A gentle introduction to IR modeling

Peter Caspers

IKB

May 23, 2015

Table of contents

What it is all about

Price a derivative contract !

Constraints

Do that *consistently* with all observable relevant market prices.

Second Thoughts

What do you mean by “price” after all ?

1. the amount of money that can be realized in the market (“objective” price)
2. the amount of money that we would require to receive or pay to close the deal (“subjective” price)

... we will focus on the first kind of price.

Price Adjustments

1. CVA
2. DVA
3. FVA
4. KVA
5. MVA
6. TVA

... we exclude all of them in the following.

Randomness

We need a source of randomness to model the market which is (or appears to be) random.

Stochastic Processes 1

To model a random observable we assume $X(0) = x_0$ and

$$dX(t,\omega) = \mu(t,\omega)dt + \sigma(t,\omega)dW \tag{1}$$

which means, knowing $X(t_0)$ we can evolve forward by

$$X(t_0 + \Delta t) \approx X(t_0) + \mu(t)\Delta t + \sigma(t)Z \tag{2}$$

with $Z \sim N(0, \Delta t)$ a normal distributed random variable with zero mean and variance Δt.

Stochastic Processes 2

The processes above are the processes with *continuous* paths.
There is a wider class (Levy processes) which add *jumps*. They are also used in finance, but a bit less common.

Asset Classes

Usually we distinguish

1. Equity
2. FX
3. Interest Rates
4. Credit
5. Inflation
6. Commodity

Today we focus on Interest Rates.

Interest Rate Building Blocks

Quoted, liquidly traded instruments:

1. Cash deposits
2. Forward Rate Agreements
3. Vanilla Swaps
4. Short Futures
5. Bond Futures

and in addition options

1. (Ibor) Caps, Floors
2. Swaptions
3. Futures options

Market Models for Options

Prices for Caps, Floors and Swaptions are quoted using volatilities σ, assuming a black model for the underlying F, either in lognormal style

$$dF = F\sigma dW \tag{3}$$

or shifted lognormal style

$$dF = (F+\alpha)\sigma dW \tag{4}$$

or in normal style

$$dF = \sigma dW \tag{5}$$

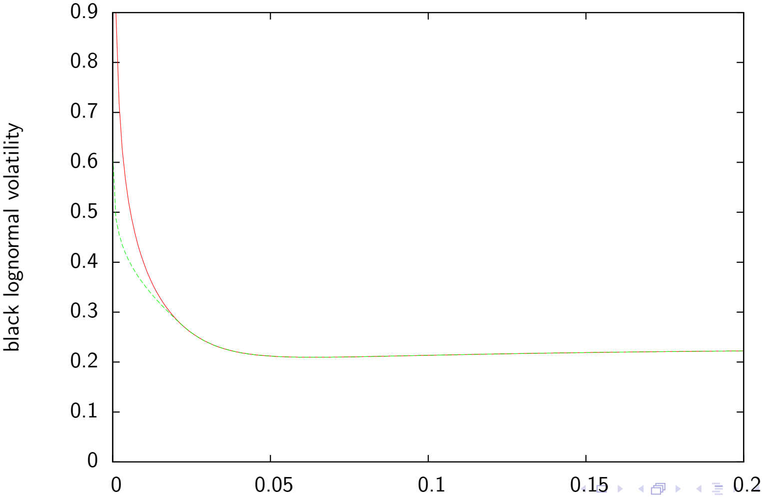
all of them having a simple closed form solution (Black formula).

Volatility smile

Options with different strike and maturity are priced using different volatilities. This seems weird at first sight (at least for the strike direction), but simply means that the Black model is more a quoting convention than a global model.
In strike direction this is referred to as the volatility smile. It simply says that the underlying distirbution is not (shifted) lognormal.

Volatility smile example

SABR 14y/1y implied black lognormal volatilities as of 14-11-2012, input (solid) and Kahale (dashed)



Underlying density

It is simple to derive the underlying density as

$$\phi(K) = \frac{\partial^2 c}{\partial K^2}(K) \tag{6}$$

for undiscounted call prices. The density is in the natural pricing measure (T-forward measure for caplets, Annuity measure for swaptions). This means you can get (in principle) the market density for an expiry time from a continuum of quoted option prices in strike direction.

