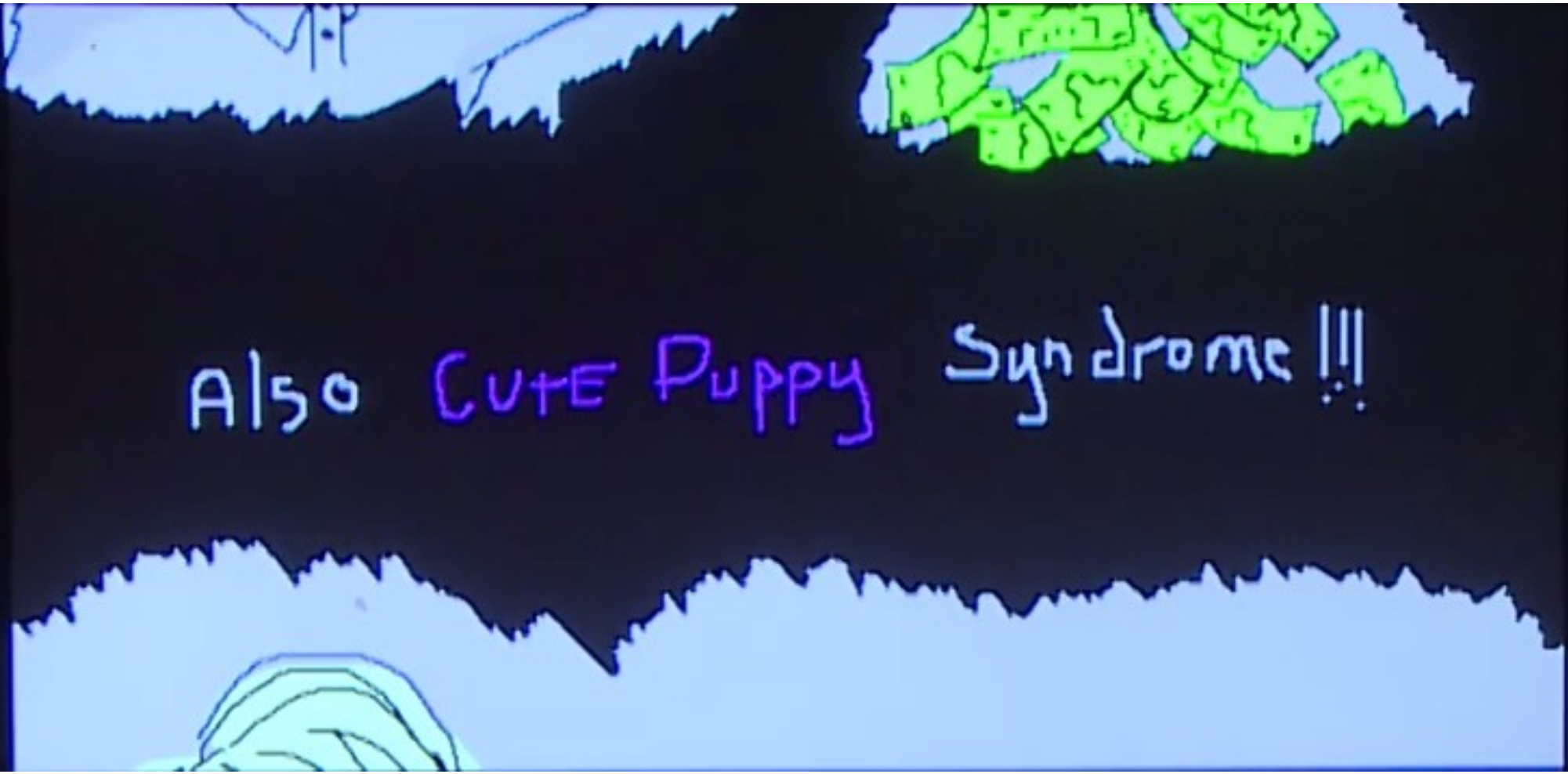
