Interest Rate Derivatives Analytics

Version 11.0

June 2009 — First Edition

The purpose of the Calypso Financial Analytics guide for Interest Rate Derivatives is to provide an understanding of the analytics that underlie the pricing of IRD trades.

Revision date	Edition	Comments
June 2009	First edition	First edition for version 11.0.

Contents

Section 1. Market Data Generation

1.1 Interpolation Methods

Interpolation implementations usually are defined to be applied to any quantities, such as interest rates or discount factors, without regard to their type. Thus they are purely mathematical, and the following sections specify the algorithms.

When used with yield curve generators, another parameter is required to specify whether the interpolation is on the discount factors or on the zero rates. This parameter is named **INT_CURVE_INTERP_RATE_B** and is an environment parameter set when the system is launched. Bring up the User Env application, and under Properties set **INT_CURVE_INTERP_RATE_B** to one of the following:

Y : yield curves will interpolate on zero rates, using whatever interpolator is chosen

N : yield curves will interpolate on discount factors, using whatever interpolator is chosen.

1.1.1 InterpolatorLinear

For value x that lies between x1 and x2 the interpolated value y for given ordinate values y1 and y2 is given by:

$$y(x) = A\,y_1 + B\,y_2 \qquad \text{with} \qquad A = \frac{x_2 - x}{x_2 - x_1} \qquad \text{and} \qquad B = \frac{x - x_1}{x_2 - x_1}$$

1.1.2 InterpolatorLogLinear

For value x that lies between x1 and x2 the interpolated value y for given ordinate values y1 and y2 is given by:

$$y(x) = e^{\ln(y_1) + \frac{(\ln(y_2) - \ln(y_1))(x - x_1)}{x_2 - x_1}}$$

1.1.3 InterpolatorSpline

We implemented the cubic spline algorithm as described in Numerical Recipes 2nd edition (NR), pp.115; the only difference to the NR code is that we allow having an array of ordinates to be processed at once rather than one by one. So in one computation, we can process for example an array of curves rather than a single curve and upon interpolation we will be returned an array of (interpolated) rates rather than a single rate.

1.1.4 Interpolator3DLinear

We implemented the multi-dimensional bilinear interpolation algorithm as described in *Numerical Recipes 2nd edition (NR),* pp.123. Our implementation is for 3 dimensions only.

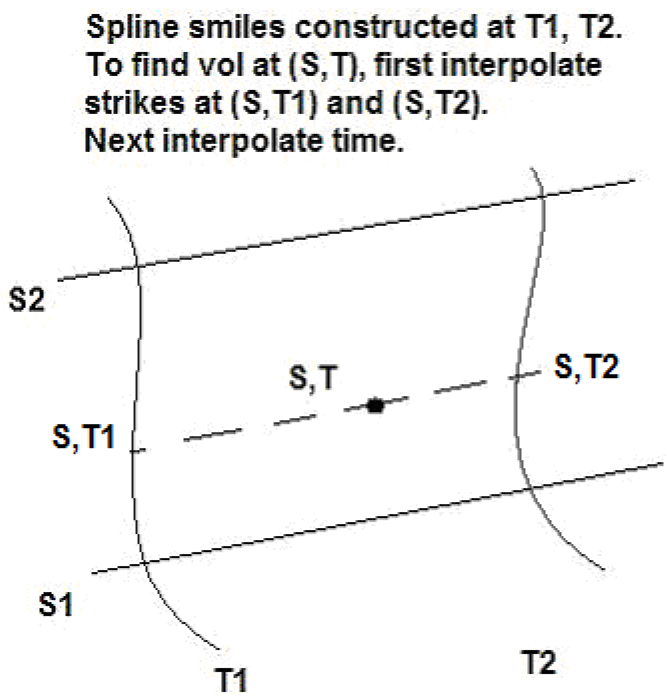
1.1.5 Interpolator3DSpline1D

Same as Interpolator3DLinear except that the points on the third axis are interpolated using a cubic spline, while the others interpolated linearly. Constant extrapolation is used beyond the known points in both dimensions. The most frequent application is to spline interpolation of strikes for volatility surfaces.

For example, for FX options, the spline is used for the delta (strike) axis while linear interpolation for the expiry time axis.The procedure is illustrated below. The "strike" axis for FX options is the delta axis. Define

V(S,T) : Volatility at delta strike S and expiry time T

For each expiry time in the surface, a spline is constructed along the delta axis.



