d-fine

QuantLib(XL) for Model Validation

Dr. Georg Schöchtel

QuantLib User Meeting 2016 - IKB/Düsseldorf

Düsseldorf, 7th December 2016

Agenda

1. Introduction Model Validation
2. QuantLibXL for Model Validation
3. Typical Deliverables
4. Summary

Introduction Model Validation

Regulatory Requirements

BCBS	» BCBS 153, “Supervisory guidance for assessing banks’ financial instrument fair value practices” › Guidelines for fair value valuation (10 principles) › Valuation processing and valuation control by the bank › Risk management and disclosure of risk as a result of the model valuation › Monitoring of the valuation process by the supervision › Initial, periodic and event-based validation
BaFin	» Risk identification and risk measurement (§25a KWG, MaRisk)
FED / OCC	» SR letter 11-7 / Bulletin 2011-12, “Supervisory Guidance on Model Risk Management” » Due to bank’s Comprehensive Capital Analysis and Review (CCAR) submissions, banks are required to submit documentation regarding their model risk management policy and practices.

Model validation should be in accordance with regulatory guidelines

d-fine

A sound model validation and model certification is crucial for risk mitigation in an organisation

New Products – New Market Data- New Industry Standards – New IT Infrastructure

Operational Risk	Model Risk	Market Risk
Can systems process all products in current market environment and risk scenarios	Are models in line with peers and market standard given the current market environment (e.g. negative rates)?	Can models calibrate to market environment and yield reasonable pricing and risk numbers?

What if the answer is 'No'?

» Interruption in EoD valuation runs » Limitations of new business	» Collateral disputes » Disadvantages when competing for deals	» Wrong prices/hedges/risks » Bleeding P&L due to potential arbitrage

Model validation is aimed at answering above questions, identifying potential gaps and proposing remediation actions

d-fine

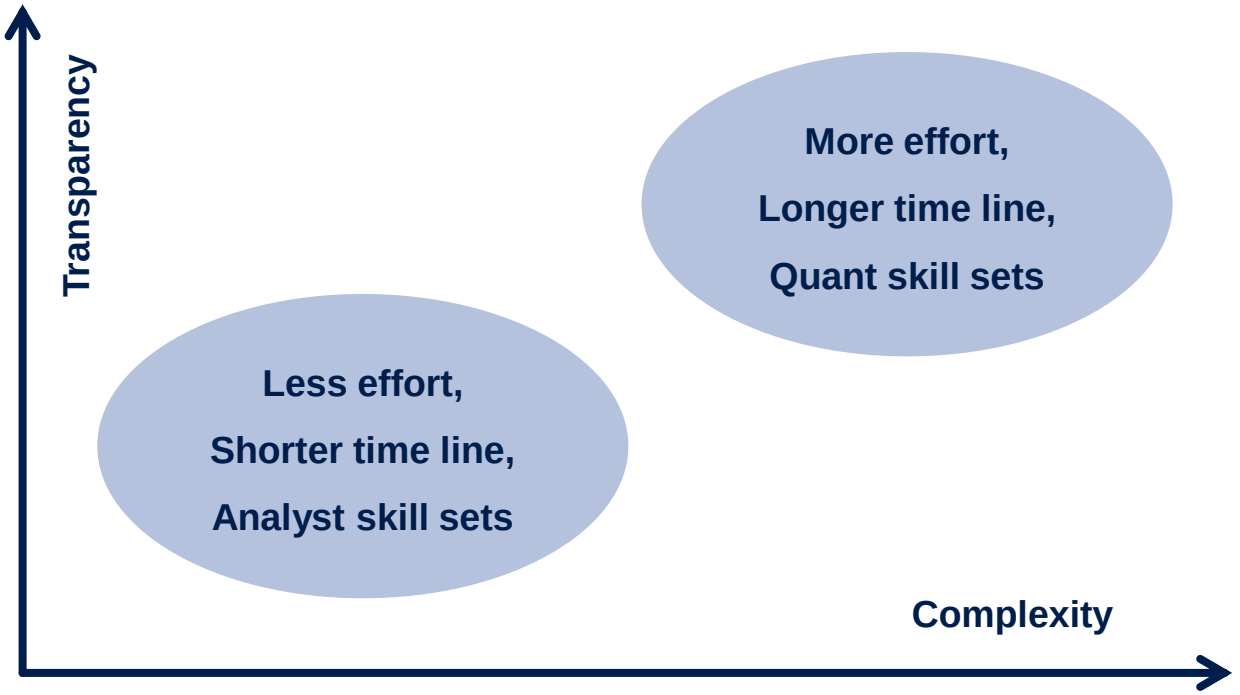
Model validation is driven by the complexity of products/market data/models involved and the level of transparency desired

full

- **Can we independently reproduce all calibration pricing and risk results?**
- Does model replicate **calibration inputs**?
- Are **model-independent properties** preserved (e.g. put-call parity)?
- Does model behave **reasonable in stress** (i.e. model limits)?