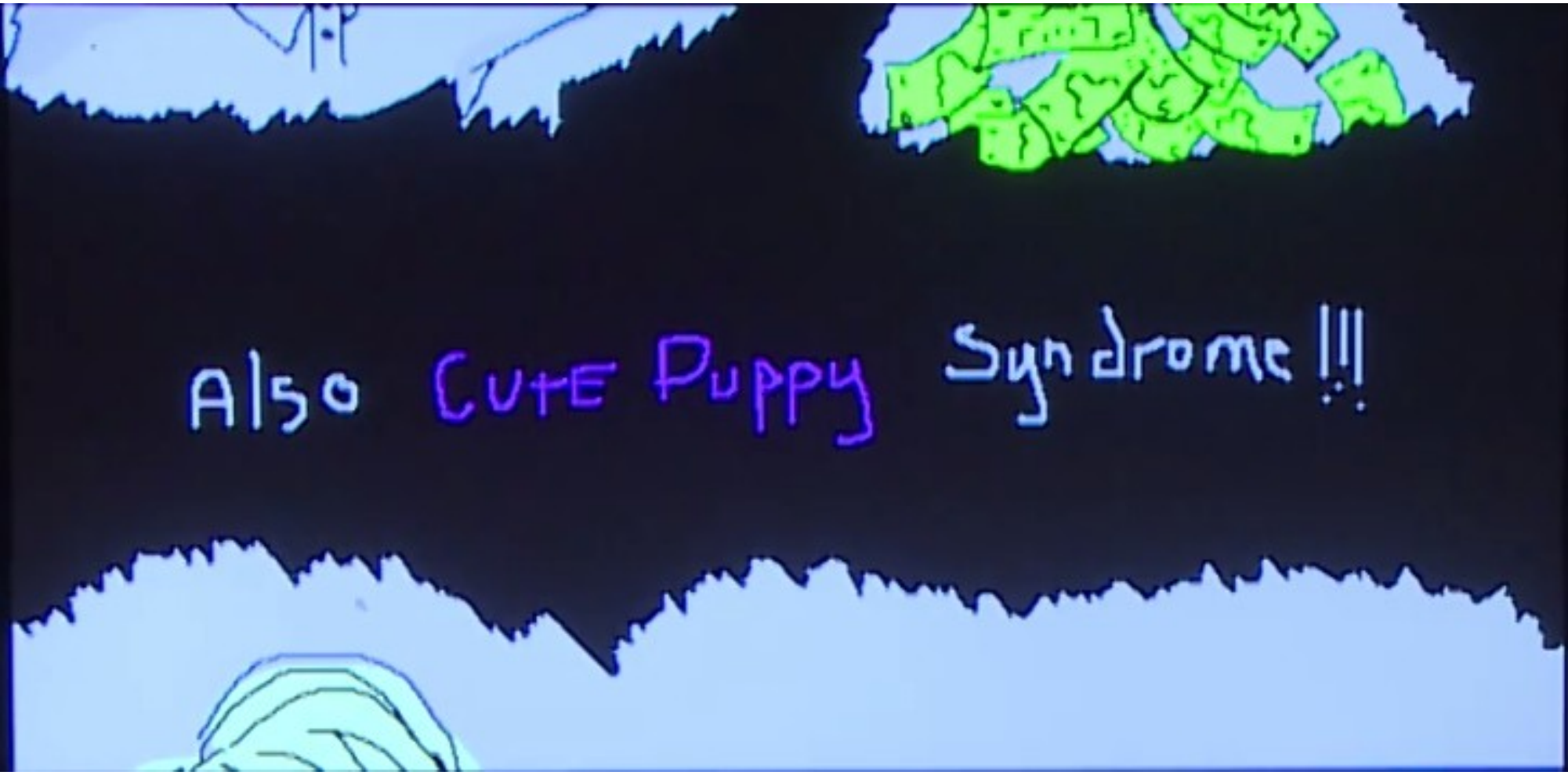
QuantLib at IKB

