
Interest Rate Models

Marco Marchioro
www.marchioro.org

October 6th, 2012

Details

- Università degli Studi di Milano-Bicocca
- Facoltà di Economia
- Corso di laurea Magistrale in Economia e Finanza
- Curriculum: Asset and Risk Management
- Class: Advanced Derivatives
- Module: Interest-Rate Models

Lecture details

- Every Saturday from October 1st to December 17th (2011)
- From 8:45 am to 11:30am (15 mins break)
- Usually in room U16-12
- Teaching-assistant sessions: not present, however, refer to Edit Rroji
- Official class language: English (Italian also available if **all** students agree)

Syllabus (1/2)

Introduction

- Introduction to exotic derivatives

Interest rate derivatives

- Linear interest-rate derivatives
- Bootstrap of interest-rate term structures
- Options on interest rates, caps/floors, and swaptions
- Basic interest-rate models
- Advanced interest-rate models
- Libor Market Model

Syllabus (2/2)

Credit derivatives

- Introduction to credit derivatives
- Instruments with embedded credit risk

Current topics

- Crisis of 2007: multi-curve bootstrapping

Class Requirements

- Basic calculus (same level, or less, required for Quantitative Finance)
- Basic computer literacy
- Excel 2003/2007/2010 on Windows (other OS's for advanced students only)
- Willingness to learn

Lecture style

- As close as possible to real-life quantitative finance
- Examples on spreadsheets for basic computations
- Students (called at the blackboard) teaching to students (sitting at their desks)
- Questions are not only welcome, they are encouraged

How to achieve higher marks

- Come to the lectures (encouraged, however, not technically mandatory)
- Do your homework and the exercises
- Ask those "trivial" questions
- Do not be overzealous with the teacher (proper attire required)
- Choose easy *"favorite subjects"*

Contacts

- www.marchioro.org
- marco (dot) marchioro (at) unimib.it
- e (dot) rroji (at) campus.unimib.it

Questions?

Introduction to exotic derivatives

(this lecture)

Lecture summary (1/2)

- A real-life examples of exotic derivatives
- Structure of a derivative termsheet
- Pricing tools and pricing software

Lecture summary (2/2)

- QuantLib: an open-source tool
- QuantLibXL: QuantLib on a spreadsheet
- Day counters, business calendars, and end-of-month conventions
- Examples using QuantLibXL

Example of an exotic derivative

- Prospectus, terms, and conditions
 - XS0505139120.pdf:
 - EUR Auto-callable Recovery Binary Up- Short DI put
- Input file for a risk-management system
 - customer-exotic-instruments.xml

Structure of a derivative termsheet (1/2)

- Legal details
- Contract title and summary
- Definition of parties involved
- Notionals, amounts, min/max shares
- Relevant dates

Structure of a derivative termsheet (2/2)

- Description of cash flows (deterministic or not)
- Reference to quote provider (e.g., Reuters)
- Risks taken with the instrument purchase
- More legal details
- Final legal notes

Questions?

Pricing tools and pricing software (1/2)

- In-house libraries (Matlab, C++)
 - Hard to use. Not standard. Highly customizable. Very expensive. Quick bug resolution
- Commercial libraries (NumeriX, FinCad)
 - Well documented. Can be expensive. Source code not available. Slow bug resolution

Pricing tools and pricing software (2/2)

- Open-source library (QuantLib)
 - Not-so-well documented. Highly customizable. Source code available. Quick bug resolution
- Data-provider terminals (Bloomberg)
 - Easy to use. Hard to customize. Only interactive (no batch)
- Commercial pricing services (StatPro, Markit)
 - Easy to use. Hard to customize. Moderately expensive

QuantLib

- Available since November 2000 (Ametrano, Ballabio, Marchioro)
- Completely free (as in no charge, no constraints)
- On average about 15,000 downloads each year
- Used all-over the world (five continents)
- Implements the *standard* financial models
- Advanced and experimental implementations
- Great Excel interface (QuantLibXL)
- Object-oriented library