No Arbitrage SABR Example

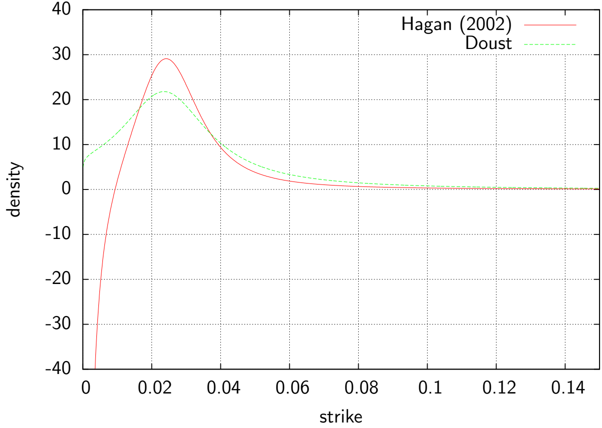


Figure : SABR smile $\alpha = 0.02,\ \beta = 0.40,\ \nu = 0.30,\ \rho = 0.30,\ \tau = 30.0,\ f = 0.03$

IR Smile modeling

In interest rates by far the most common single maturity smile model is the (shifted) SABR model

$$dF(t) = (F(t)+\alpha)^\beta \sigma(t) dW \tag{7}$$

$$d\sigma(t) = \sigma(t)\nu dV \tag{8}$$

$$dV dW = \rho dt \tag{9}$$

with parameters β (skew), $\alpha = \sigma(0)$ (initial level for σ), ν (volatility of volatility) and ρ (correlation between forward and vol process).

IR Smile modeling 2

Other models include

1. Heston
2. ZABR
3. SVI

Model independent pricing

Coupons of the following type can be priced only using a no arbitrage argument on today's yield curves:

1. fixed coupons
2. natural floating rate coupons

Natural means an IBOR - styled coupon with payment date = index accrual period end.

Single maturity model pricing

Coupons that depend on the distribution of an underlying variable on a single time instance, like

1. in arrears fixed IBOR coupons
2. capped / floored IBOR coupons
3. CMS coupons
4. CMS Spread Coupons

Usually the effective rate r for coupon estimation is then written

$$r = r_0 + c \tag{10}$$

as the sum of the zero volatility forward (i.e. the forward on today's curve) r_0 plus a convexity adjustment c.

Closed form convexity adjustments

Assuming a black model for the underlying one can derivate closed form approximations for convexity adjustments

1. timing adjustments
2. CMS adjustments

Replication CMS convexity adjustments

To incorporate the market volatility smile into convexity adjustments one uses the underlying density representation and integration by parts together with a simple model.

TSR CMS Coupon Pricers

A terminal swap rate (TSR) model is given by a mapping α

$$\alpha(S(t)) = \frac{P(t, t_P)}{A(t)} \tag{11}$$

where t_p is the coupon payment date and $A(t)$ the annuity of the underlying swap rate S. Then (integration by parts) the npv of a general CMS coupon $A(0)E^A(P(t,t_p)A(t)^{-1}g(S(t)))$ is given by

$$A(0)S(0)\alpha(S(0)) + \int_{-\infty}^{S(0)} w(k)R(k)dk + \int_{S(0)}^{\infty} w(k)P(k)dk \tag{12}$$

with t begin the fixing date of the coupon, R and P prices of market receiver and payer swaptions and weights $w(s) = \{\alpha(s)g(s)\}''$.

Hagan non parallel shifts model

In Hagan's classic paper, the model A.4 "non parallel shifts" corresponds to the following choice of α

$$\alpha(S) = \frac{S e^{-|h(t_p)-h(t)|x}}{1 - \frac{P(0,t_n)}{P(0,t)} e^{-|h(t_n)-h(t)|x}} \tag{13}$$

with t_n being the last payment date of the underlying swap and $h(s) - h(t) = \frac{1-e^{-\kappa(s-t)}}{\kappa}$ with a mean reversion parameter κ and x implicitly given by

$$S(t) \sum \tau_j P(0,t_j) e^{-|h(t_j)-h(t)|x} + P(0,t_n) e^{-|h(t_n)-h(t)|x} = P(0,t) \tag{14}$$

with τ_j, t_j being the yearfractions and payment dates of the fixed leg of the underlying swap.

Linear TSR model

The linear terminal swap rate model is defined by

$$\alpha(S) = as + b \tag{15}$$

b is determined by the no arbitrage condition

$$P(0, t_p)/A(0) = E^A(P(t, t_p)/A(t)) = aS(0) + b \tag{16}$$

a can be specified indirectly via a reversion κ by setting

$$a = \frac{\partial}{\partial S(t)} \frac{P(t, t_p)}{A(t)} \tag{17}$$

and evaluating the r.h.s. within a one factor gaussian model.