basic

- Are models and market data **assigned and configured** as specified?
- Does the model capture all **relevant risk factors?**
- Does the model produce all relevant prices and risk **figures?**



low	medium	high
PR: FRA	PR: Caps/Floors	PR: CMS-Spread
MD: IR Quotes	MD: Caplet Vol. Surf.	MD: Swaption Vol.
MO: Discounting	MO: Bachelier	MO: 2F Hull-White

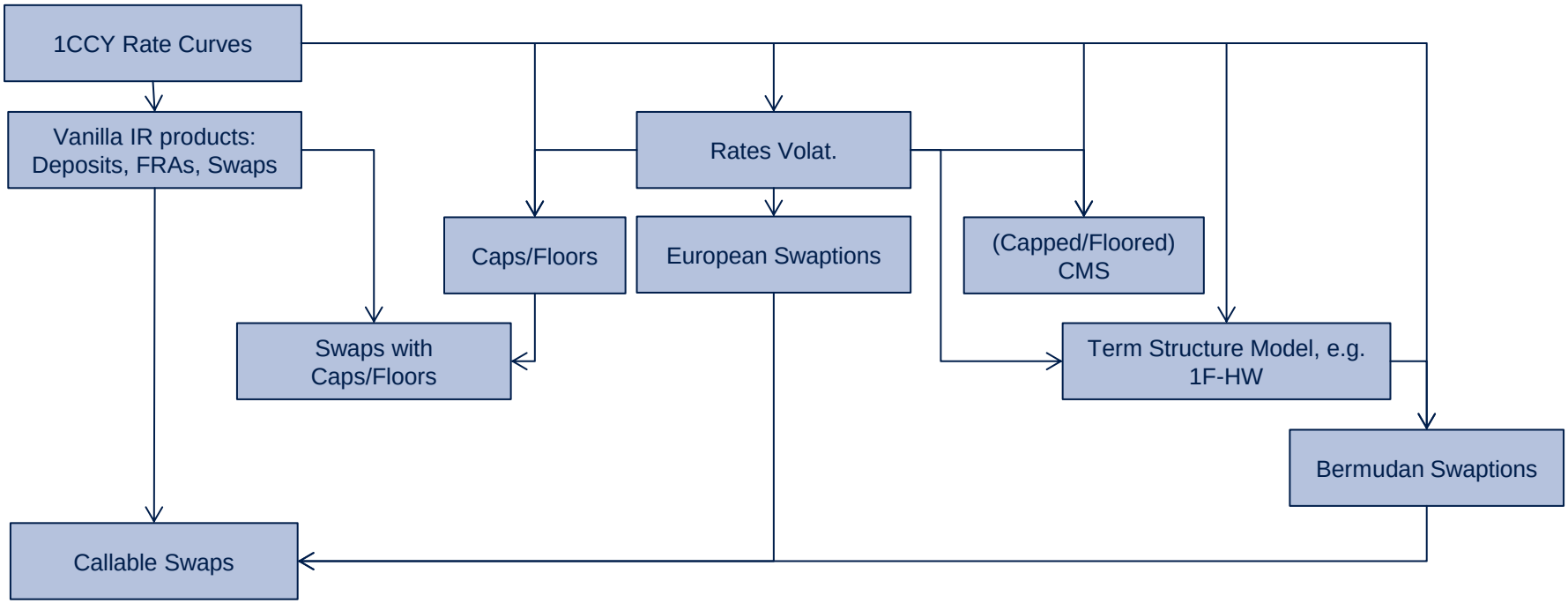
Model validation approaches to product scope and desired level of transparency taking into account timely and budgetary constraints

PR … Product, MD … Market Data, MO … Model

d-fine

Risk and Product Dependencies in Model Validation

- Validation processing is based on risk and product dependencies
- Example: From 1CCY Vanilla IR products to callable swaps



Validation should start with risk factors and advance to (complex) products

Main criteria for reference pricer are product coverage, transparency and available analytical tools

Pricer	Pros	Cons
Commercial in-house library	» Very good product coverage of Vanilla and exotic products » Detailed documentation » In-house established	» Model details not always transparent (proprietary software) » Extensions/Adjustments not possible (at most upgrade)
Bloomberg Swap Manager	» (Very) good product coverage of Vanilla and exotic products » Direct integration of market data » (In-house established)	» Model details not always transparent (proprietary software) » Restricted analyse tools
QuantLib	» (Very) good product coverage of Vanilla and exotic products » Full transparency since open source » In principle flexibly extensible » No licence fees	» Expert tool with appropriate know-how requirements » For the most part documentation in source code

QuantLibXL for Model Validation

Using QuantLibXL for Model Validation

Why do we use QuantLibXL in validation projects instead of other QuantLib interfaces (e.g. Python)?

