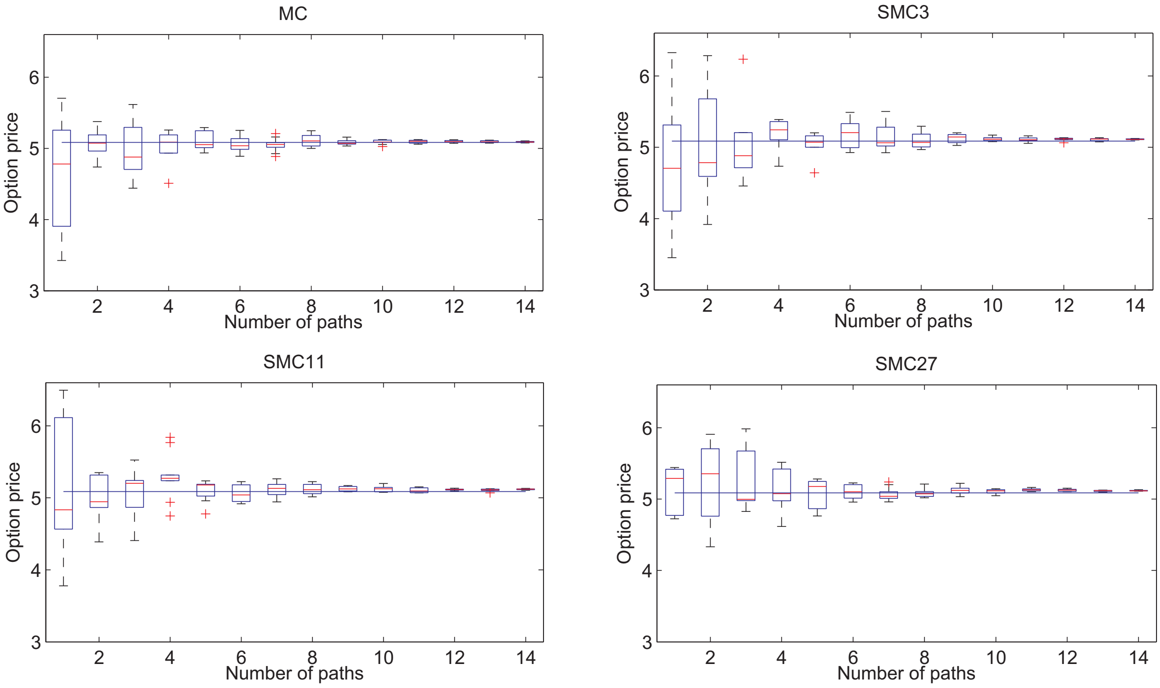


Figure 9.4: Box plots for the MC and SMC implemementation of the Geometric Asian option pricing problem. The analytical solution (the black line) is 5.086 in arbitrary currency unit.

we feel that the visual evidence put forward in figure 9.4 are solid enough to justify us when we say that the shuffling introduced in our SMC implementations do not *break* MC convergence property.

Now that we checked that the SMC implementation we propose in this study do not introduce errors of higher magnitude than the traditional MC scheme using products admitting closed form analytical solution. We now move on to our real target: the Wasabi option, introduced in section 9.2.1, where we will see in terms of speed up the major interest of the SMC technique; for sake of coherency we will also display the accuracy benchmark for the SMC pricing of Wasabi option in the next section. They will confirm the comments we made in the current section, and therefore are merely introduced for the most curious readers.

9.4.2 Speed

The Wasabi pricing parameter set is as such

- maturity T: 1 year

- annual risk-free rate r: 0.03
- annualized volatility σ: .2
- strike price K: 100
- initial price S_0: 100
- wasabi level B: 100
- wasabi time d: .5

Readers interested in the analytical solution involving multiple numerical integrations are redirected to [CCO14b]. We present here the results of our timing benchmark for the same target platforms as the ones considered in section 9.4.1. We will however limit the speedup study to the only Wasabi case, that is our target. Other and simpler products exhibit similar speedup behavior and therefore are omitted.

As can be observed in figure 9.5, the goal we set ourselves is properly achieved for any number of paths, meaning that the SMC exhibit lower computation time for any number of computed paths. This is mainly due, as we claimed earlier, to the differential in computational cost between *__shfl_xor()* and *curand_normal()*.

The speed-up with respect to the MC implementations are respectively

- $\approxeq$ 1.14 (SMC27)
- $\approxeq$ 1.32 (SMC11)
- $\approxeq$ 1.97 (SMC3)

Evidently enough, for a low volume of computed trajectory the relative gain is hardly noticeable: computing 8192 paths take $1.25s$ for the basic MC while it takes $.66s$ for the SMC3. However once we reach the 10^5 computed paths the gain is important enough to make our strategy relevant: for 2^{18} paths the computation time is $65.37s$ for the basic MC while it takes the SMC3 only $33.15s$. Therefore we feel that our approach is best suited to simulations where a very large volume of simulated trajectories is required.

Computational time scales linearly as $a \times m + \varepsilon$ where a, ε are implementation dependent and listed above for sake of completeness

- MC : $a \approxeq .00025, \varepsilon = .45615$
- SMC27 : $a \approxeq .00022, \varepsilon = .40837$
- SMC11 : $a \approxeq .00018, \varepsilon = .4066$
- SMC3 : $a \approxeq .00012, \varepsilon = .27524$