Object-oriented programming

From Wikipedia, the free encyclopedia

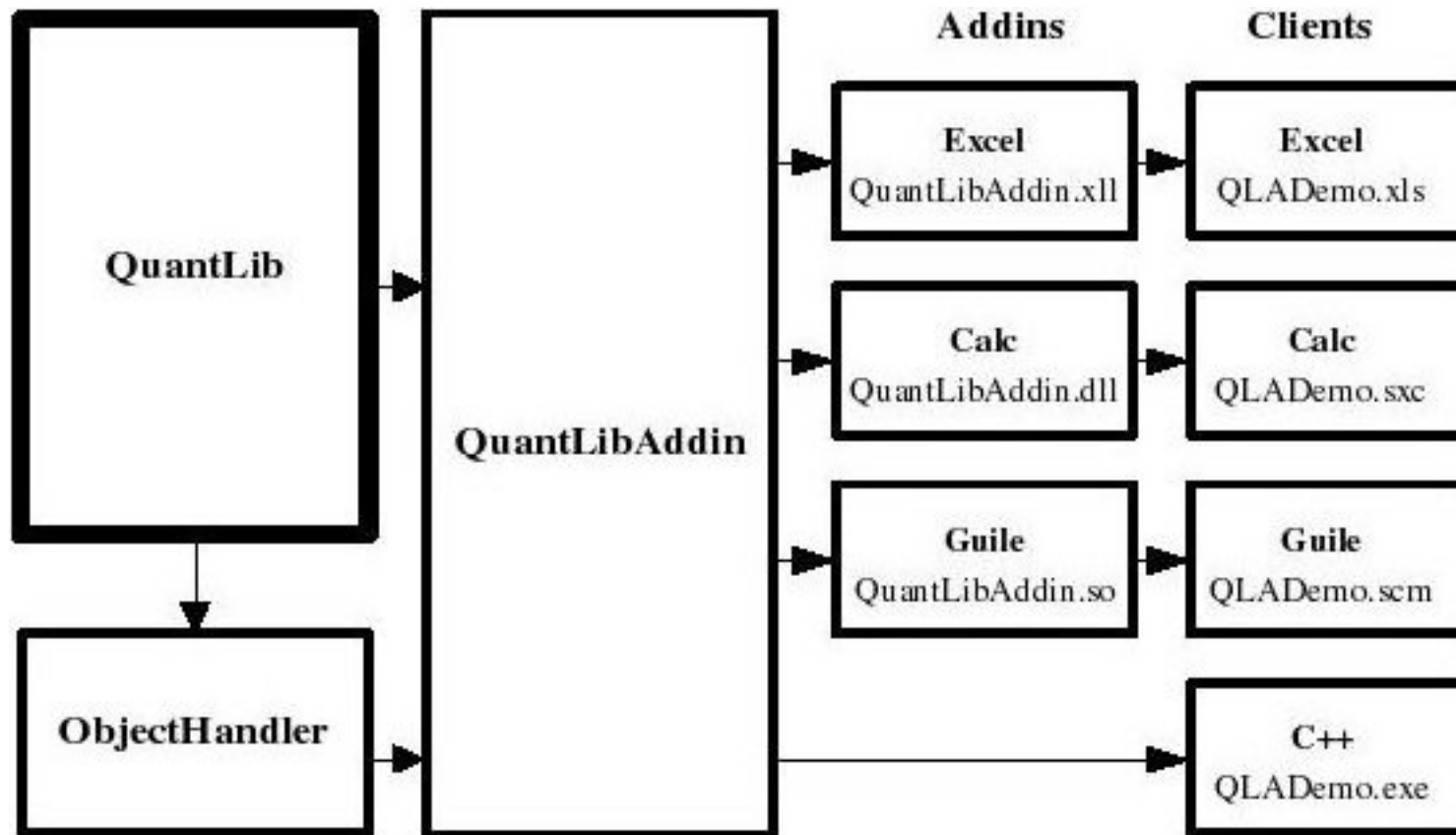
Object-oriented programming is a programming paradigm that uses *objects*—data structures consisting of data fields and methods together with their interactions—to design applications and computer programs. Programming techniques may include features such as data abstraction, encapsulation, modularity, polymorphism, and inheritance. Most modern programming languages support object-oriented programming.