Client's IT infrastructure	» Installation of special software usually not allowed due to security reasons » Easy availability and integration via Excel addin » Easy integration of additional required data (e.g. market data via Bloomberg Excel addin) » Client and internal/external auditors should be able to reproduce the calculations
Skill requirements and further use	» Client and internal/external auditors are familiar with Excel/VBA applications » To a certain extent only analyst skills are required » Constructed QuantLib-Excel workbooks as repository for (re)validation or validation of new product features with (small) adjustments

» **Remark:** At the moment we use the "old" QuantLibAddin with gensrc Python script instead of the SWIG Reposit module

QuantLibXL is used due to infrastructure restrictions and skill requirements

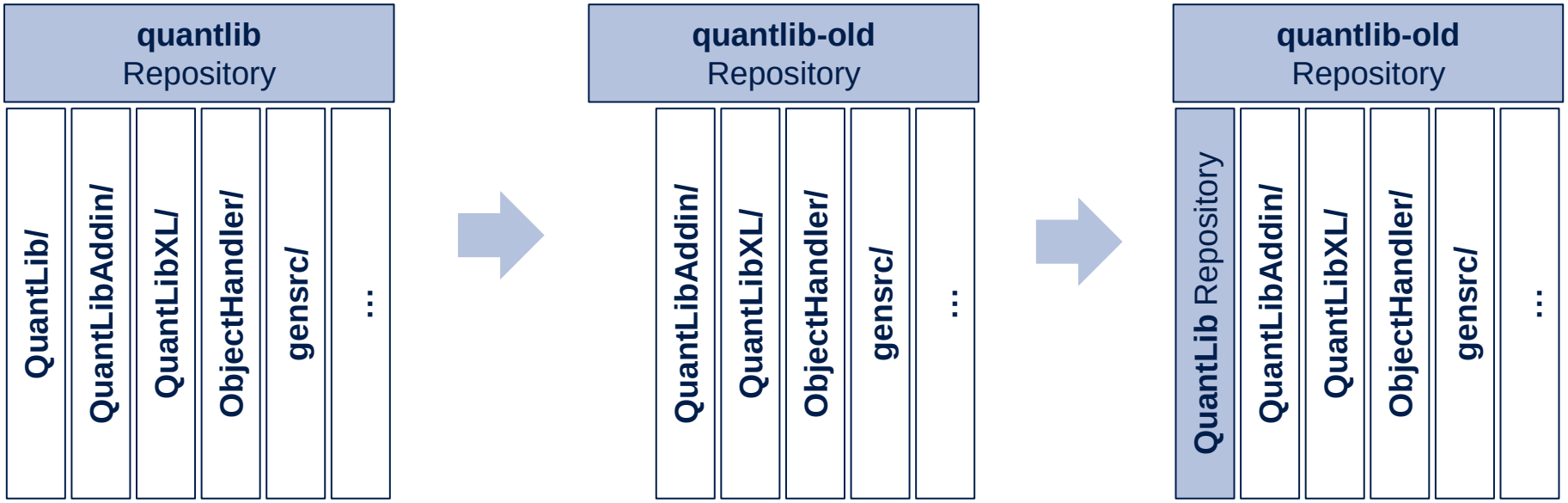
d-fine

We manage two repositories to maintain legacy QuantLibXL and updated QuantLib library

Prior Dec'15 setting
» Various projects in a single Git repository

Project split in Dec'15
» Keep legacy QuantLibXL and gensrc infrastructure
» Strip off QuantLib/ folder

Current setup
» include QuantLib library as independent submodule



We aim at migrating from gensrc/QuantLibXL to Reposit/QuantLibXL in 2017

QuantLib repository is forked from lballabio/QuantLib and set up as Git Submodule to keep up with recent developments