HJM model class

With a d - dimensional driving process we have

$$df(t,T) = \sigma_f(t,T)^t \int_t^T \sigma_f(t,u)du dt + \sigma_f(t,T)^t dW(t) \tag{18}$$

for instantaneous forward rates f with volatility σ_f in the risk neutral measure. The drift is fully specified by the volatilities.
This defines a wide class of interest rate models, on a theoretical level.

Special case: One factor Hull White

Any gaussian one factor HJM model which satisfies *separability*, i.e.

$$\sigma_f(t,T) = g(t)h(T) \tag{19}$$

for the instantaneous forward rate volatility with deterministic $g, h > 0$, necessarily fulfills

$$dr(t) = (\theta(t) - a(t)r(t))dt + \sigma(t)dW(t) \tag{20}$$

for the short rate r, which means, it is a Hull White one factor model.

The T-forward numeraire

Set $x(t) := r(t) - f(0,t)$ and fix a horizon T, then in the T-forward measure the numeraire can be written

$$N(t) = P(t,T) = \frac{P(0,T)}{P(0,t)} e^{-x(t)A(t,T)+B(t,T)} \tag{21}$$

with A, B dependent on the model parameters. The Hull White model is called an *affine* model.

Smile in the Hull White Model

- The distribution of $N(t)$ is lognormal. The shape of the distribution can not be controlled by any of the model parameters.
- For fixed t you can calibrate the model to one market quoted interest rate optoin (typically a caplet or swaption).
- You can choose the strike of the option, but the rest of the smile is implied by the model.

Callable vanilla swaps

Pricing of callable fix versus Libor swaps may be done in a Hull White model which is calibrated as follows:

- For each call date find a market quoted swaption which is equivalent to the call right (in some sense, e.g. by matching the npv and its first and second derivative of the underlying at $E(x(t))$).
- Calibrate the volatility function $\sigma(t)$ to match the basket of these swaptions.
- Choose the mean reversion of the model to control serial correlations.

Intertemporal correlations

To understand the role of the reversion parameter assume σ and a constant for a moment. Then it is easy to see

$$\text{corr}(x(T_1), x(T_2)) = \sqrt{\frac{e^{2aT_2} - 1}{e^{2aT_1} - 1}} = e^{-a(T_2 - T_1)} \sqrt{\frac{1 - e^{-2aT_1}}{1 - e^{-2aT_2}}} \tag{22}$$

which shows that for $a = 0$ the correlation is $\sqrt{T_1/T_2}$ and goes to zero if $a \to \infty$ and to one if $a \to -\infty$.

Callable cms swaps

The calll rights in a callable cms swap are options on a swap exchanging cms coupons against fix or Libor rates. Such underlying swaps are drastically mispriced in the Hull White model in general.

- cms coupons are replicated using swaptions covering the whole strike continuum $(0, \infty)$
- The swaption smile in the Hull White model is generally not consistent with the market smile and so are the prices of cms coupons

Obviously we need a more flexible model to price such structures

Model requirements

The wishlist for the model is as follows

- We want to be capable of calibrating to a whole smile of (constant maturity) swaptions, not only to one strike, for all fixing dates of the cms coupons. This is to match the coupons of the underlying.
- In addition we would like to calibrate to (possibly strike / maturity adjusted) coterminal swaptions to match the options representing the call rights.
- Finally we need some control over intertemporal correlations, i.e. something operating like the reversion parameter in the Hull White model

The idea to do so is to relax the functional dependency between the state variable x and the numeraire $N(t, x)$.

Markov Functional Model

We start with a markov process driving the dynamics of the model as follows:

$$dx = \sigma(t)e^{at}dW(t) \tag{23}$$

and $x(0) = 0$. The intertemporal correlation of the state variable x is the same as for the Hull White model, see (22), i.e. the parameter a can be used to control the correlation just as the reversion parameter in the Hull White model.

The numeraire surface

The model is operated in the T-forward measure, T chosen big enough to cover all cashflows relevant for the actual pricing under consideration. The link between the state $x(t)$ and the numeraire $P(t,T)$ is given by

$$P(t,T,x) = N(t,x) \tag{24}$$

which we allow to be a non parametric surface to have maximum flexibility in calibration.

Calibrating the numeriare surface to market smiles

The price of a digital swaption paying out an annuity $A(t)$ on expiry t if the swap rate $S(t) \geq K$ in our model is

$$\text{dig}_{\text{model}} = P(0,T) \int_{y^*}^{\infty} \frac{A(t,y)}{P(t,T)} \phi(y) dy \tag{25}$$

where y^* is the strike in the normalized state variable space (the correspondence between y and $S(t)$ is constructed to be monotonic).

