

**END OF STUDIES PROJECT REPORT IMAFA2 2013/2014**

**QuantLib 1.3**

**LIBRAIRIE QUANTLIB 1.3:** DESIGN AND IMPLEMENTATION OF NEW OPTIONS’ PRICING ENGINES.

|  |  |
| --- | --- |
| **Presented by:** | **Supervised by:** |
|  |  |
| Ahmed AYADI  Nathan D.KRUCK  Nolan POTIER | Mme. Anne Marie HUGUES  &  Mr. Luigi BALLABIO |
|  |  |

Index

[Chapter I Introduction 2](#_Toc376174635)

[Chapter II Barrier Options: 2](#_Toc376174636)

[I. Partial-Time Barrier Options 3](#_Toc376174637)

[Chapter III Netography 3](#_Toc376174638)

Table of Figures

[Table ‎II‑1Barrier Options versus Vanilla Options 3](#_Toc376174683)

# Introduction

**QuantLib** is a C++ library that eases writing applications for quantitative finance. In its turn, it uses the **Boost** library **[**[1](http://www.codeproject.com/Articles/4496/An-Introduction-to-Boost)**]**. Since its publication in November 2000, it has continued to grow in popularity in the financial sector. QuantLib is an open source project, it has many advantages. First of all, it is free. It is also thoroughly tested. This is due to the collective intelligence of all users and developers. All errors are quickly detected and corrected. Finally, it is easy to extend because the source code is freely available, the developer can learn from the implementation of its functionality. QuantLib is structured into modules [[2](http://quantlib.org/reference/modules.html)]. Each module covers a distinctive aspect of all features, and contains a set of components.

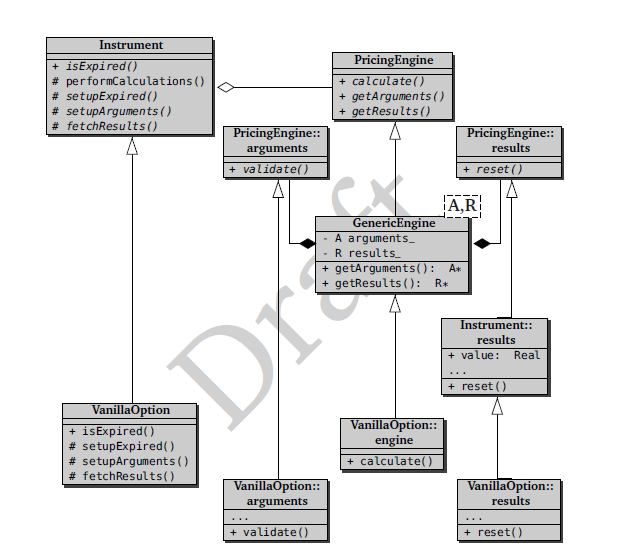
Our work involves the mastery of the use of existing pricing engines, in the first place and the design and development of others, in the second. So our planning for the project will cover a learning period including documentation reading and testing of pricers, then comes the other part of our solution implementation.

# QuantLib Architecture

The QuantLib architecture has been made to use it easily and a financial library must also provide developers with the means to extend it by adding new pricing functionality. For this it is necessary to know the 2 main element of the library.

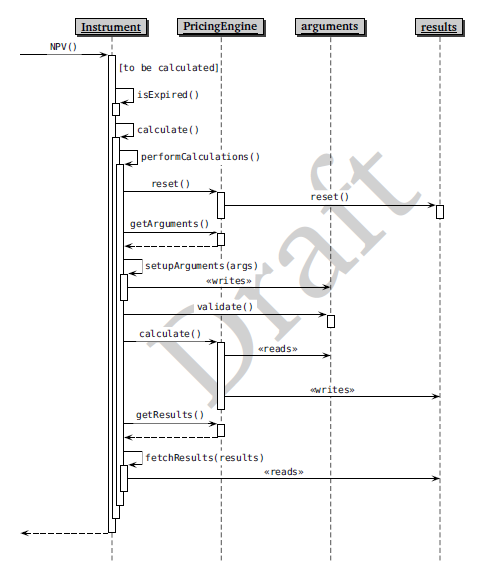
The « Instrument » element that allow to represent a financial instrument like a Vanilla Option with this different parameters (volatility, strike, maturity …). And the « Pricing Engine » element that allow to calculate the instrument values.