Michael von den Driesch

November 13, 2013



"Stolen" from Jacob Thorntons presentation @dotJS, Paris 2012
What Is Open Source & Why Do I Feel So Guilty?



"Stolen" from Jacob Thorntons presentation @dotJS, Paris 2012
What Is Open Source & Why Do I Feel So Guilty?

Disclaimer

The contents of this presentation are the sole and personal opinion of the author and do not express IKB's opinion on any subject presented in the following.

Table of contents

Why am I giving this talk ?

- Create a network of people that know, trust and potentially help each other
- Inspire other people to use QuantLib
- Learn from others what we could do better

Structure

- Group title - "Pricing and financial modelling"
- Group is part of IKB's Risk Control
- Currently 8 members
- Background of personnel
  - Mathematics
  - Physics
  - Quantitative Finance

Main functions

- Front office independent financial engineering for various asset classes
- Continuous validation of used models
- Inhouse consultancy for various pricing and model related issues

Applications utilising QuantLib functionality within our:

- Executable based end of day valuation:
  - Pricing + Risk (hist. scenarios) of inflation bonds and swaps
  - Daily CVA and PFE calculations
  - External calibration of SABR vol surfaces
- Validation of FO system pricing and sensitivities
- Pricing of structured loans
- Pricing of CDO like structures
- Low (below zero) rate cap vol fall-back
- Second opinion for "important" prices

- Each of our production applications get's the hash tag of the git repository that it has been build on.
- Use dos batch to create include file that contains the tag
- Use prebuild step in Visual Studio 2010 to call the batch
- Use msbuild in batch context to create nigthly builds of apps
- Current work in progress: create regression test framework, to do daily tests of existing apps against default results.

The dos batch for our revision include looks like this:

```
1  @echo off
2  :: vdd: The next line does a change directory into the directory where the batch is stored
3  cd /d %0\..
4  :: Check if git is installed under x:\git\bin
5  if not exist x:\git\bin\git.exe
6      echo Attention: there seems to be no git installation under x:\git\bin
7
8  x:\git\bin\git log -1
9      --format="format:#ifndef gitrevision_h%%n#define gitrevision_h%%n
10     #define GITREVISION \"%%H\"%%n#endif" > gitrevision.h_tmp
11
12 :: us git diff tool to check,whether the two revision strings are identical
13 :: only if they differ, create a new header file.
14 :: This is to avoid the infamous recompilation loop
15 x:\git\bin\diff.exe -q gitrevision.h_tmp gitrevision.h > NUL
16 if '%errorlevel%' NEQ '0' (
17     copy gitrevision.h_tmp gitrevision.h
18 )
19
20 :End
```

and returns the following include:

```
#ifndef gitrevision_h
#define gitrevision_h
#define GITREVISION "4552e9e5f6c315d4fc1ba00300c8fa5e78a76591"
#endif
```

The git hash is then readily available in all our applications and any of our production codes will log this revision into the eod log file (which is compulsory).

Examples are mostly build on pure QuantLib Code using the existing swig python wrapper.
Examples will be based on interactive IPython sessions.

It is very handy, to have a simple and solid market data repository ... let's see, what I mean.

Examples are mostly build on pure QuantLib Code using the existing swig python wrapper.
Examples will be based on interactive IPython sessions.

It is very handy, to have a simple and solid market data repository ... let's see, what I mean.

Example is based on Peter Casper's implementation of a Gaussian Short Rate (GSR) Model (for a reference see [1]).
It can be show, that the mean reversion can be interpreted as a level of correlation of the short rate process at different times. If one assumes σ and a as constant, then one can show, that:

$$corr(x(T_1), x(T_2)) = e^{-a(T_2-T_1)} \sqrt{\frac{1-e^{-2aT_1}}{1-e^{-2aT_2}}}.$$

This means, that $a = 0 \implies corr(\bullet, \bullet) = \sqrt{(T_1/T_2)}$,
$a \to \infty \implies corr(\bullet, \bullet) = 0$ and $a \to -\infty \implies corr(\bullet, \bullet) = 1$.
Idea: validate, whether the implemented model delivers the expected qualitative dependence of the mean reversion.

[1] Leif B.G. Andersen and Vladimir V. Piterbarg; Interest Rate Modelling

Swaption Termsheet

- Pricing Date: 27.08.2013
- Underlying Notional: 10 Mio EUR
- Underlying Recieve 2.85%, quarterly, daycount=Actual360
- Underlying Pay EURIBOR 3M, quarterly, daycount=Actual360
- Exercise Dates: 30.09.2013, 30.09.2014

There is always potential for improvement. At IKB I see the following issues:

- Establish internal design authority
- Continuous integration into IT processes
- Knowledge transfer and standby personnel
- Documentation
- Better unit tests and regression tests

Personal view:

- Extend standard market data framework
- Extend usage for validation purposes
- Abandon Excel for any "near production" use case and instead
- Extend usage of QL Python interface to build maintainable, versionizable and easily customizable frameworks around core pricing

The End ...

Thank you for listening.
Any questions or comments?