Implying the swap rate

Given the market smile of $S(t)$ we can compute the market price $\text{dig}_{\text{mkt}}(\text{K})$ of digitals for strikes K. For given y^* we can solve the equation

$$\text{dig}_{\text{mkt}}(K) = P(0,T)\int_{y^*}^{\infty} \frac{A(t,y)}{P(t,T)}\phi(y)dy \qquad (26)$$

for K to find the swap rate corresponding to the state variable value y^*. For this $\text{dig}_{mkt}(\cdot)$ should be a monotonic function whose image is equal to the possible digital prices $(0, A(0)]$. We will revisit this later.

Computing the deflated annuity

To compute the deflated annuity

$$\frac{A(t)}{P(t,T)} = \sum_{k=1}^{n} \tau_k \frac{P(t,t_k)}{P(t,T)} \tag{27}$$

we observe that

$$\left.\frac{P(t,u)}{P(t,T)}\right|_{y(t)} = E\left(\frac{1}{P(u,T)}\middle| y(t)\right) \tag{28}$$

i.e. we have to integrate the reciprocal of the numeraire at future times. Working backward in time we can assume that we know the numeraire at these times (starting with $N(T) \equiv 1$).

Converting swap rate to numeraire

Having computed the swap rate $S(t)$ we have to convert this value to a numeraire value $N(t)$. Since

$$S(t)A(t) + P(t,t^*) = 1 \tag{29}$$

we get (by division by $N(t)$)

$$N(t) = \frac{1}{S(t)\frac{A(t)}{N(t)} + \frac{P(t,t^*)}{N(t)}} \tag{30}$$

all terms on the right hand side computable via deflated zerobonds as shown above. Note that we use a slightly modified swap rate here, namely one without start delay.

Calibration to a second instrument set

Up to now we have not made use of the volatility $\sigma(t)$ in the driving markov process of the model. This parameter can be used to calibrate the model to a second instrument set, however only a single strike can be matched obviously for each expiry. A typical set up would be

- calibrate the numeraire to an underlying rate smile, e.g. constant maturity swaptions for cms coupon pricing
- calibrate $\sigma(t)$ to (standard atm or possibly adjusted) coterminal swaptions for call right calibration

Note that after changing $\sigma(t)$ the numeraire surface needs to be updated, too.

Input smile preconditioning

To ensure a bijective mapping

$$\mathsf{dig}_{\mathsf{mkt}} : (0, \infty) \to (0, A(0)) \tag{31}$$

it is sufficient to have an arbitrage free input smile with a C^1 call price function. It is possible to allow for negative rates and generalize the interval $(0, \infty)$ to $(-\kappa, \infty)$ with some suitable $\kappa > 0$, e.g. $\kappa = 1\%$. In general input smiles are not arbitrage free, so some preconditioning is advisable, since arbitrageable smiles will break the numeraire calibration.

A full example: Market Data

```
#include <ql/quantlib.hpp>
using namespace QuantLib;

int main(int, char * []) {

    try {

        Date refDate(13, November, 2013);
        Date settlDate = TARGET().advance(refDate, 2 * Days);
        Settings::instance().evaluationDate() = refDate;

        Handle<Quote> rateLevel(new SimpleQuote(0.03));
        Handle<YieldTermStructure> yts(
            new FlatForward(refDate, rateLevel, Actual365Fixed()));

        boost::shared_ptr<IborIndex> iborIndex(new Euribor(6 * Months, yts));
        boost::shared_ptr<SwapIndex> swapIndex(
            new EuriborSwapIsdaFixA(10 * Years, yts));

        iborIndex->addFixing(refDate, 0.0200);
        swapIndex->addFixing(refDate, 0.0315);

        Handle<Quote> volatilityLevel(new SimpleQuote(0.30));
        Handle<SwaptionVolatilityStructure> swaptionVol(
            new ConstantSwaptionVolatility(refDate, TARGET(), Following,
                                           volatilityLevel, Actual365Fixed()));
```