Using QuantLibXL for Model Validation

Available in QuantLibXL

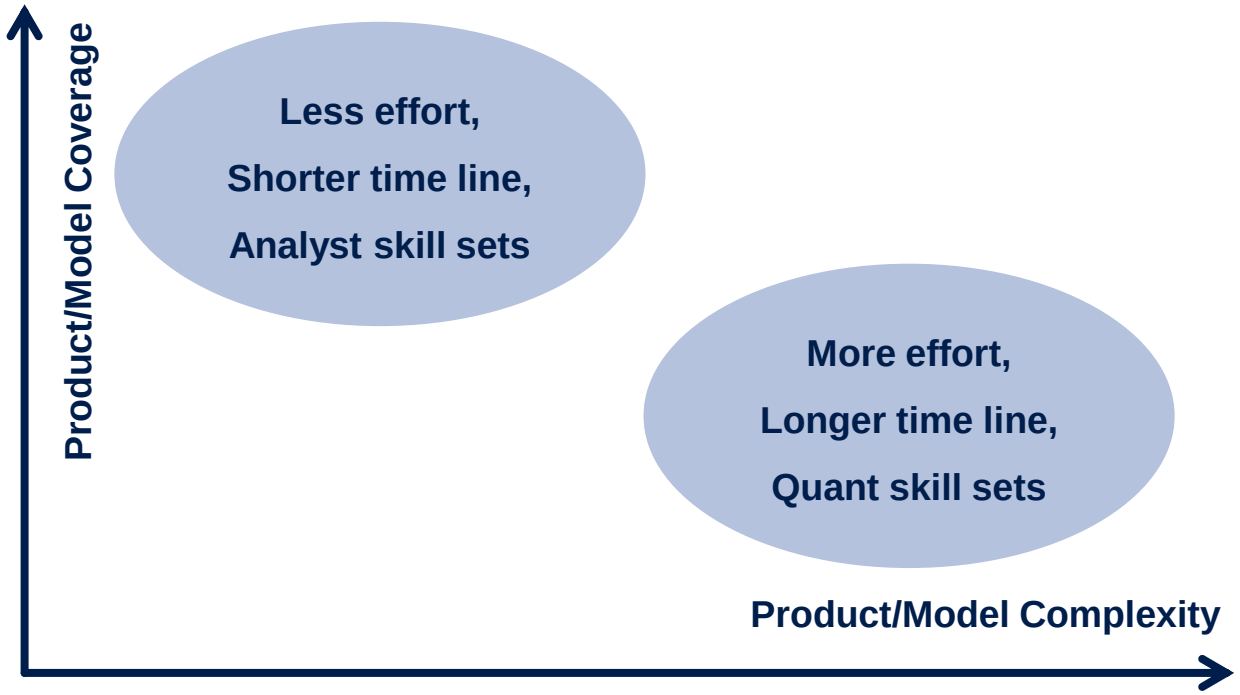
- Direct use of QuantLib-Excel functions

Implemented in QuantLib, but not in QuantLibXL

- Transfer to Excel via “old” QuantLibAddin or SWIG Reposit required
- Test of successful transfer

Partially implemented in QuantLib

- Implementation
- Transfer to Excel via “old” QuantLibAddin or SWIG Reposit required
- Test of implementation and transfer



Vanilla linear products

- Vanilla Swaps
- Fix/Float Bonds
- Discounting CashFlow

Vanilla Options

- Caps/Floors
- Swaptions
- FX Options

Complex Products

- Callable Swaps
- CMS-Spread
- Asian Options

Time effort for model validation depends also on product/model coverage in QuantLibXL

PR … Product, MD … Market Data, MO … Model

d-fine

Case: Required functionality available in QuantLibXL

» Test case: Validation of 1CCY swap curves and Vanilla swap

1CCY swap curves

- Construction
  - RateHelpers (qlOISRateHelper, qlDepositRateHelper, qlFraRateHelper, qlSwapRateHelper, …) are used for defining benchmark instruments of the curves
  - Yield curves are constructed with qlPiecewiseYieldCurve