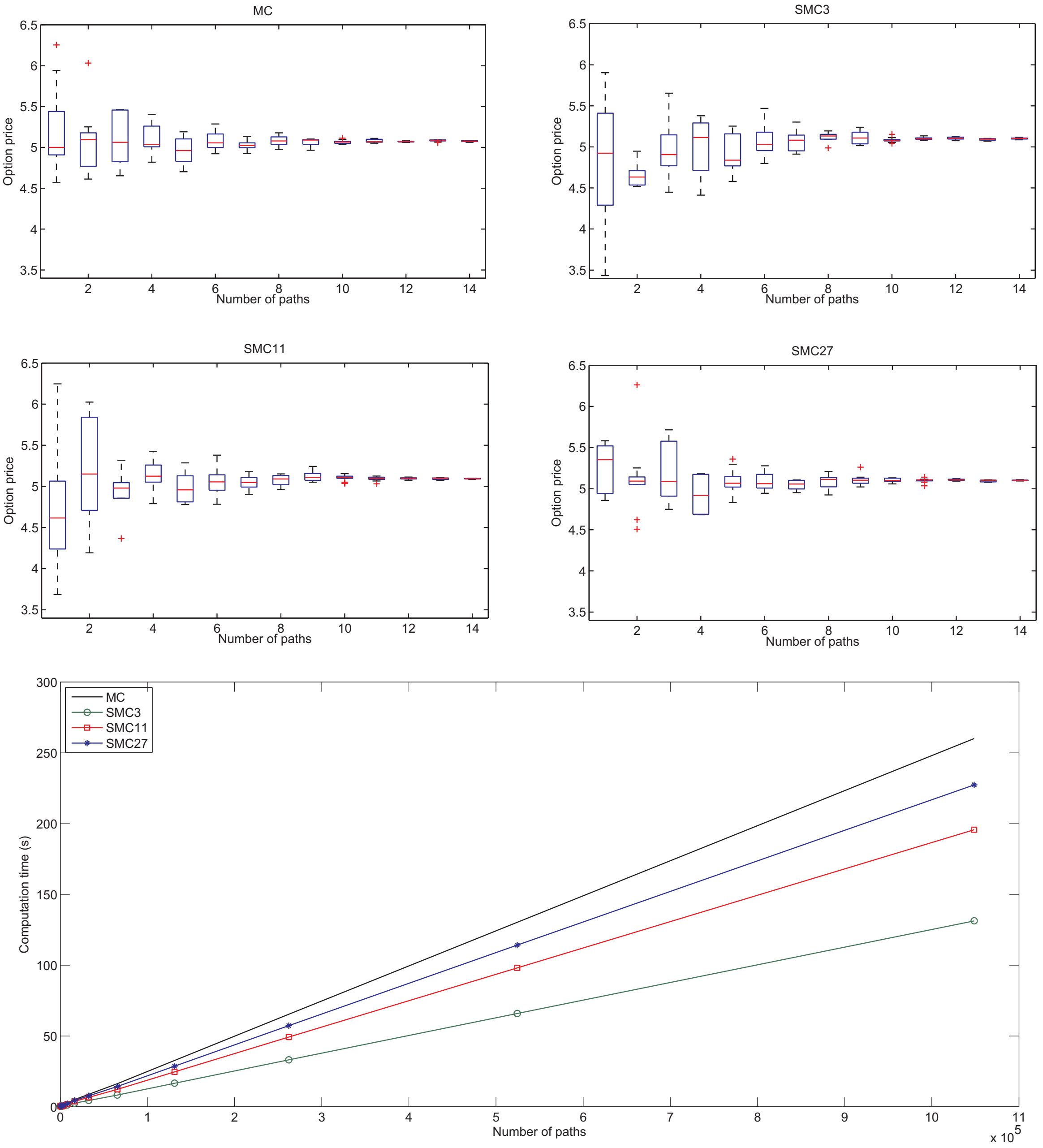


Figure 9.5: Top panels: Box plots for the MC and SMC implemementation of the Wasabi option pricing problem. Bottom panel: Computation time (s) for the MC and SMC implemementation of the Wasabi option pricing problem.

9.5 Conclusion and future work

In this study we proposed a novel and simple way to fasten MC simulation by recycling a fraction of generated random numbers for other trajectories under construction. We built this idea into a GPGPU implementation called *Shuffled Monte-Carlo* and study two factors: the pattern and period of shuffling. We elected to implement a butterfly XOR swap pattern based on its ease of implementation and score on causality test. The results both in terms of speed and accuracy were demonstrated to be satisfying enough to motivate this study.

Future study should concentrate on proposing a heterogeneous implementation of our SMC technique. We feel that two possible heterogeneous architectures may improve the current SMC implementation but ultimately should be empirically demonstrated. The most straightforward and easy to implement would be to decide a ratio r based on $CPU/GPU + CPU$ computational throughput and distribute rM paths to the CPU and $(1 - r)M$ to the GPU. No synchronization and no memory transfer occur until kernel completion. The second architecture to be tested, would introduce a third phase in the path building: a load phase. The CPU would be focused on filling multiple buffers with RNbs, and triggering asynchronous memory transfer onto the GPU. Since the memory engine can execute load instructions in parallel with the computation engine on recent GPUs, we feel that some improvements are still available for the taking. It would also reduce the amount of correlation introduced between paths. However we are still very conscious of the fine tuning that it might require regarding synchronization points between CPU and GPU.

Another axis of study finally concerns the shuffling patterns available. We considered here only the most direct ones, but one may be more curious and takes on the study more exotic shuffling, and their impact on accuracy and speed. However deeper theoretical results regarding the connection between the pattern/period of shuffling, and the correlation introduced seems to us problem hardly tractable as of now.

Chapter 10

Conclusion

10.1 Summary of the thesis

The present work has been focusing on the problem of exotic options pricing, where both the theoretical and the implementation sides of the problem have been considered.