To find the volatility at any delta and expiry (S,T), the following procedure is used:

1. First existing points (S1, T1) and (S2, T2) are found on the surface that obey:

   S1 <= S < S2, T1 <= T < T2.

2. Splines in the strike dimension exist at the times T1 and T2. Each spline is interpolated upon individually to find the volatility at the strike S for each of those times. The result is two volatilities, V(S, T1) and V(S, T2).
3. Linear interpolation in the time direction is performed on V(S,T1) and V(S,T2) to obtain V(S,T).

1.1.6 InterpolatorDailyCompFwdRate

InterpolatorDailyCompFwdRate is designed specifically for yield curves and implements Compound Daily Forward interpolation. This method finds discount factors on a date between two dates with known discount factors by assuming the forward rate between the known dates is constant. It is equivalent to log linear interpolation on discount factors. Thus the interpolator ignores the value **INT_CURVE_INTERP_RATE_B** described previously.

InterpolatorDailyCompFwdRate is the preferred interpolator for use with BootstrapForwards curve generation.

The Compound Daily Forward interpolation method proceeds as follows Let $D(t_1)$ and $D(t_2)$ be the known discount factors on dates t_1 and t_2, respectively. We want to find the discount factor on a date t that falls between t_1 and t_2.

The assumption is that for each day between t_1 and t_2 there is a constant rate at which interest will compound. That is, there is a daily rate *f* so that:

$$D(t_1)\frac{1}{(1+f)^{(t_2-t_1)}} = D(t_2).$$

The difference $t_2 - t_1$ is expressed in days. To interpolate on t, first find this implied rate *f* from this equation. Then use that rate to find the discount factor on t:

$$D(t) = D(t_1)\frac{1}{(1+f)^{(t-t_1)}}.$$

Or, more directly, one has:

$$D(t) = \left(D(t_1)^{(t_2-t)} D(t_2)^{(t-t_1)}\right)^{1/(t_2-t_1)}.$$

1.2 Interest Rate Curves

1.2.1 Zero Curve

Bootstrap Generator

The Bootstrap Generator generates a zero curve based on user-selected underlying instruments: Money Market, FRA, Future, Swap, Spread, Turn Rate, Bond. The algorithm supports future convexity correction (manual spreads or using correlation-driven adjustments), and the generation of additional curve points at each flow date of multi-period instruments (swaps).

Calypso orders the instruments, first adding single-period rate instruments and then multi-period “priced” instruments; single-period instruments with forward end dates falling on or after the maturity of any multi-period instrument are excluded from the bootstrapping procedure.

Single-Period Instruments

A single-period instrument has one continuous interest period associated with it, defined by start and end dates. A quote for the instrument defines the interest rate applicable between those dates.

Money Market underlyings have an interest start date on the Spot Date, usually two business days after the curve date; the end date is found by adding the tenor of the underlying to the Spot Date and adjusting for holidays and weekends, according to the convention defined in the underlying. To cover the period between the curve date and the Spot Date, one usually begins a curve with an O/N (overnight) rate and possibly also a T/N (tomorrow next) rate; otherwise, the rate over the gap will need to be extrapolated.

Interest Rate Futures underlyings are derivatives of an underlying term deposit that starts on the Spot Date after the futures settlement date and continues for the tenor of the deposit. Note that the actual days of the deposit are used for curve generation. Some market practitioners simplify the yield curve calculation by assuming the futures rate applies to the period between futures settlement dates. As this produces inaccuracies, Calypso does not employ this approximation.

Once the forward rates and their interest periods are determined, they must be combined to calculate the discount factors that will define the curve. This requires the handling of gaps and overlaps between the interest periods. The section "Gaps and Overlaps of Single-Period Instruments" (see below) describes the handling of these periods in detail.

Multi-Period Instruments

In generating with multi-period instruments such as swaps and bonds, Calypso adds only a single point to the curve at the last date on which a discount factor needs to be known to price the instrument. This last date is usually the maturity date, but it may be the end date of the last forward rate period, as a discount factor is needed on that date in order to project the forward rate. Discount factors on all points prior to the last added point are interpolated as needed using the curve’s interpolation method.

The generation procedure for a swap is as follows:

1. A swap trade is created with the quoted rate as its fixed rate.
2. A guess is made for the discount factor as of the last date of the instrument. This creates a candidate curve.
3. The swap is priced with the candidate curve to find Net Present Value implied by that curve.
4. From the calculated Net Present Value, an improved guess of the discount factor is made.
5. Steps 2 – 4 are repeated until a candidate curve produces a zero Net Present Value. More exactly, the computation halts when an improvement in the discount factor cannot be found the produces a more than 10^{-14} improvement in the Net Present Value.