A full example: Cms Swap and Exercise Schedule

```
Date termDate = TARGET().advance(settlDate, 10 * Years);

Schedule sched1(settlDate, termDate, 1 * Years, TARGET(),
                ModifiedFollowing, ModifiedFollowing,
                DateGeneration::Forward, false);
Schedule sched2(settlDate, termDate, 6 * Months, TARGET(),
                ModifiedFollowing, ModifiedFollowing,
                DateGeneration::Forward, false);

Real nominal = 100000.0;
boost::shared_ptr<FloatFloatSwap> cmsswap(new FloatFloatSwap(
    VanillaSwap::Payer, nominal, nominal, sched1, swapIndex,
    Thirty360(), sched2, iborIndex, Actual360(),
    false,false,1.0,0.0,Null<Real>(),Null<Real>(),1.0,0.00267294));

std::vector<Date> exerciseDates;
std::vector<Date> sigmaSteps;
std::vector<Real> sigma;

sigma.push_back(0.01);
for (Size i = 1; i < sched1.size() - 1; i++) {
    exerciseDates.push_back(swapIndex->fixingDate(sched1[i]));
    sigmaSteps.push_back(exerciseDates.back());
    sigma.push_back(0.01);
}
```

A full example: Call Right

```
boost::shared_ptr<Exercise> exercise(
    new BermudanExercise(exerciseDates));
boost::shared_ptr<FloatFloatSwaption> callRight(
    new FloatFloatSwaption(cmsswap, exercise));

std::vector<Date> cmsFixingDates(exerciseDates);
std::vector<Period> cmsTenors(exerciseDates.size(), 10 * Years);
```

A full example: Models and Engines

```
Handle<Quote> reversionLevel(new SimpleQuote(0.02));

boost::shared_ptr<NumericHaganPricer> haganPricer(
    new NumericHaganPricer(swaptionVol,
                           GFunctionFactory::NonParallelShifts,
                           reversionLevel));
setCouponPricer(cmsswap->leg(0), haganPricer);

boost::shared_ptr<MarkovFunctional> mf(new MarkovFunctional(
    yts, reversionLevel->value(), sigmaSteps, sigma, swaptionVol,
    cmsFixingDates, cmsTenors, swapIndex));

boost::shared_ptr<Gaussian1dFloatFloatSwaptionEngine> floatEngine(
    new Gaussian1dFloatFloatSwaptionEngine(mf));

callRight->setPricingEngine(floatEngine);
```

A full example: Calibration Basket

```
boost::shared_ptr<SwapIndex> swapBase(
    new EuriborSwapIsdaFixA(30 * Years, yts));

std::vector<boost::shared_ptr<CalibrationHelper> > basket =
    callRight->calibrationBasket(swapBase, *swaptionVol,
                                 BasketGeneratingEngine::Naive);

boost::shared_ptr<Gaussian1dSwaptionEngine> stdEngine(
    new Gaussian1dSwaptionEngine(mf));

for (Size i = 0; i < basket.size(); i++)
    basket[i]->setPricingEngine(stdEngine);
```

A full example: Model Calibration and Pricing

```
        LevenbergMarquardt opt;
        EndCriteria ec(2000, 500, 1E-8, 1E-8, 1E-8);
        mf->calibrate(basket, opt, ec);

        std::cout << "model vol & swaption market & swaption model \\\\" << std::endl;
        for (Size i = 0; i < basket.size(); i++) {
            std::cout << mf->volatility()[i] << " & "
                      << basket[i]->marketValue() << " & "
                      << basket[i]->modelValue() << " \\\\" << std::endl;
        }
        std::cout << mf->volatility().back() << std::endl;

        Real analyticSwapNpv = CashFlows::npv(cmsswap->leg(1), **yts, false) -
                               CashFlows::npv(cmsswap->leg(0), **yts, false);
        Real callRightNpv = callRight->NPV();
        Real firstCouponNpv = - cmsswap->leg(0)[0]->amount() * yts->discount(cmsswap->leg(0)[0]->date()) +
                              cmsswap->leg(1)[0]->amount() * yts->discount(cmsswap->leg(1)[0]->date());
        Real underlyingNpv = callRight->result<Real>("underlyingValue") + firstCouponNpv;

        std::cout << "Swap Npv (Hagan)     & " << analyticSwapNpv << "\\\\" << std::endl;
        std::cout << "Call Right Npv (MF)  & " << callRightNpv << "\\\\" << std::endl;
        std::cout << "Underlying Npv (MF)  & " << underlyingNpv << "\\\\" << std::endl;
        std::cout << "Model trace : " << std::endl << mf->modelOutputs() << std::endl;
    }
    catch (std::exception &e) {
        std::cerr << e.what() << std::endl;
        return 1;
    }
}
```

A full example: Model Calibration (Coterminals)

model vol	swaption market	swaption model
0.01	0.0273707	0.0273706
0.0107835	0.0337373	0.0337373
0.0106409	0.0354739	0.0354741
0.0109936	0.0344716	0.0344715
0.0109297	0.0315097	0.0315096
0.0111361	0.0270958	0.0270957
0.0111917	0.0215014	0.0215015
0.0112365	0.0150493	0.0150493
0.0113241	0.00783272	0.00783278
0.00998595		

A full example: Pricing

The expectation is to get a similar price of the underlying cms swap both in the Markov and the replication model, since both are consistent with the input swaption smile. Note that the underlying of the swaption is receiving the cms side while the cms swap is paying.