The theoretical part of this thesis occupies chapter 2 throughout to chapter 6, and has been devoted to providing new formula for some quite complex financial derivative instruments. The derivation of those formula using econophysics as a working framework would serve to prove the relevance of econophysics methods along with the more traditional probabilistic angle of attack.
Chapter 2 was devolved to introducing the necessary theoretical concepts connected to the calculus of stochastic processes that will be used to model financial risky assets; the critically important properties of the Wiener process/Brownian motion were introduced.
In chapter 3 a background on the financial market was provided, broke down and we singled out the financial derivatives as our target of study. Complex derivatives known as exotic options were presented and the problem of option pricing was discussed for it is the main target of the present thesis. The Black-Scholes model being the model in which our study takes place was introduced in the same chapter.
The principal tool we used throughout the theoretical part of this thesis in order to address the option pricing problem was the path integral method as originally developed by Weiner and Feynman. Therefore an introduction to this technical framework was provided in chapter 4, along with some equivalent formulation of path integral equations and partial differential equations. Since path integration is a tool that has been greatly championed by quantum physicists, the path integral formulation of quantum mechanics is also discussed in the same chapter.
We improved over an existing pricing formula for the outside barrier Asian option in chapter 5. Using path integrals we solved the problem and benchmarked the accuracy of our solution, it was found to reduce the error when compared to previous study[DLT10]. We did not limit ourself to improving over the existing formula but studied the problem of adding

another barrier condition to the payoff structure. We wrote the path integral representation of this pricing problem and the connection with the physics problem of a particle in a box was drawn. We solved it , the solution taking the form of a sum of normal cumulative distribution functions. The solution accuracy was benchmarked, with respect to both correlation and initial asset proximity to barriers, and was found to fall well within acceptable range.
The path integral framework was fully exploited in the following chapter, no.6, where we studied the problem of the Parisian option. We showed how it related to a path integral describing the motion of a particle in a potential, a problem rather common in the physics field. Once we wrote the path integral representation of our pricing problem, we used the Feynman-Kac formula to work with the equivalent partial differential equation and solve it. Benchmarking was found to yield accurate results while the formula did not exhibit penalizing computational cost. Using the newly derived path integral results, we proposed then a new type of option that we baptized “Wasabi option” that builds on top of the Parisian option. As we mentioned that we do not aim at deriving formula unsuitable for implementation, we argued a method to simplify the pricing problem by introducing a second process into the system. We studied its accuracy and found an efficient if inelegant way to reduce the error margin. In the end the error was found to be nothing short of negligible.

The implementation part of our thesis covers chapter 7,8 and 9.
The necessary technical background covering general purpose computing on graphic processing units (GPGPU) was introduced in chapter 7. We started by the most general consideration regarding distribution computation and parallelizable problem, before describing the general hardware architecture of a GPU card. The single program multiple data model which is the one used when programming for GPU is introduced along with the CUDA programming platform. We then discuss in more detail how CUDA handle the distribution of computational workload onto the GPU with a case study. Taking a basket option as the target we benchmarked various heterogeneous architectures to select a superior one. Finally in the last chapter we proposed an original Monte Carlo scheme targeting the GPU with the goal to improve its running time, when compared with the traditional Monte Carlo simulation.

10.2 Did we reach our goal ?

We recall the goal that we set ourselves in the present thesis and our driving motivation: to exhibit sufficient proof that beyond the usual probabilistic/stochastic approach to option pricing there was room for another interdisciplinary approach. We argued that using insights from physics we could solve pricing problems in an intuitive fashion and derive formula that led themselves to efficient implementation. We also aimed at improving or proposing original implementation of pricing formula with a special consideration for speed.

How well did we succeed on those two fronts?

- **Demonstrate Econophysics methods adequacy to the exotic option pricing problem**
  - Improved the results for simple outside barrier Asian options
  - Derived an original result for the double outside barrier Asian options
  - Derived the propagator for the Parisian option
  - Designed a Wasabi product and priced it

The original work we conducted in chapters 5 and 6 have been both derived using path integral techniques as found in the physics fields, more especially in quantum physics. Even though the original was rather abstract and could easily have led to the common obtuse formula found in mathematical publication, we used intuition and straightforward physics consideration to reach our results. The formula derived were either a finite sum of plain normal cumulative distribution function, or quickly converging sum of similar terms.
Therefore in regards to putting forward enough evidence that econophysics and path integral methods specifically were well adapted to the study of the option pricing problem, we think that our original results are an unquestionable proof. On the most exotic products, the Wasabi option, that we devised in order to test how far we could go while keeping an easy-to-implement formula, we feel that arguments could be made either way. It is ultimately to the reader to decide if a term that is a numerical integration of the error function is a costly operation, then one should decide if summing four of those terms is arbitrary complex. We think that this is well within the acceptable computational cost of useful pricing formula.

- **Propose optimized GPGPU implementation**
  - Case studied the problem of heterogeneous CPU/GPU distribution of computation workload
  - Designed an original and fast variation “shuffled Monte Carlo” on traditional Monte Carlo simulation

---

The empirical study we conducted on a heterogeneous CPU/GPU target architecture is an alternative that is receiving as of today too few attention. We however, in this study, optimized both CPU and GPU and had them working together to yield a faster answer than both a pure CPU and a pure GPU solution. We also want to point out that on the contrary to a lot of implementation papers, we did not chose a vanilla product as of our scope of study. By selecting a formula that is not straightforward to implement we feel that the claims we make can not be argued on a relevancy ground. To discuss technical points: we optimized the numerical scheme used with respects to GPU constraints, we limited the amount of hard coded parameters that is often used in order to yield faster results, at the detriment of re-usability, by doing the load balancing in an online fashion. The results we obtained justified the design of an heterogenous solution by exhibiting faster results than our other benchmarked implementation.

Then we moved on to accelerate Monte Carlo simulation. That could seem paradoxical since we claimed we used econophysics as a way to derive analytical formula that we could study and implement fairly easily. Yet, when one derive a new formula and would like to test its validity, Monte Carlo simulation seems to be the most efficient way to do so. Therefore we see Monte Carlo simulation as a test to pass in order to justify validity of our results. This is the reason why we aimed at also improving computation time for Monte Carlo simulation. By proposing an original recycling technique to reduce the generation of random numbers on GPU via shuffling, we could reduce the necessary computation time from 12% to 50%.