Eonia Curve

RateHelpers | BootstrapIndex | Eonia-Bootstrapindex#0000

ObjectID	Quote	Period Tenor	FixingDays	Day Counter Eonia Index	RateHelper	Error
EMMI EURO OverNight Index	-0.2370%	1D	0	Eonia-Bootstrapindex#0000	EONIA=#0000	
EUR SWAP (EONIA) 1 WK	-0.2380%	1W	2	Eonia-Bootstrapindex#0000	EUREONSW=#0000	
EUR SWAP (EONIA) 2 WK	-0.2380%	2W	2	Eonia-Bootstrapindex#0000	EUREON2W=#0000	
EUR SWAP (EONIA) 3 WK	-0.2380%	3W	2	Eonia-Bootstrapindex#0000	EUREON3W=#0000	
EUR SWAP (EONIA) 1 MO	-0.2370%	1M	2	Eonia-Bootstrapindex#0000	EUREON1M=#0000	
EUR SWAP (EONIA) 2 MO	-0.2710%	2M	2	Eonia-Bootstrapindex#0000	EUREON2M=#0000	
EUR SWAP (EONIA) 3 MO	-0.2970%	3M	2	Eonia-Bootstrapindex#0000	EUREON3M=#0000	
EUR SWAP (EONIA) 4 MO	-0.3130%	4M	2	Eonia-Bootstrapindex#0000	EUREON4M=#0000	
EUR SWAP (EONIA) 5 MO	-0.3310%	5M	2	Eonia-Bootstrapindex#0000	EUREON5M=#0000	
EUR SWAP (EONIA) 6 MO	-0.3450%	6M	2	Eonia-Bootstrapindex#0000	EUREON6M=#0000	
EUR SWAP (EONIA) 7 MO	-0.3580%	7M	2	Eonia-Bootstrapindex#0000	EUREON7M=#0000	
EUR SWAP (EONIA) 8 MO	-0.3670%	8M	2	Eonia-Bootstrapindex#0000	EUREON8M=#0000	
EUR SWAP (EONIA) 9 MO	-0.3760%	9M	2	Eonia-Bootstrapindex#0000	EUREON9M=#0000	
EUR SWAP (EONIA) 10 MO	-0.3840%	10M	2	Eonia-Bootstrapindex#0000	EUREON10M=#0000	
EUR SWAP (EONIA) 11 MO	-0.3910%	11M	2	Eonia-Bootstrapindex#0000	EUREON11M=#0000	
EUR SWAP (EONIA) 1 YR	-0.3970%	1Y	2	Eonia-Bootstrapindex#0000	EUREON1Y=#0000	
EUR SWAP (EONIA) 15 MO	-0.4080%	15M	2	Eonia-Bootstrapindex#0000	EUREON15M=#0000	
EUR SWAP (EONIA) 18 MO	-0.4170%	18M	2	Eonia-Bootstrapindex#0000	EUREON18M=#0000	
EUR SWAP (EONIA) 21 MO	-0.4200%	21M	2	Eonia-Bootstrapindex#0000	EUREON21M=#0000	
EUR SWAP (EONIA) 2 YR	-0.4190%	2Y	2	Eonia-Bootstrapindex#0000	EUREON2Y=#0000	
EUR SWAP (EONIA) 3 YR	-0.3780%	3Y	2	Eonia-Bootstrapindex#0000	EUREON3Y=#0000	
EUR SWAP (EONIA) 4 YR	-0.2990%	4Y	2	Eonia-Bootstrapindex#0000	EUREON4Y=#0000	
EUR SWAP (EONIA) 5 YR	-0.1960%	5Y	2	Eonia-Bootstrapindex#0000	EUREON5Y=#0000	
EUR SWAP (EONIA) 6 YR	-0.0750%	6Y	2	Eonia-Bootstrapindex#0000	EUREON6Y=#0000	

Yield Curve Stripping

SwapEngine	EONIA-Curve#0004
Error	
ObjectID	EONIA-Curve
NDays (advance)	0
Calendar	TARGET
RateHelpers	
DCC	Actual/365 (Fixed)
Jumps	
JumpDates	
Accuracy	
Traits	ZeroYield
Interpolator	Linear
Permanent	
Trigger	
OverWrite	
Extrapolator	TRUE

Eonia-Index	Eonia#0003
Error	
ObjectID	Eonia
Tenor	
FwdCurve	EONIA-Curve#0004
Permanent	
Trigger	
OverWrite	

Case: Required functionality available in QuantLibXL (cont.)

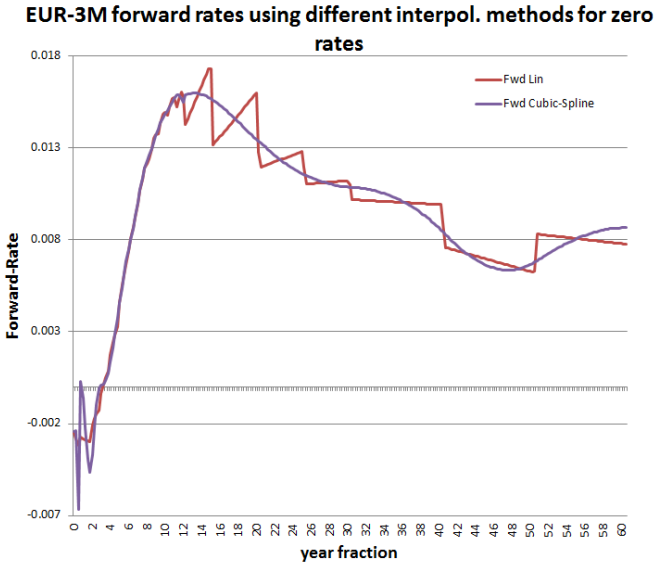
» Test case: Validation of 1CCY swap curves and Vanilla swap (cont.)

1CCY swap curves

- Validation
  - Compare bootstrapped QuantLib zero rates (qlYieldTSZeroRate) with zero rates of the respective system
  - Use QuantLibXL for further tests (e.g. impact on forward rates using different interpolation methods for zero rates)