Classes: generic containers

- Main Class (method) [implementation]
 - E.g.: Options (premium) [virtual]
- Sub classes (method)
 - E.g.: Vanilla options (premium) [Black-Scholes]
 - E.g.: Asian options (premium) [Monte Carlo method]

Objects: specific instances

- Objects (class instances)
 - Vanilla call on SP500, with strike 1100, expiring, ...
 - Asian option on EUR-USD, strike 1.4, ...
 - Asian option on FTSE-100, ...



QuantLibAddin (1/3)

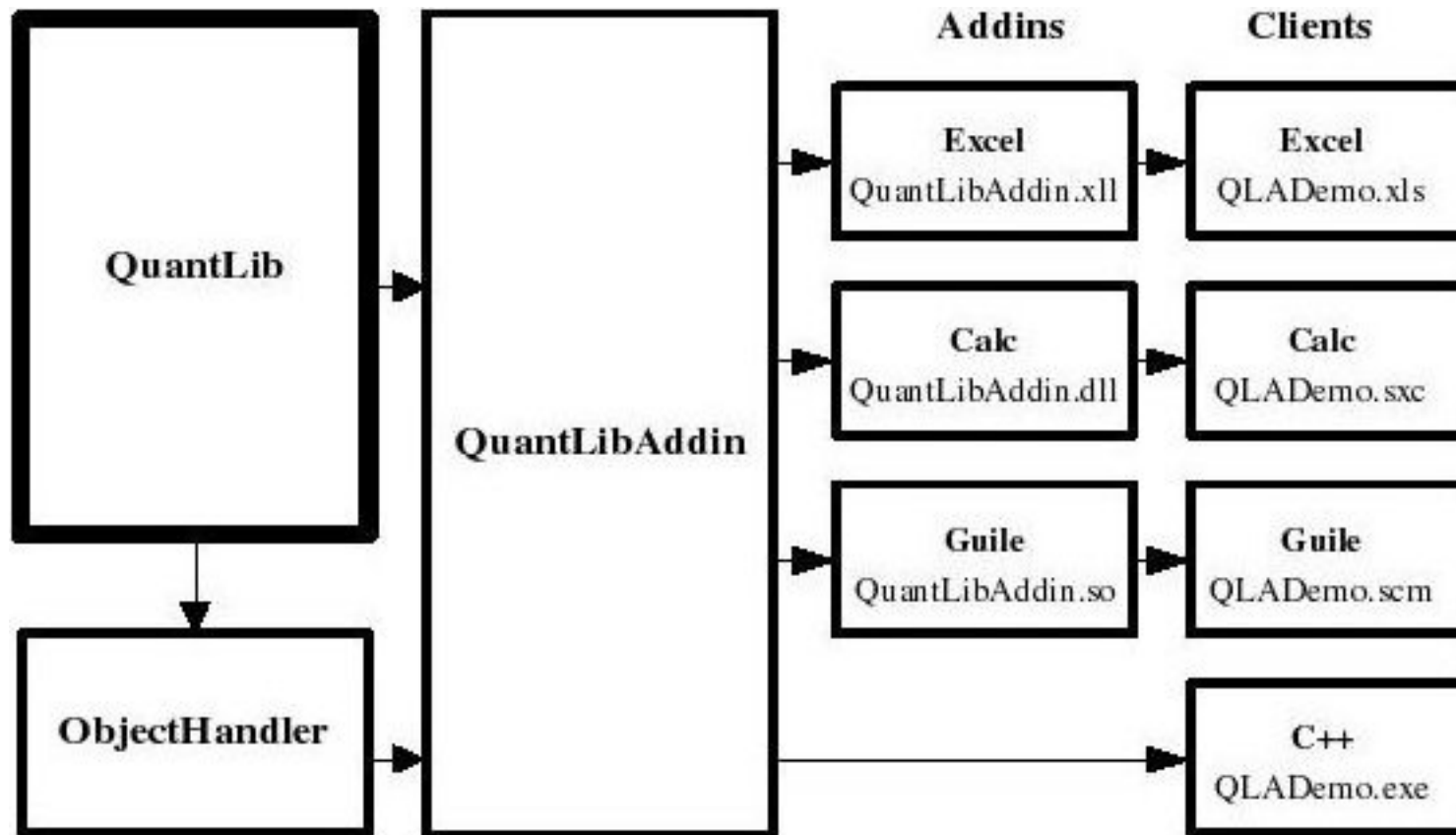
- Object Orientation: objects are the focus
 - QuantLib objects may be constructed, interrogated, updated, passed as arguments to other functions, and destroyed
- Polymorphism: methods are inherited by subclasses
 - For example, function `qINPV()` returns the NPV of an Instrument and may be invoked on an instance of any derived class such as Swap or Bond

