QuantLib Erlkönige

Peter Caspers

IKB

December 4th 2014

Table of contents

CMS Spread Coupons

Still missing: a coupon class which models cms spread coupons

$$\tau(\text{CMS10y} - \text{CMS2y}) \tag{1}$$

possibly capped and / or floored.

Approach 1: Formula index

Introduce an artificial index derived from `InterestRateIndex`

```
SwapSpreadIndex(const std::string& familyName,
                const boost::shared_ptr<SwapIndex>& swapIndex1,
                const boost::shared_ptr<SwapIndex>& swapIndex2,
                const Real gearing1 = 1.0,
                const Real gearing2 = -1.0);
```

and build everything else on top of it as with the other coupons based on ibor or cms indexes.

Approach 1: Repairing the class hiearchy

Since the formula index does not have own fixings, we would have to adjust the index base class by adding

```
//! check if index allows for native fixings
virtual void checkNativeFixingsAllowed() {}
```

and forbid native fixings in formula based indices

```
//! check if index allows for native fixings
virtual void checkNativeFixingsAllowed() {}
void checkNativeFixingsAllowed() {
    QL_FAIL("native fixings not allowed in swap spread index, refer to "
            "underlying indices instead");
}
```

Approach 2: Construct coupons with two swap indexes

If two swap indexes are used to construct a cms spread coupon we would need a more flexible way to construct floating legs, since

```
template <typename InterestRateIndexType,
          typename FloatingCouponType,
          typename CappedFlooredCouponType>
Leg FloatingLeg(const Schedule& schedule,
                const std::vector<Real>& nominals,
                const boost::shared_ptr<InterestRateIndexType>& index,
                const DayCounter& paymentDayCounter,
                BusinessDayConvention paymentAdj,
                const std::vector<Natural>& fixingDays,
                const std::vector<Real>& gearings,
                const std::vector<Spread>& spreads,
                const std::vector<Rate>& caps,
                const std::vector<Rate>& floors,
                bool isInArrears, bool isZero) {
```

only allows for one index.

Approach 2: Coupon Factories

We could introduce a factory instead of the template parameters

```
Leg FloatingLeg(const FloatingCouponFactory& factory,
                const Schedule& schedule,
                ...
```

which can generate plain, capped / floored and digital couons for the ibor, cms, cms spread flavours.

CMS Spread Coupons - Summary

- Introducing a formula based index would not exactly fit the semantics of the Index class. We would have to distinguish between native indexes (with own fixings) and derived ones.
- Using two indexes in the spread coupon class forces to introduce a more flexible way to construct floating legs, e.g. via factories.

OIS Curves Helpers

OIS Curves Helpers

Linear TSR CMS Coupon Pricer

Gaussian1d Models

No Arbitrage SABR

ZABR, BDK, Kahale, SVI

Simulated Annealing

Runge Kutta ODE Solver

Dynamic Creator of Mersenne Twister

Thank you

Questions?