Comparison with system zero rates

Dates	Period	QuantLib zero rates	System zeros rates	Delta in BP
02.02.2016	0D	-0,2400%		
03.02.2016	1D	-0,2400%	-0,2400%	0,0000
11.02.2016	1W	-0,2408%	-0,2408%	-0,0023
18.02.2016	2W	-0,2409%	-0,2409%	-0,0013
25.02.2016	3W	-0,2410%	-0,2409%	-0,0009
04.03.2016	1M	-0,2400%	-0,2400%	-0,0007
04.04.2016	2M	-0,2733%	-0,2733%	-0,0003
04.05.2016	3M	-0,2995%	-0,2995%	-0,0002
06.06.2016	4M	-0,3158%	-0,3158%	-0,0002
04.07.2016	5M	-0,3340%	-0,3340%	-0,0001
04.08.2016	6M	-0,3483%	-0,3483%	-0,0001
05.09.2016	7M	-0,3616%	-0,3616%	-0,0001
04.10.2016	8M	-0,3708%	-0,3708%	-0,0001
04.11.2016	9M	-0,3800%	-0,3800%	-0,0001
05.12.2016	10M	-0,3882%	-0,3882%	-0,0001
04.01.2017	11M	-0,3954%	-0,3954%	-0,0001
06.02.2017	1Y	-0,4016%	-0,4016%	-0,0001
04.05.2017	15M	-0,4128%	-0,4131%	0,0359
04.08.2017	18M	-0,4219%	-0,4226%	0,0644
06.11.2017	21M	-0,4251%	-0,4259%	0,0840
05.02.2018	2Y	-0,4243%	-0,4243%	0,0000
04.02.2019	3Y	-0,3831%	-0,3831%	0,0000
04.02.2020	4Y	-0,3035%	-0,3035%	0,0000
04.02.2021	5Y	-0,1994%	-0,1994%	0,0000
04.02.2022	6Y	-0,0766%	-0,0766%	0,0000



Case: Required functionality available in QuantLibXL (cont.)

» Test case: Validation of 1CCY swap curves and Vanilla swap (cont.)

Vanilla Fix-Float Swap

- Construction and Validation
  - After successful validation of 1CCY swap curves use the zero rates of the system and qlInterpolatedYieldCurve for defining the respective discount and forward curves
  - Construction and pricing of swap with qlSchedule, qlFixedRateLeg, qlIborLeg, qlSwap and qlInstrument
  - In addition check schedule dates and cashflows with qlLegFlowAnalysis
  - Use QuantLibXL for further tests (e.g. BPV sensitivity calculation)

Swap NPV

Swap-Engine	SwapEngine_Test_Swap#0003
Error	
ObjectID	SwapEngine_Test_Swap
YieldCurve	Eonia-Continuous-Linear#0005

Set Pricing-Engine	TRUE
Error	
Instrument (Swap)	Swap-Test#0003
Swap-Engine	SwapEngine_Test_Swap#0003

Swap-NPV QuantLib	3.110.526,34
Swap-NPV System	3.110.526,33
Delta NPV	0,01

Quick validation of products if all required functionality is available in QuantLibXL

d-fine

Case: Partially implemented in QuantLib

» Test case: Implement “special cubic” local interpolation and convention “compounded”

Problem

- Some of client’s interest rate curves based on zero rates use
  - “special cubic” local interpolation method $z(t_x)=z(t_a)+(z(t_b)-z(t_a))\ \frac{(t_x^3-t_a^3)}{(t_b^3-t_a^3)}$ with $t_a \leq t_x \leq t_b$
  - and compounding convention “compounded”, i.e. $D(t,T)=\frac{1}{(1+z(t,T))^T}$.
- “Special cubic” interpolation is not implemented in QuantLib
- qlInterpolatedYieldCurve(ObjectID, Dates, Data, Calendar, DayCounter, Jumps, JumpDates, TraitsID, InterpolatorID, Permanent, Trigger) does not provide compounding argument and uses continuous compounding for zero rates (TraitsID = ZeroYield) by default