In the end, we think that the various techniques we proposed on the implementation side fulfilled our goal of studying design past beyond the tradition CPU vs. GPU , and also at proposing powerful technique to reduce the computation time.

Appendices

Appendix A

Propagator for two coupled wiener processes with drift using change of measure

In this section we derive the propagator for a system of coupled drifting wiener processes using a change of measure argument to reduce the problem to uncoupled and driftless processes. Starting with the following SDEs under probability measure $\mathbb{P}$.

$$\begin{aligned} dX_T &= \mu_X dt + \sigma_X dW_{X,T} \\ dY_T &= \mu_Y dt + \sigma_Y \left(\rho dW_{X,T} + \sqrt{1-\rho^2} dW_{Y,T}\right) \end{aligned} \tag{A.1}$$

After applying the following transformation to X_T and Y_T

$$\begin{bmatrix} d\tilde{X}_T \\ d\tilde{Y}_T \end{bmatrix} = \begin{bmatrix} \frac{1}{\sigma_X} & 0 \\ \frac{-\rho}{\sigma_X\sqrt{1-\rho^2}} & \frac{1}{\sigma_Y\sqrt{1-\rho^2}} \end{bmatrix} \begin{bmatrix} dX_T \\ dY_T \end{bmatrix}$$

we are left with

$$\begin{aligned} d\tilde{X}_T &= \frac{\mu_X}{\sigma_X} dt + dW_{X,T} \\ d\tilde{Y}_T &= \left(\frac{\mu_Y \sigma_X - \rho \mu_X \sigma_Y}{\sigma_Y \sigma_X \sqrt{1-\rho^2}}\right) dt + dW_{Y,T} \end{aligned} \tag{A.2}$$

Now we define a new probability measure $\mathbb{Q}_X$, where $\mathbb{Q}_X \ll \mathbb{P}$ and $\mathbb{P} \ll \mathbb{Q}_X$, by its Radon-Nikodym derivative

$$\left.\frac{d\mathbb{Q}_X}{d\mathbb{P}}\right|_{\mathcal{F}_T} := e^{\frac{-T}{2}\left(\frac{\mu_X}{\sigma_X}\right)^2 - \left(\frac{\mu_X}{\sigma_X}\right)W_{X,T}} \tag{A.3}$$

similarly for $\mathbb{Q}_Y$

$$\left.\frac{d\mathbb{Q}_Y}{d\mathbb{P}}\right|_{\mathcal{F}_T} := e^{\frac{-T}{2}\left(\frac{\mu_Y \sigma_X - \rho \mu_X \sigma_Y}{\sigma_Y \sigma_X \sqrt{1-\rho^2}}\right)^2 - \left(\frac{\mu_Y \sigma_X - \rho \mu_X \sigma_Y}{\sigma_Y \sigma_X \sqrt{1-\rho^2}}\right)W_{Y,T}} \tag{A.4}$$

where $\mathcal{F}_T$ is the natural filtration for $[X_T, Y_T]$.
Now from Cameron-Martin-Girsanov theorem we know that

$$d\tilde{X}_T = \frac{\mu_X}{\sigma_X}dt + dW_{X,T} = d\tilde{W}_{X,T} \tag{A.5}$$

where

$$d\tilde{W}_{X,T} = dW_{X,T} + \left(\frac{\mu_X}{\sigma_X}\right)dt \tag{A.6}$$

is a standard Wiener process under $\mathbb{Q}_X$. Similarly for $\tilde{Y}_T$

$$d\tilde{Y}_T = \left(\frac{\mu_Y\sigma_X - \rho\mu_X\sigma_Y}{\sigma_Y\sigma_X\sqrt{1-\rho^2}}\right)dt + dW_{Y,T} = d\tilde{W}_{Y,T} \tag{A.7}$$

with

$$d\tilde{W}_{Y,T} = dW_{Y,T} + \left(\frac{\mu_Y\sigma_X - \rho\mu_X\sigma_Y}{\sigma_Y\sigma_X\sqrt{1-\rho^2}}\right)dt \tag{A.8}$$

a standard Wiener process under $\mathbb{Q}_Y$. Since $\tilde{X}_T$ and $\tilde{Y}_T$ are now driftless and uncorrelated it is direct that

$$\begin{aligned} W_{\mathbb{Q}_X}\left(\tilde{X}_T|0\right) &= \frac{e^{\frac{-\tilde{X}_T^2}{2T}}}{\sqrt{2\pi T}} \\ W_{\mathbb{Q}_Y}\left(\tilde{Y}_T|0\right) &= \frac{e^{\frac{-\tilde{Y}_T^2}{2T}}}{\sqrt{2\pi T}} \end{aligned} \tag{A.9}$$

The original transition probability for X_T and Y_T under $\mathbb{P}$ is retrieved by first multiplying through with the inverse Radon-Nikodym derivatives, where we used the fact that our process has joint Gaussian distribution to justify the leap from uncorrelatedness to independence

$$W_{\mathbb{P}}\left(\tilde{X}_T, \tilde{Y}_T|0,0\right) = W_{\mathbb{Q}_X}\left(\tilde{X}_T|0\right)\left.\frac{d\mathbb{P}}{d\mathbb{Q}_X}\right|_{\mathcal{F}_T} W_{\mathbb{Q}_Y}\left(\tilde{Y}_T|0\right)\left.\frac{d\mathbb{P}}{d\mathbb{Q}_Y}\right|_{\mathcal{F}_T} \tag{A.10}$$

then reverting back to the original variables

$$\begin{bmatrix} X_T \\ Y_T \end{bmatrix} = \begin{bmatrix} \sigma_X & 0 \\ \sigma_Y\rho & \sigma_Y\left(1-\rho^2\right) \end{bmatrix}\begin{bmatrix} \tilde{X}_T \\ \tilde{Y}_T \end{bmatrix}$$