Swap Npv (Hagan)	-0.00
Call Right Npv (MF)	604.50
Underlying Npv (MF)	1.00

The match is very close (0.1 bp times the nominal). It should be noted that from theory we can not even expect a perfect match since the rate dynamics is not the same in the Hagan model and the Markov model respectively.

A full example: Benchmarking against Hull White

We change the code to replace the Markov model with a Hull White model. This is easily done by

```
std::vector<Date> sigmaSteps2(sigmaSteps.begin(), sigmaSteps.end() - 1);
std::vector<Real> sigma2(sigma.begin(), sigma.end() - 1);
boost::shared_ptr<Gsr> gsr(new Gsr(yts,sigmaSteps2,sigma2,reversionLevel->value()));
```

and replacing `mf` by `gsr` (we could have used a generic name of course...). The calibration now reads

```
gsr->calibrate(basket, opt, ec, Constraint(), std::vector<Real>(), model->FixedReversions());
```

or also

```
gsr->calibrateVolatilitiesIterative(basket, opt, ec);
```

A full example: Pricing in the Hull White model

The fit to the coterminals is exact, just as above (with different model volatilities of course). The pricing compares ot the Markov model as follows:

Swap Npv (Hagan)	-0.00	-0.00
	Markov	Hull White
Call Right Npv	604.50	1012
Underlying Npv	1.00	657.81

The underlying and the option are drastically overpriced (both) by over 60bp in the Hull White model.

A full example: A more realistic smile surface

Swaption smiles are usually far from being flat. We test the model with a SABR volatility cube (with constant parameters $\alpha = 0.15$, $\beta = 0.80$, $\nu = 0.20$, $\rho = -0.30$) by setting

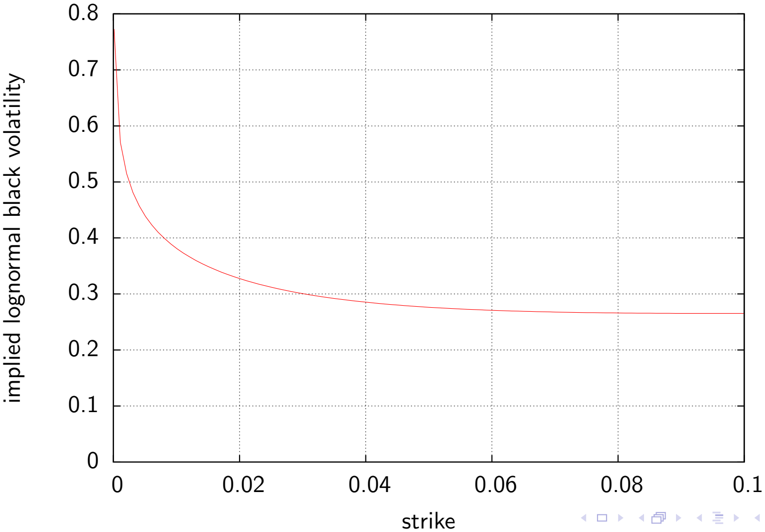
```
Handle<SwaptionVolatilityStructure> swaptionVol(
    new SingleSabrSwaptionVolatility(refDate, TARGET(), Following, 0.15,
                                     0.80, -0.30, 0.20,
                                     Actual365Fixed(), swapIndex));
```

Swap Npv (Hagan)	246.00	246.00
	Markov	Hull White
Call Right Npv	696.96	1004.79
Underlying Npv	259.96	648.26

Again the underlying and the option is overpriced (by 40bp resp. 53bp) in the Hull White model. In the Markov model the fit is still good (1.4bp underlying price difference).

A full example: A more realistic smile surface ctd

Example smile 10y into 10y, $\alpha = 0.15$, $\beta = 0.80$, $\nu = 0.20$, $\rho = -0.30$



A full example: Model diagnostics

The markov model can generate information on the calibration process with

```
std::cout << "Model trace : " << std::endl << mf->modelOutputs()
          << std::endl;
```

The output contains information on

- numerical model parameters
- settings for smile preconditioning
- yield term structure calibration results
- volatility smile calibration results

and can help identifying calibration problems, resp. confirm a successful calibration.