Tasks

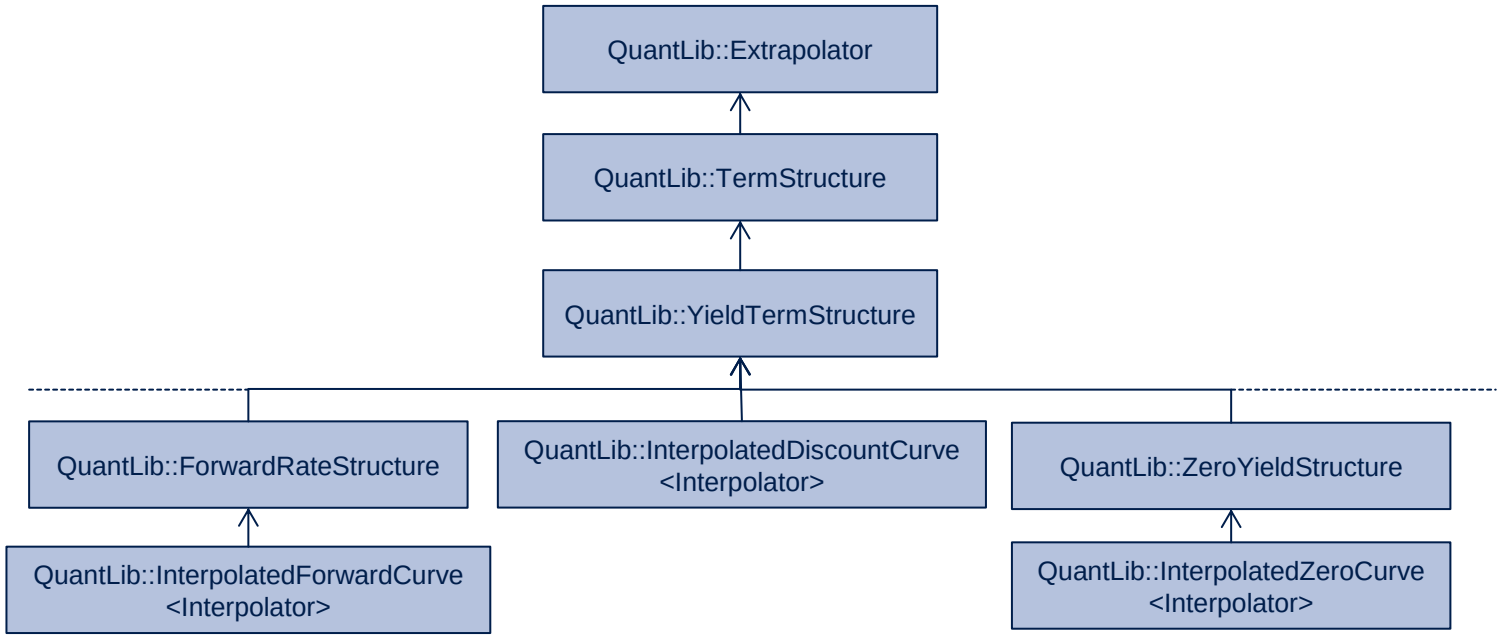
- Implement “special cubic” interpolation in QuantLib
- Implement new class/ QuantLib-Excel function (ql)InterpolatedYieldCurveFromInterpolatedZeroCurve including compounding argument

Case B: Partially implemented in QuantLib (cont.)

» **Test case: Implement “special cubic” local interpolation and convention “compounded”**

(ql)Interpolated YieldCurve

- QuantLibAddin::InterpolatedYieldCurve based on YieldTermStructure is implemented using the factory pattern with TraitsID (Discount, ZeroYield, ForwardRate) and InterpolatorID
- Compounding argument with TraitsID “Discount” does not make sense
- Quick integration of “special cubic” interpolation due to the factory pattern after implementing it analogous to the linear Interpolation

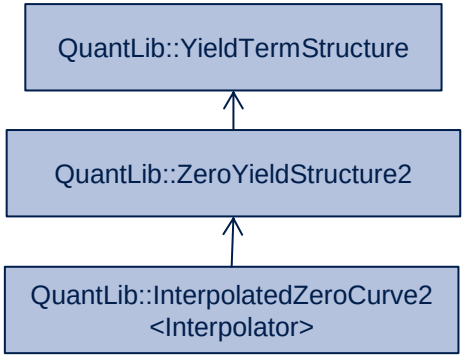


Case B: Partially implemented in QuantLib (cont.)

» **Test case: Implement “special cubic” local interpolation and convention “compounded”**

(ql)InterpolatedYieldCurveFromInterpolatedZeroCurve

- InterpolatedYieldCurveFromInterpolatedZeroCurve is implemented analogous to InterpolatedYieldCurve using the factory pattern now with CompoundingID (Simple, Compounded, Continuous) and InterpolatorID.
- New classes ZeroYieldStructure2 and InterpolatedZeroCurve2 with appropriate adjustments incorporating compounding feature
- Quick integration of “special cubic” interpolation due to the factory pattern