leads to the usual result where we dropped the understood $\mathbb{P}$ measure subscript

$$\begin{aligned} W\left(X_T, Y_T|0,0\right) = \frac{1}{2\pi\sigma_X\sigma_Y T\sqrt{1-\rho^2}}\exp\Bigg(\frac{-1}{2(1-\rho^2)}\Bigg[\left(\frac{X_T-\mu_X T}{\sigma_X\sqrt{T}}\right)^2 \\ + \left(\frac{Y_T-\mu_Y T}{\sigma_Y\sqrt{T}}\right)^2 - \frac{2\rho\left(X_T-\mu_X T\right)\left(Y_T-\mu_Y T\right)}{\sigma_X\sigma_Y T}\Bigg]\Bigg) \end{aligned} \tag{A.11}$$

Appendix B

Distribution of $\bar{X}_T$

In this appendix we derive the distribution of the continuously monitored geometric average

$$\bar{X}_T := \frac{1}{T}\int_0^T X_t dt \tag{B.1}$$

where

$$X_t = \mu t + \sigma \int_0^t d_W t \tag{B.2}$$

is normally distributed $\mathcal{N}\left(\mu T, \sigma^2 T\right)$. Starting with the expectation

$$\begin{aligned}
\mathbb{E}\left[\bar{X}_T\right] &= \mathbb{E}\left[\frac{1}{T}\int_0^T \left(\mu t + \sigma W_t\right) dt\right] \\
&= \frac{\mu T}{2} + \frac{\sigma}{T}\mathbb{E}\left[\int_0^T W_t dt\right] \\
&= \frac{\mu T}{2} + \frac{\sigma}{T}\mathbb{E}\left[\int_0^T \left(T - t\right) d_W t\right] \\
&= \frac{\mu T}{2}
\end{aligned} \tag{B.3}$$

where the last equality follows by null expectation of the Ito Integral. We now move on to finding the variance of $\bar{X}_T$

$$\begin{aligned}
\text{Var}\left[\bar{X}_T\right] &= \frac{\sigma^2}{T^2}\text{Var}\left[\int_0^T W_t dt\right] \\
&= \frac{\sigma^2}{T^2}\mathbb{E}\left[\int_0^T \left(T - t\right)^2 dt\right] \\
&= \frac{\sigma^2 T}{3}
\end{aligned} \tag{B.4}$$

where Ito isometry was used between the first and second line.
Let us point out here that

$$\mathbb{E}\left[\bar{X}_T\right] = \frac{\mathbb{E}\left[X_T\right]}{2}, \mathrm{Var}\left[\bar{X}_T\right] = \frac{\mathrm{Var}\left[X_T\right]}{3} \tag{B.5}$$

Since $\bar{X}_T$ is a linear functional of a Gaussian process, only the first two moments are necessary. Therefore the distribution of $\bar{X}_T$ is now fully characterized.

Appendix C

Simplification of propagator integrals appearing in Wasabi options

In this appendix we give an equivalent form to the definite integrals in the propagator (6.18) expression as mere evaluation of $\operatorname{erfc}(\cdot)$, computationally less costly than relying on numerical integration schemes. In order to do so we will make repetitive use of the following definite integral equality for $\delta > 0$

$$\begin{aligned}&\int_0^{+\infty} \left(\alpha x^2 + \beta x + \gamma\right) e^{-\left(\delta x^2 + \mu x + \eta\right)}\, dx \\ &= \frac{e^{-\eta}}{8\delta^{5/2}} \left[2\sqrt{\delta}(2\beta\delta - \alpha\mu) + \sqrt{\pi} e^{\frac{\mu^2}{4\delta}} \operatorname{erfc}\left(\frac{\mu}{2\sqrt{\delta}} \right) \left(2\delta(\alpha + 2\gamma\delta) + \alpha\mu^2 - 2\beta\delta\mu \right) \right]\end{aligned} \tag{C.1}$$

First we look at the following integral

$$\int_0^{\infty} \frac{\left(y + \frac{b}{\sigma}\right)\left(y + \frac{X_T - b}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left(-\frac{\left(y + \frac{b}{\sigma}\right)^2}{2(T-\lambda)} - \frac{\left(y + \frac{X_T - b}{\sigma}\right)^2}{2\lambda} \right) dy \tag{C.2}$$

appearing in the propagator expression (6.18) for $X_T > b$ is then equal to

$$\frac{e^{-\frac{(X_T-2b)^2}{2\lambda\sigma^2}}}{\sqrt{2}\pi\sigma^3 T^{5/2}} \left[\sigma(2b - X_T)\sqrt{\frac{2\lambda T}{T-\lambda}} - \sqrt{\pi}\left((X_T - 2b)^2 - \sigma^2 T \right) e^{\frac{(X_T-2b)^2(T-\lambda)}{2\lambda\sigma^2 T}} \operatorname{erfc}\left(\frac{2b - X_T}{\sigma\sqrt{\frac{2\lambda T}{T-\lambda}}} \right) \right] \tag{C.3}$$