It’s necessary to separate this 2 element because for any given instrument, it is not always the case that a unique pricing method exists; moreover, one might want to use multiple methods for different reasons.



*Fig 1. Class diagram*

The figure 1 show the class diagram for the implementation of a Vanilla option.



*Fig. 2: Diagram sequences*

The figure 2 show the sequences exchanges between different objects to calculate the net present value (NPV) of an instrument.

# Barrier Options:

Barrier option is a path-dependent option where the price is based on the fluctuations in the underlying's value during all or part of the contract term. It was traded in the late 60s and it has been used to manage the risk. It comes in many flavors and forms, but its key characteristic is that this types of options is either initiated or exterminated upon reaching a certain barrier level; that is, it is either knocked in or knocked out.

“A basic American option is one type of path dependent option. Because it can be exercised at any time prior to expiration, its value will change as the underlying asset's value changes. An Asian option, also called an average option, is another type of path dependent option, because its payoff is based on the average price of the underlying asset during the contract term. Similarly, a barrier option would be considered a path dependent option because its value changes if the underlying asset reaches or surpasses a specified price. The lookback option and Russian option are also path-dependent options.”[[3](http://www.investopedia.com/terms/p/pathdependentoption.asp)]

In our work we consider the most basic type of barrier option; “**the single barrier**”. This option comes in 4 types where each type can be a Call or a Put option:

* Up & In
* Up & Out
* Down & In
* Down & Out

An "In" barrier option, for example, means that it becomes active once crossing the barrier level. If the underlying asset fails to cross the barrier, the Barrier Option you bought becomes worthless piece of paper upon expiration even if the underlying asset is trading above ( call option ) or below ( put option ), its strike price!

So it’s clear, at first thoughts, that Barrier Options are more dangerous than normal plain vanilla options. Then why would investors use them? Basically because they carry a much **lower extrinsic value** than plain vanilla options. [[5](http://www.optiontradingpedia.com/barrier_options.htm)]

There are various barriers types including Parisians, double barriers, and partial time barriers. Here, we discuss partial-time barrier options.

|  |  |  |
| --- | --- | --- |
|  | Advantage | Disadvantage |
| Knock-In Barrier Options | 1. Cheaper, resulting in higher profits 2. Ideal for speculating huge moves | 1. Higher risk of loss if underlying asset moves moderately 2. Commonly traded for forex, not stocks. |
| Knock-Out Barrier Options | 1. Cheaper, resulting in higher profits 2. Ideal for speculating small moves | 1. Higher risk of loss if underlying asset rallies 2. Commonly traded for forex, not stocks. |

Table ‎II‑1Barrier Options versus Vanilla Options

## Partial-Time Barrier Options

The watching period for a barrier crossing is restricted to a fraction of the option’s lifetime. Here we have two types: **partial-time-start** and **partial-time-end**.

The monitoring period of **Partial-time-start** barrier options starts at time zero and ends at a random date before expiration. As for **partial-time-end** barrier options, they have the monitoring period start at an arbitrary date before expiration and end at expiration.

There are two classes of **partial-time-end** barrier options [[4](http://hosho.ees.hokudai.ac.jp/~kubo/Rdoc/library/fExoticOptions/html/BarrierOptions.html)]: Type **B1** is defined such that only a barrier hit or crossed causes the option to be knocked out, and hence there is no difference between up and down options. Type **B2** options are defined such that a down-and-out call is knocked out as soon as the underlying price is below the barrier. Similarly, an up-and-out call is knocked out as soon as the underlying price is above the barrier.

# Netography

[1] Boost Library: <http://www.codeproject.com/Articles/4496/An-Introduction-to-Boost>

[2] QuantLib1.3 modules: <http://quantlib.org/reference/modules.html>

[3] <http://www.investopedia.com/terms/p/pathdependentoption.asp>

[4] <http://hosho.ees.hokudai.ac.jp/~kubo/Rdoc/library/fExoticOptions/html/BarrierOptions.html>

[5] <http://www.optiontradingpedia.com/barrier_options.htm>