A full example: Model diagnostics (model parameters / smile settings)

```
Markov functional model trace output
Model settings
Grid points y          : 64
Std devs y             : 7
Lower rate bound       : 0
Upper rate bound       : 2
Gauss Hermite points : 32
Digital gap            : 1e-05
Adjustments            : Kahale SmileExp
Smile moneyness checkpoints:
```

A full example: Model diagnostics (yield term structure fit)

The raw output

```
Yield termstructure fit:
expiry;tenor;atm;annuity;digitalAdj;ytsAdj;marketzerorate;modelzerorate;diff(bp)
November 13th, 2014;10Y;0.03047961644430512;8.267635590910791;1;1;0.03000000000000008;0.03008016185429107;-0.8
November 12th, 2015;10Y;0.0304803836917478;8.023747530203391;1;1;0.02999999999999999;0.03003959599137232;-0.39
[...]
```

is meant to be analyzed in another application like office

	A	B	C	D	E	F	G	H	I
1	expiry	tenor	atm	annuity	digitalAdj	ytsAdj	marketzerorate	modelzerorate	diff(bp)
2	November 13th, 2014	10Y	0.03048	8.26764	1.00000	1.00000	0.03000	0.03008	-0.80162
3	November 12th, 2015	10Y	0.03048	8.02375	1.00000	1.00000	0.03000	0.03004	-0.39596
4	November 11th, 2016	10Y	0.03047	7.78712	1.00000	1.00000	0.03000	0.03003	-0.26059
5	November 13th, 2017	10Y	0.03047	7.55030	1.00000	1.00000	0.03000	0.03002	-0.20398
6	November 13th, 2018	10Y	0.03048	7.32698	1.00000	1.00000	0.03000	0.03001	-0.13839
7	November 13th, 2019	10Y	0.03048	7.11026	1.00000	1.00000	0.03000	0.03001	-0.12939
8	November 12th, 2020	10Y	0.03047	6.90704	1.00000	1.00000	0.03000	0.03001	-0.09758
9	November 11th, 2021	10Y	0.03047	6.70335	1.00000	1.00000	0.03000	0.03001	-0.14477
10	November 11th, 2022	10Y	0.03048	6.50203	1.00000	1.00000	0.03000	0.03002	-0.15427
11	November 15th, 2023	1Y	0.03054	0.72444	1.00000	1.00000	0.03000	0.03001	-0.11838
12	November 15th, 2024	1Y1M	0.03038	0.76312	1.00000	1.00000	0.03000	0.03001	-0.10332
13	December 19th, 2024	11M	0.03045	0.64848	1.00000	1.00000	0.03000	0.03001	-0.10074
14	November 17th, 2025	1Y	0.03045	0.68031	1.00000	1.00000	0.03000	0.03001	-0.08105
15	November 16th, 2026	1Y	0.03045	0.66026	1.00000	1.00000	0.03000	0.03001	-0.05946
16	November 15th, 2027	1Y	0.03054	0.64075	1.00000	1.00000	0.03000	0.03000	-0.04400
17	November 15th, 2028	1Y	0.03045	0.62514	1.00000	1.00000	0.03000	0.03001	-0.08793
18	November 15th, 2029	1Y	0.03045	0.60667	1.00000	1.00000	0.03000	0.03001	-0.11201
19	November 15th, 2030	1Y1M	0.03038	0.63736	1.00000	1.00000	0.03000	0.03001	-0.08345
20	December 19th, 2030	11M	0.03045	0.54161	1.00000	1.00000	0.03000	0.03001	-0.08143
21	November 17th, 2031	1Y	0.03054	0.56815	1.00000	1.00000	0.03000	0.03001	-0.10500
22									

A full example: Model diagnostics (volatility smile fit)