Similarly the integral

$$\int_0^{\infty} \frac{y\left(y + \frac{2b - X_T}{\sigma}\right)}{\sigma\pi\sqrt{\lambda^3(T-\lambda)^3}} \exp\left(-\frac{y^2}{2(T-\lambda)} - \frac{\left(y + \frac{2b - X_T}{\sigma}\right)^2}{2\lambda} \right) dy \tag{C.4}$$

appearing in the propagator expression (6.18) for $X_T < b$ is found to be equal to

$$
\begin{aligned}
&\frac{(b-X_T)(-b\lambda+3bT-2TX_T+\lambda X_T)+\lambda\sigma^2 T}{\sqrt{2\pi}\lambda\sigma^3 T^{5/2}}\exp\left(\frac{b(2X_T-3b)}{2\lambda\sigma^2}-\frac{(b-X_T)^2}{2\sigma^2 T}\right)\\
&\times \operatorname{erfc}\left(\frac{-b\lambda+2bT-TX_T+\lambda X_T}{\sigma\sqrt{2\lambda T(T-\lambda)}}\right)-\frac{(b-X_T)\,(2T-\lambda)}{\pi\sigma^2T^2\sqrt{\lambda(T-\lambda)}}\exp\left(\frac{(b-X_T)^2}{2\lambda\sigma^2}-\frac{b^2}{2\sigma^2(T-\lambda)}\right)
\end{aligned}
\tag{C.5}
$$

therefore the expression for the propagator (6.18) can be rewritten under the equivalent form

$$
\begin{aligned}
&W_P\left(X_T,T,\lambda|0,0\right)=\Theta\left(b-X_T\right)\left[\frac{(b-X_T)(-b\lambda+3bT-2TX_T+\lambda X_T)+\lambda\sigma^2 T}{\sqrt{2\pi}\lambda\sigma^3T^{5/2}}\right.\\
&\exp\left[\frac{b(2X_T-3b)}{2\lambda\sigma^2}-\frac{(b-X_T)^2}{2\sigma^2T}\right]\operatorname{erfc}\left(\frac{-b\lambda+2bT-TX_T+\lambda X_T}{\sigma\sqrt{2\lambda T(T-\lambda)}}\right)-\frac{(b-X_T)\,(2T-\lambda)}{\pi\sigma^2T^2\sqrt{\lambda(T-\lambda)}}\\
&\left.\times\exp\left[\frac{(b-X_T)^2}{2\lambda\sigma^2}-\frac{b^2}{2\sigma^2(T-\lambda)}\right]\right]+\Theta\left(X_T-b\right)\left[\frac{\exp\left[-\frac{(X_T-2b)^2}{2\lambda\sigma^2}\right]}{\sqrt{2}\pi\sigma^3T^{5/2}}\left(\sigma\left(2b-X_T\right)\sqrt{\frac{2\lambda T}{T-\lambda}}\right.\right.\\
&\left.\left.-\sqrt{\pi}\left((X_T-2b)^2-\sigma^2T\right)\exp\left[\frac{(X_T-2b)^2(T-\lambda)}{2\lambda\sigma^2T}\right]\operatorname{erfc}\left(\frac{2b-X_T}{\sigma\sqrt{\frac{2\lambda T}{T-\lambda}}}\right)\right)\right]
\end{aligned}
\tag{C.6}
$$

References

[All07] E. Allen. *Modeling with Itô Stochastic Differential Equations.* Mathematical Modelling: Theory and Applications. Springer, 2007.

[BA05] Giovanni Barone-Adesi. The saga of the american put. *Journal of Banking and Finance*, 29(11):2909 – 2918, 2005.

[Baa04] B.E. Baaquie. *Quantum Finance.* Cambridge University Press, 2004.

[Bel97] Belal E. Baaquie. A path integral approach to option pricing with stochastic volatility: Some exact results. *J. Phys. I France*, 7(12):1733–1753, 1997.

[BKS00] B. E. Baaquie, L. C. Kwek, and M. Srikant. Simulation of Stochastic Volatility using Path Integration: Smiles and Frowns. *eprint arXiv:cond-mat/0008327*, August 2000.

[BMMN06] G. Bormetti, G. Montagna, N. Moreni, and O. Nicrosini. Pricing exotic options in a path integral approach. *Quantitative Finance*, 6(1):55–66, 2006.

[Boa08] OpenMP Architecture Review Board. Openmp application program interface version 3.0, May 2008.

[Boy01] John P. Boyd. *Chebyshev and Fourier Spectral Methods.* Dover Books on Mathematics, 2001.

[BS73] Fischer Black and Myron S Scholes. The pricing of options and corporate liabilities. *Journal of Political Economy*, 81, 1973.

[BY03] Mark Broadie and Yusaku Yamamoto. Application of the fast gauss transform to option pricing. *Manage. Sci.*, 49(8):1071–1088, 2003.

[BZ12] Cornelis W. Bowen Zhang, Oosterlee. Acceleration of option pricing technique on graphics processing units. *Concurrency and Computation: Practice and Experience*, 2012.

[CCO14a] Aurelien Cassagnes, Yu Chen, and Hirotada Ohashi. Path integral pricing of outside barrier asian options. *Physica A: Statistical Mechanics and its Applications*, 394(0):266 – 276, 2014.

[CCO14b] Aurelien Cassagnes, Yu Chen, and Hirotada Ohashi. Path integral pricing of wasabi option in the black-scholes model. *Physica A: Statistical Mechanics and its Applications*, 413(0):1 – 10, 2014.

[CJPY97] Marc Chesney, Monique Jeanblanc-Picqué, and Marc Yor. Brownian excursions and parisian barrier options. *Advances in Applied Probability*, pages 165–184, 1997.

[CLM97] J.Y. Campbell, A.W.C. Lo, and A.C. MacKinlay. *The Econometrics of Financial Markets*. Princeton University Press, 1997.

[CMS99] Peter Carr, Dilip B. Madan, and Robert H Smith. Option valuation using the fast fourier transform. *Journal of Computational Finance*, 2:61–73, 1999.

[Coh13] D.L. Cohn. *Measure Theory: Second Edition*. Birkhäuser Advanced Texts Basler Lehrbücher. Springer New York, 2013.

[Cox75] J. Cox. Notes on option pricing i: Constant elasticity of diffusions., 1975.

[CPVR10] Mauricio Contreras, Rely Pellicer, Marcelo Villena, and Aaron Ruiz. A quantum model of option pricing: When black–scholes meets schrödinger and its semi-classical limit. *Physica A: Statistical Mechanics and its Applications*, 389(23):5447 – 5459, 2010.