Quick implementation and transfer to Excel of minor features

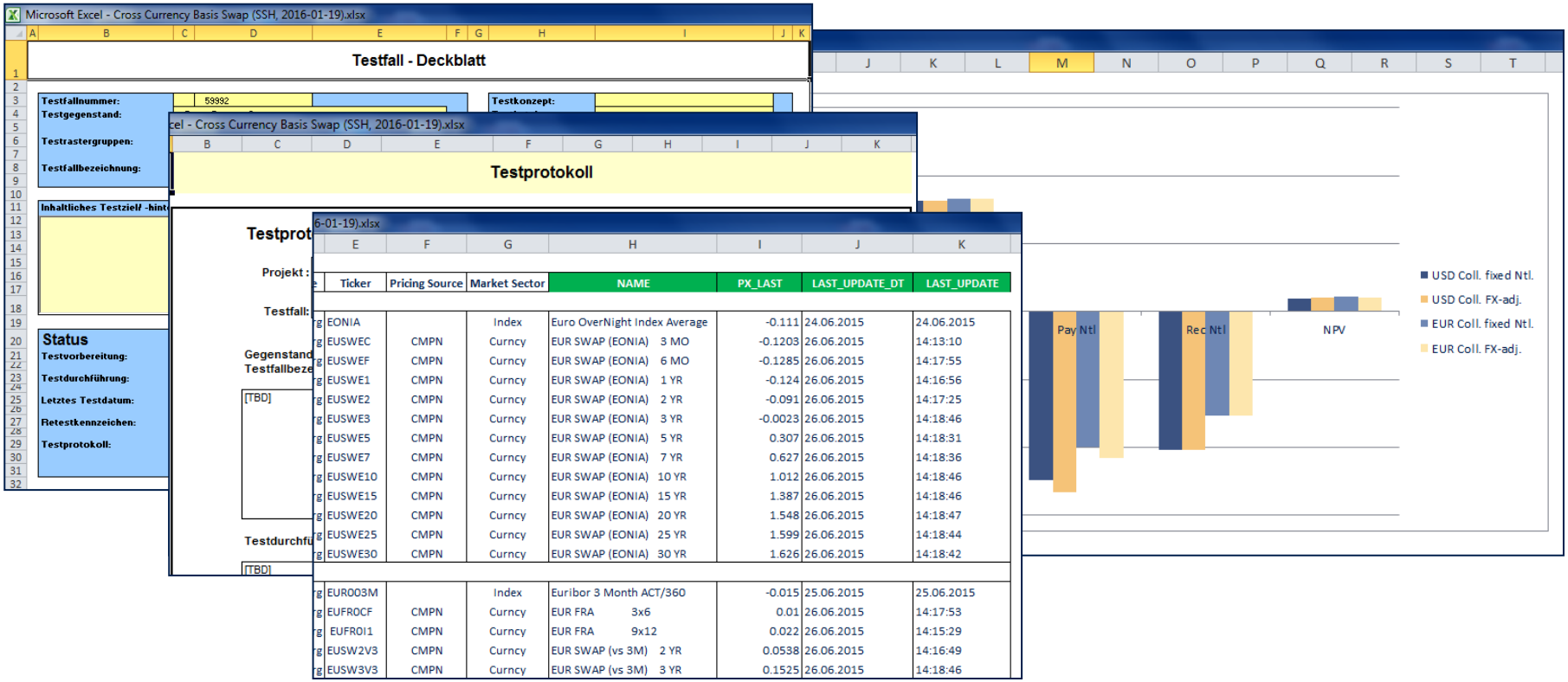
Some advanced extensions of QuantLib(XL) used in Model Validation

1-Factor Hull-White	» Current core implementation includes only 1-Factor Hull White process with scalar short rate volatility » For validation of Callable swaps and bonds a generic 1-Factor Hull White model with time-dependent short rate volatility was implemented and transferred to Excel
FX Barrier Options	» For model validation purposes the valuation of FX Barrier options with analytic pricing formulas based on the Garman-Kohlhagen model was required » Available in QuantLib, but had to be transferred to Excel
(Capped/Floored) CMS	» Implementation and transfer to Excel of an affine Terminal Swap Rate (TSR) model using normal swaption volatilities for analytic pricing of (Capped/Floored) CMS

Typical Deliverables

Typical Deliverables – QuantLib-Excel test workbooks

» Reproducibility of test processing and of test results are ensured by standardized QuantLib-Excel sheets



Typical Deliverables – Validation Documents and Work Shops

Standardized validation document

- Mathematical background and market standard valuation
- Portfolio analysis and reference trades
- Test processing and test results
- Criticality of findings and improvement proposals

Work shops (~ 2 to 3 h)

- Work shops present the validation results with further information on mathematical background and market standard valuation
  - Advanced training on (new) models and valuation methods
  - Discussion on validation findings and improvement proposals
- Previous work shops
  - „Multi Curve Discounted Cashflow Method“
  - „Valuation of IR Options with Black76 Model and Bachelier Model“
  - „Valuation of (Capped/Floored) CMS“
  - „Hull White Model and Valuation of Callable Swaps“

Summary

Summary

- Model Validation is important in view of regulatory requirements and risk organisation
- QuantLib(XL) is suitable for model validation in view of product coverage, transparency and available analytical tools
- In comparison to other interfaces QuantLibXL is used due to infrastructure restrictions and skill requirements
- Time effort for model validation depends also on product/model coverage in QuantLibXL
- Typical Deliverables in a validation project are QuantLib-Excel test sheets, validation documents and work shops

Contact

Dr. Georg Schoechtel
Senior Consultant
Tel +49 89 7908617-0
Mobil +49 162 2631514
E-Mail Georg.Schoechtel@d-fine.de

d-fine

Frankfurt
München
London
Wien
Zürich

Zentrale

d-fine GmbH
An der Hauptwache 7
D-60313 Frankfurt/Main

Tel +49 69 90737-0
Fax +49 69 90737-200

www.d-fine.com


d—fine