QuantLibAddin (2/3)

- Enumerations: lists of fixed parameters
 - Minor or transient types are represented by a string, for example the string "TARGET" stands for `QuantLib::Calendar::TARGET`
- Portability: write once, use everywhere
 - The same interface is implemented in standard, i.e. platform neutral, C++ and is available on all supported environments

QuantLibAddin (3/3)

- Serialization: save objects as text (XML)
 - A pricing environment may be created in Excel workbooks and transmitted to a Linux process for the computation
- Coercion: convert type automatically
 - The user inputs for a function may be one of a number of data types; any necessary conversion is performed automatically



The Object Handler

- Objects are stored in the Excel-process memory
- The handler keeps track of objects
- Special function, constructors, create objects
- Objects are versioned using a version number (#0001)
- Methods can be called on objects depending on their class (e.g. discount on yield curves)

QuantLibXL

- Project started by Eric Ehlers (2004)
- Continually evolving
- Based on object-oriented programming
- Exposes QuantLib classes on an Excel interface
- Built on top of the QuantLibAddin: www.quantlibaddin.org
- In turn based on the Object Handler

Questions?

Accounting conventions

Pricing a financial instrument means computing the schedule of its future cash flows and the determining their current fair value

- Determine the financially relevant dates
- Determine the actual cash flows
- Determine the cash-flows present values

For example, what does an interest of 4% means in cash-flow terms?

Business calendars and bank holidays

- The set of days for which two legal entities agree to exchange cash flows is usually called a *business calendar*.
- The days in which it is not possible to determine or exchange cash flows are referred as *bank holidays*.
- For example, most banks in the Euro area follow the convention named *Trans-European Automated Real-time Gross settlement Express Transfer*, or *TARGET* for short

TARGET bank holidays

- Any Saturday
- Any Sunday
- New Year's Day: January 1st
- Good Friday (since 2000)
- Easter Monday (since 2000)
- Labor Day, May 1st (since 2000)
- Christmas, December 25th
- December 26th (since 2000)
- December 31st (1998, 1999, and 2001 only)

Business day conventions (1/2)

Business day conventions are used to determine what business days should be used for a payment when the given date falls on a bank holiday

- **Following** (ISDA) Choose the first business day after the given holiday.
- **Modified Following** (ISDA) Choose the first business day after the given holiday, unless it belongs to a different month. If the first business day is in a different month, choose the first business day before the holiday.

Business day conventions (2/2)

- **Preceding** (ISDA) Choose the first business day before the given holiday.
- **Modified Preceding** Choose the first business day before the given holiday, unless it belongs to a different month, in which case choose the first business day after the holiday.
- **Unadjusted** Do not adjust: use the given date even if it is an holiday (usually the year fraction, defined next, is computed on the holiday but the payment is deferred to the next day).

Day count conventions and year fractions (1/2)

- It is customary to express interest payments as the percentage on a certain notional amount per year
- Hence, we need to determine what the payments are when we are not dealing with a whole number of years
- For any two dates a *year fraction* is defined as the the number of years between them (possibly a decimal number)

Day count conventions and year fractions (2/2)

For example, if n is the number of calendar days between the start date d_1 and the end date d_2 (d_1 excluded and d_2 included), the year fraction according to the *Actual/360* convention is defined as,

$$T_{\text{actual}/360}(d_1, d_2) = \frac{n}{360} \quad (1)$$

More formally, given two dates d_1 and d_2 , the year fraction between them is denoted as

$$\tau = \tau_{1,2} = T(d_1, d_2) \quad (2)$$

Questions?

References

1. Marco Marchioro *Paper: Pricing simple interest-rate derivatives*,
<http://www.marchioro.org>
2. Wikipedia:
<http://en.wikipedia.org/wiki/TARGET>
http://en.wikipedia.org/wiki/Day_count_convention