[CRR79] John C. Cox, Stephen A. Ross, and Mark Rubinstein. Option pricing: A simplified approach. *Journal of Financial Economics*, pages 229–263, 1979.

[Den] Peter J. Denning. The locality principle. *Commun. ACM*.

[DGM09] Daniel Dufresne, Jose Garrido, and Manuel Morales. Fourier inversion formulas in option pricing and insurance. Technical Report 3, 2009.

[DLT10] J.P.A. Devreese, D. Lemmens, and J. Tempere. Path integral approach to asian options in the black–scholes model. *Physica A: Statistical Mechanics and its Applications*, 389(4):780 – 788, 2010.

[DW10] Kirk David and Hwu Wen, mei. *Programming Massively Parallel Processors: A Hands-on Approach*. Morgan Kaufmann, 1st edition, 2010.

[EBT99] Marco Rosa-Clot Eleonora Bennati and Stefano Taddei. A path integral approach to derivative security pricing: I. formalism and analytical results. *International Journal of Theoretical and Applied Finance*, 1999.

[Ein05] A. Einstein. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, 322:549–560, 1905.

[Eva98] L.C. Evans. *Partial Differential Equations*. Graduate studies in mathematics. American Mathematical Society, 1998.

[FH12] R.P. Feynman and A.R. Hibbs. *Quantum Mechanics and Path Integrals: Emended Edition*. Dover Publications, Incorporated, 2012.

[FO08] Fang Fang and Cornelis W. Oosterlee. A novel pricing method for european options based on fourier-cosine series expansions. *SIAM J. Sci. Comput.*, 31(2):826–848, 2008.

[GHK$^+$11] Benedict R. Gaster, Lee Howes, David R. Kaeli, Perhaad Mistry, and Dana Schaa. *Heterogeneous Computing with OpenCL*. 2011.

[Gla04] P. Glasserman. *Monte Carlo Methods in Financial Engineering*. Applications of mathematics : stochastic modelling and applied probability. Springer, 2004.

[Goo81] Mark Goodman. Path integral solution to the infinite square well. *American Journal of Physics*, 49(9):843–847, 1981.

[Gra69] C. W. J. Granger. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica*, 37(3):pp. 424–438, 1969.

[Gri05] D.J. Griffiths. *Introduction to Quantum Mechanics*. Pearson international edition. Pearson Prentice Hall, 2005.

[GY60] I. M. Gelfand and A. M. Yaglom. Integration in functional spaces and its applications in quantum physics. *Journal of Mathematical Physics*, 1(1):48–69, 1960.

[Hav02] Emmanuel E Haven. A discussion on embedding the black–scholes option pricing model in a quantum physics setting. *Physica A: Statistical Mechanics and its Applications*, 304(3–4):507 – 524, 2002.

[Hes93] SL Heston. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Review of Financial Studies*, 6(2):327–343, 1993.

[Hul08] J.C. Hull. *Options, Futures, and Other Derivatives*. Pearson Prentice Hall, 7th edition, 2008.

[Ito44] Kiyoshi Ito. Stochastic integral. *Proceedings of the Imperial Academy*, 20(8):519–524, 1944.

[J.D88] J.Dash. Path integrals and options - i. CNRS preprint, 1988.

[J.D89] J.Dash. Path integrals and options - ii. CNRS preprint, 1989.

[JYC09] M. Jeanblanc, M. Yor, and M. Chesney. *Mathematical Methods for Financial Markets*. Springer Finance. Springer, 2009.

[Kac49] M. Kac. On distributions of certain wiener functionals. *Transactions of the American Mathematical Society*, 65(1):pp. 1–13, 1949.

[Kac66] M. Kac. Wiener and integration in function spaces. *American Mathematical Society. Bulletin. New Series*, (72), 1966.

[KdKKM04] M. Krekel, J. de Kock, R. Korn, and T. K. Man. An analysis of pricing methods for basket options. *Wilmott magazine*, pages 82–89, 2004.

[Kle09] H. Kleinert. *Path Integrals in Quantum Mechanics, Statistics, Polymer Physics, and Financial Markets.* EBL-Schweitzer. World Scientific Publishing Company, Incorporated, 2009.

[KS91] I. Karatzas and S.E. Shreve. *Brownian Motion and Stochastic Calculus.* Graduate Texts in Mathematics. Springer New York, 1991.

[KV90] A.G.Z. Kemna and A.C.F. Vorst. A pricing method for options based on average asset values. *Journal of Banking and Finance*, 14(1):113 – 129, 1990.

[Kwo08] Yue-Kuen Kwok. *Mathematical Models of Financial Derivatives.* Springer, 2008.

[Lin98] Vadim Linetsky. The path integral approach to financial modeling and options pricing. *Computational Economics*, 11(1-2):129–63, 1998.

[LKC$^+$10] Victor W. Lee, Changkyu Kim, Jatin Chhugani, Michael Deisher, Daehyun Kim, Anthony D. Nguyen, Nadathur Satish, Mikhail Smelyanskiy, Srinivas Chennupaty, Per Hammarlund, Ronak Singhal, and Pradeep Dubey. Debunking the 100x gpu vs. cpu myth: An evaluation of throughput computing on cpu and gpu. *SIGARCH Comput. Archit. News*, 38(3):451–460, June 2010.

[LL09] CÉLINE LABART and JÉRÔME LELONG. Pricing double barrier parisian options using laplace transforms. *International Journal of Theoretical and Applied Finance*, 12(01):19–44, 2009.

[LLT11] L. Z. J. Liang, D. Lemmens, and J. Tempere. Path integral approach to the pricing of timer options with the duru-kleinert time transformation. *Phys. Rev. E*, 83:056112, May 2011.

