Negative rates in QuantLib

Peter Caspers

Quaternion

November 30, 2015

Table of contents

History of negative fixings

- it started with negative EONIA fixings end of 2014
- then we had negative Euribor 1m, later 3m, even 6m fixings
- as of 27-Oct-2015 we have a negative CMS2Y fixing (at -3.5 bp)

Implications of negative fixings

- interest compounding on collateral accounts, ISDA negative rates protocol
- payment reversal in swaps under ISDA and DRV
- floored coupons for bonds, schuldscheindarlehen, loans, ...

Implications on pricing

- rate curves should allow for negative forwards
- lognormal models can not reproduce market prices for zero (or negative strike) floors
- lognormal can even fail to produce high enough prices for forward levels like $F = 1\%$ or 2%, because e.g. for shifted lognormal models with shift $d \geq 0$, $c(K)/N(0) \to F + d$ if $\sigma \to \infty$. You could actually observe this recently by first exploding, then missing implied lognormal volatility quotes for EUR swaptions with long option tenor

Implications on pricing

- shifted Black76 and normal Black76 models were established as market models for low and negative rates
- shifting is generic, e.g. the shifted SABR model has also become part of the new basic standard of market models
- with a different motivation (produce skew) a shift was introduced in Libor forward models a long time ago
- new models / model variants are discovered to handle negative rates in a more sophisticated way (free boundary SABR, mixed SABR)
- other models need adjustments as well (cms replication coupon pricers, Markov functional model)

Negative rates switch

- QL_NEGATIVE_RATES
- allows for negative zero yields, forwards, increasing discount factors

```
+2012-07-31 14:11  Ferdinando Ametrano
+
+ * [r18305] ql/userconfig.hpp, test-suite/piecewiseyieldcurve.cpp:
+
+   defaulted to allow negative rates (define QL_NEGATIVE_RATES) as this
+   is happening for EUR OIS, CHF and German treasury yields, etc.
```

Volatility type

- ql/termstructures/volatility/volatilitytype.hpp
- distinguishes between normal and (shifted) lognormal volatilities

```
enum VolatilityType { ShiftedLognormal, Normal };
```

Cap Floor Volatilities

- market quotes normal or shifted lognormal volatilities, with a constant shift across strikes and tenors

```
OptionletStripper(const boost::shared_ptr<CapFloorTermVolSurface>&,
                  const boost::shared_ptr<IborIndex>& iborIndex_,
                  const Handle<YieldTermStructure>& discount =
                                          Handle<YieldTermStructure>(),
                  const VolatilityType type = ShiftedLognormal,
                  const Real displacement = 0.0);
```

Swaption Volatilities

- market quotes normal or shifted lognormal volatilities, with different shifts per underlying
- swaption cubes inherit the shift structure from their embedded atm matrix
- swaption volatility cube 1 uses shifted SABR models
- the shift is bilinearly interpolated in (option, underlying) space

```
SwaptionVolatilityMatrix(
            const Calendar& calendar,
            BusinessDayConvention bdc,
            ...
            const VolatilityType type = ShiftedLognormal,
            const std::vector<std::vector<Real> >& shifts
                        = std::vector<std::vector<Real> >());
```

Libor in arrears adjustments

- convexity adjustment is amended in a straightforward way for shifted lognormal or normal volatilities
- timing adjustment is generalized at the same time for arbitrary non-natural fixing times[1]

```
enum TimingAdjustment { Black76,
                        BivariateLognormal };
BlackIborCouponPricer(const Handle<OptionletVolatilityStructure> &v =
                      Handle<OptionletVolatilityStructure>(),
                      const TimingAdjustment timingAdjustment = Black76,
                      const Handle<Quote> correlation =
                      Handle<Quote>(boost::make_shared<SimpleQuote>(1.0)))
```

[1] see http://ssrn.com/abstract=2170721

Linear TSR pricer

- volatility type is recognized through the abstraction of `SmileSection`
- the replication range is shifted by the appropriate (i.e. user bounds set to $[0, 200\%]$ and transformed to $[-1\%, 199\%]$ automatically if the applicable shift is 1% (to keep the user input universal under changing shifts in market quotations)

CMS Spread Option pricer

- swap rate adjustments use shifted lognormal or normal smiles to determine the drifts of the single swap rate models
- the bivariate model for the swap rates is still purely lognormal currently
- with negative 2Y fixings, we will neeed to extend this pricer as well
- plan: allow for shifts in the single rate models or for normal single rate models

Calibration Helper

- can be set up with normal and shifted lognormal volatilities
- cooperative with `HullWhite`, `Gsr`, `Lgm`, `MarkovFunctional` models

```
SwaptionHelper(const Period& maturity,
               const Period& length,
               const Handle<Quote>& volatility,
               ...
               const VolatilityType type = ShiftedLognormal,
               const Real shift = 0.0);
```

Markov Functional

- replicates a market smile / density per expiry via the numeraire calibration
- therefore also replicates the density for negative strike ranges
- currently, only shifted lognormal smile input allowed
- todo: allow normal smile input for numeraire calibration

Questions / Discussion

thank you for your attention