	A	B	C	D	E	F	G	H	I	J
1	strike(November 13th, 2014/10Y)	marketCallRaw	marketCall(No	modelCall(Nov	marketPutRaw	marketPut(No	modelPut(Nov	marketVega(N	strike(November 12th, 2015/10Y)	marketCall
2	0.000%	25.199%	25.199%	25.199%	0.000%	0.000%	0.000%	0.000%	0.000%	24.4
3	0.030%	24.947%	24.947%	24.947%	0.000%	0.000%	0.000%	0.000%	0.030%	24.2
4	0.152%	23.939%	23.939%	23.939%	0.000%	0.000%	0.000%	0.000%	0.152%	23.2
5	0.305%	22.679%	22.679%	22.679%	0.000%	0.000%	0.000%	0.000%	0.305%	22.0
6	0.762%	18.900%	18.900%	18.900%	0.000%	0.000%	0.000%	0.000%	0.762%	18.3
7	1.219%	15.132%	15.132%	15.132%	0.012%	0.012%	0.012%	0.003%	1.219%	14.7
8	1.524%	12.655%	12.655%	12.655%	0.056%	0.056%	0.055%	0.010%	1.524%	12.5
9	1.829%	10.262%	10.262%	10.262%	0.182%	0.182%	0.182%	0.024%	1.829%	10.4
10	2.134%	8.028%	8.028%	8.029%	0.468%	0.468%	0.468%	0.045%	2.134%	8.4
11	2.438%	6.037%	6.037%	6.037%	0.998%	0.998%	0.996%	0.069%	2.438%	6.7
12	2.743%	4.357%	4.357%	4.358%	1.837%	1.837%	1.837%	0.089%	2.743%	5.3
13	3.048%	3.018%	3.018%	3.018%	3.018%	3.018%	3.017%	0.099%	3.048%	4.1
14	3.810%	1.022%	1.022%	1.020%	7.321%	7.321%	7.318%	0.083%	3.810%	2.0
15	4.572%	0.286%	0.286%	0.286%	12.886%	12.886%	12.883%	0.043%	4.572%	0.9
16	5.334%	0.071%	0.071%	0.071%	18.970%	18.970%	18.967%	0.017%	5.334%	0.4
17	6.096%	0.016%	0.016%	0.016%	25.216%	25.216%	25.213%	0.005%	6.096%	0.1
18	15.240%	0.000%	0.000%	0.000%	100.798%	100.798%	100.789%	0.000%	15.240%	0.0
19	22.860%	0.000%	0.000%	0.000%	163.796%	163.796%	163.784%	0.000%	22.860%	0.0
20	30.480%	0.000%	0.000%	0.000%	226.795%	226.795%	226.778%	0.000%	30.480%	0.0
21	45.719%	0.000%	0.000%	0.000%	352.792%	352.792%	352.766%	0.000%	45.721%	0.0
22	60.959%	0.000%	0.000%	0.000%	478.789%	478.789%	478.755%	0.000%	60.961%	0.0

Libor Market Models

Libor Forward Model with displaced diffusion / of Heston type, with forward rate dynamics

$$
\begin{aligned}
dF_i(t) &= (F_i(t) + d_i)\phi_i\sigma(T_i - t)\sqrt{z(t)}dW_i(t) && (32)\\
\sigma(\tau) &= (a + b\tau)e^{-c\tau} + d && (33)
\end{aligned}
$$

with time homogeneous abcd - volatility.

Heston-LFM Correlation Structure

The correlation structure is given by $dW_i dW_j = \rho_{i,j} dt$ with

$$\rho_{i,j} = \rho_{i,j}^k = \rho_\infty + (1 - \rho_\infty) e^{-\beta |(T_i - t_k) - (T_j - t_k)|} \tag{34}$$

which allows for a wide range of forward rate correlations through suitable choice of the long term correlation ρ_∞ and the decay factor β.

Heston-LFM Stochastic Volatility

The volatility part can be made stochastic by

$$dz(t) = \theta(1 - z(t))dt + \eta\sqrt{z(t)}dV(t) \tag{35}$$

Models

This is a list of some possible pricing models and their ability to adapt to market parameters.

Model	Murex	QuantLib	Smile	CMS Margin	Decorrelation
HW1F	yes	yes	poor	poor	no
MF1F	yes	yes	excellent	excellent	no
LFM	under dev	yes	poor	poor	excellent
DD-LFM	under dev	yes	good	poor	excellent
Heston-LFM	?	yes	good	poor	excellent
SABR-LFM	no	no	excellent	excellent	excellent

In Particular the 1F models always imply a correlation of 100% between rates. As we will see, lower correlation implies a higher option price.

Summary

1. Basic / Single Maturity Models
2. HJM class
3. Hull White Model
4. Markov Functional Model
5. Libor Market Models
6. DD LFM, Heston Volatility

Outlook

1. Multi factor gaussian models
2. Quasi gaussian models (allow g above to be stochastic)
3. SABR LFM
4. ...

Questions / Discussion

visit my blog http://quantlib.wordpress.com