[LS01] FA Longstaff and ES Schwartz. Valuing american options by simulation: a simple least-squares approach. *Review of Financial Studies*, 14(1):113–147, 2001.

[LSLM10] Yongchao Liu, Bertil Schmidt, Weiguo Liu, and Douglas L. Maskell. Cudameme: Accelerating motif discovery in biological sequences using cuda-enabled graphics processing units. *Pattern Recogn. Lett.*, 31(14):2170–2177, 2010.

[Man63] Benoit Mandelbrot. The Variation of Certain Speculative Prices. *The Journal of Business*, 36:394, 1963.

[Mar03] George Marsaglia. Xorshift rngs. *Journal of Statistical Software*, 8(14):1–6, 7 2003.

[Mat00] A. Matacz. Path Dependent Option Pricing: the path integral partial averaging method. *eprint arXiv:cond-mat/0005319*, May 2000.

[MC01] A. Demichev M. Chaichian. *Path Integrals in Physics: Volume I Stochastic Processes and Quantum Mechanics.* Taylor and Francis, 2001.

[MNM02] Guido Montagna, Oreste Nicrosini, and Nicola Moreni. A path integral way to option pricing. *Physica A: Statistical Mechanics and its Applications*, 310(3–4):450 – 466, 2002.

[MS95] Rosario N. Mantegna and H. Eugene Stanley. Scaling behaviour in the dynamics of an economic index. *Nature*, 376(6535):46–49, 1995.

[NVI] NVIDIA. Cuda toolkit.

[oIS14] Bank of International Settlments. Derivatives statistics, 2014.

[Øks03] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications.* Hochschultext / Universitext. U.S. Government Printing Office, 2003.

[Pas11] Andrea Pascucci. *PDE and Martingale Methods in Option Pricing.* Springer, 2011.

[PDY10] Victor B. Putz, Jarn Dunkel, and Julia M. Yeomans. Cuda simulations of active dumbbell suspensions. *Chemical Physics*, 375, 2010.

[PGLD11] Ying Peng, Bin Gong, Hui Liu, and Bin Dai. Option pricing on the gpu with backward stochastic differential equation. In *Parallel Architectures, Algorithms and Programming (PAAP), 2011 Fourth International Symposium on*, 2011.

[Pol01] A.D. Polyanin. *Handbook of Linear Partial Differential Equations for Engineers and Scientists.* Taylor & Francis, 2001.

[PP09] Bin PENG and Fei PENG. Pricing rainbow asian options. *Systems Engineering - Theory & Practice*, 29(11):76 – 83, 2009.

[Pre11a] T. Preis. Gpu-computing in econophysics and statistical physics. *The European Physical Journal Special Topics*, 194(1):87–119, 2011.

[Pre11b] T. Preis. Gpu-computing in econophysics and statistical physics. *The European Physical Journal Special Topics*, 194(1):87–119, 2011.

[Pro01] Philip Protter. A partial introduction to financial asset pricing theory. *Stochastic Processes and their Applications*, 91(2):169 – 203, 2001.

[RCT02] MARCO ROSA-CLOT and STEFANO TADDEI. A path integral approach to derivative security pricing ii: Numerical methods. *International Journal of Theoretical and Applied Finance*, 05(02):123–146, 2002.

[RO12] M. Ruijter and C. Oosterlee. Two-dimensional fourier cosine series expansion method for pricing financial options. *SIAM Journal on Scientific Computing*, 34(5):B642–B671, 2012.

[Sam65] Paul A Samuelson. Proof that properly anticipated prices fluctuate randomly. *Industrial management review*, 6(2):41–49, 1965.

[Shr04] Steven E. Shreve. *Stochastic calculus for finance. II.* Springer Finance. Springer-Verlag, 2004.

[Sim11] Jean-Guy Simonato. Computing american option prices in the lognormal jump–diffusion framework with a markov chain. *Finance Research Letters*, 8(4):220–226, 2011.

[SM13] Mutsuo Saito and Makoto Matsumoto. Variants of mersenne twister suitable for graphic processors. *ACM Trans. Math. Softw.*, 39(2):12:1–12:20, February 2013.

[SMDS11] John K. Salmon, Mark A. Moraes, Ron O. Dror, and David E. Shaw. Parallel random numbers: As easy as 1, 2, 3. In *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*, SC '11, pages 16:1–16:12, New York, NY, USA, 2011. ACM.

[Sor09] D. Sornette. *Why Stock Markets Crash: Critical Events in Complex Financial Systems.* Princeton University Press, 2009.

[TB10] Xiang Tian and Khaled Benkrid. High-performance quasi-monte carlo financial simulation: Fpga vs. gpp vs. gpu. *ACM Trans. Reconfigurable Technol. Syst.*, 3(4):26:1–26:22, 2010.

[Vec14] Jan Vecer. Asian options on the harmonic average. *Quantitative Finance*, 14(8):1315–1322, 2014.

[VSR06] Semyon V. Tsynkov Victor S. Ryaben'kii. *A Theoretical Introduction to Numerical Analysis.* Chapman and Hall/CRC, 1st edition, 2006.

[Wie21] Norbert Wiener. The average of an analytic functional and the brownian movement. *Proceedings of the National Academy of Sciences of the United States of America*, 7(10):pp. 294–298, 1921.

[Wie24] Norbert Wiener. The average value of a functional. *Proceedings of the London Mathematical Society*, s2-22(1):454–467, 1924.

[Zha95] Peter G. Zhang. A unified formula for outside barrier options. *Journal of financial engineering*, 4, 1995.

[Zhu06] Song-Ping Zhu. An exact and explicit solution for the valuation of american put options. *Quantitative Finance*, 6(3):229–242, 2006.

[ZO09] Bowen Zhang and Cornelis W. Oosterlee. Option pricing with cos method on graphics processing units. In *Parallel Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pages 1–8, 2009.