In pricing a swap, there is a choice for handling the first floating cashflow if the reset date is the day of curve generation: one can either use the quoted reset rate for the curve date, or else project the rate from the candidate curve. This choice is determined by the "Manual First Reset" parameter on the swap. If the parameter is true, the value of the rate index from the quote set or the entry in the curve window will be set on the first floating cashflow; otherwise, that reset rate is removed from the quote set for the duration of the generation and its value not taken into account.

Spread Underlying

A Spread Underlying is a swap whose fixed rate is derived by adding a quoted spread to the yield of a specified bond. The quote for the Spread is in basis points. The attributes of the swap are defined on the Spread object and although the payment frequency, day count, and so on may differ from those of the bond, the market convention is that the quoted spread is not adjusted for those attributes. That is, one simply has:

Swap all-in fixed rate = Bond yield + Spread

without any further transformation of the rate according to day count, etc. By convention, market practitioners allow for these differing attributes when they provide a quote for the spread.

In Bootstrap Generation, the latest price for the bond is obtained and the yield calculated from it using the usual pricer for that bond (unless the yield is quoted directly, so that no calculation is needed). The spread is added to the yield and a swap with that fixed rate is created. Then the procedure for finding a discount factor at the end of the swap is performed as described under "Multi-Period Instruments."

The bond can be a relative benchmark bond – for example, the 10Y on-the-run US Treasury -- or a specific issue. Two bonds can be associated with the Spread as well. For the case of two bonds, the all-in rate is found as follows:

1. The yield is found for each bond using the bond pricer. Call these y_A and y_B.
2. Identify the bond time intervals to be used for interpolation. Call these t_A and t_B. They are found as follows:
   - If both bonds are relative bonds, so that they are associated with benchmark tenors – for example, 5Y and 7Y – those tenors are used as the time intervals for interpolation. Note these tenors will usually not be exactly the same as the remaining time to maturity of the bonds.
   - If at least one of the bonds is a specific issue, use the *original* maturity of each bond, at the time of issue. Note again this will not be the same as the remaining time to maturity of the bonds.
3. Identify the Spread time interval, which is the length of the tenor of the Spread definition – for example, 8Y, or 7Y6M. Call this t_S.
4. Calculate the interpolated yield for the Spread's tenor using linear interpolation:

$$y = y_A + (y_B - y_A)\frac{t_S - t_A}{t_B - t_A}.$$

5. Calculate the all-in swap rate by adding the spread:

Swap all-in fixed rate = y + Spread

The "Manual First Reset" parameter applies to a Spread underlying in the same fashion as for swaps. If set to true, the first floating cashflow of the swap will have its rate set equal to the reset rate of the floating rate index, if the reset date is the same as the curve generation date. Otherwise the floating rate will be projected from the curve.

Gaps and Overlaps of Single-Period Instruments

Each single-period instrument produces a "forward interval" consisting of two dates; the interval's discount factor between its start and end date is found from the market rate, producing a forward rate for this interval. When all the single-period instruments have been computed in this fashion the result is a set of forward intervals which can be arbitrarily related to each other in terms of gaps and overlaps.

These intervals are sorted by start date and secondarily by end date (if start dates are equal then the end dates are considered; if end dates are also equal, the earlier instrument in the curve specification list is given precedence); then the intervals are processed one by one. This checks the relation of each successive interval with the next earlier one and applies an appropriate method to eliminate any gap or overlap between them. When completed, the result is a set of contiguous, non-overlapping forward intervals, each one having a forward rate across its interval, which are then used to compute a zero curve.

The following is a list of the cases and how they are handled in the BootstrapGenerator. The earlier of the two intervals to be compared will be called Interval A, the later will be Interval B. Sorting guarantees that the start and end dates of Interval A are both less than or equal to the corresponding dates of Interval B. The discount factor between two dates t1 and t2 will be designated F(t1, t2), and the start date of Interval A is denoted A1, the end date A2; thus F(A1,A2) and F(B1, B2) are known inputs to the procedure.

First interval added to the curve. If there is a gap between the Interval A's start date and the curve's valuation (start) date, a new interval to cover the gap is created and the continuous ACT/365 forward rate of the first interval is applied to it. Thus this uses "pull-back constant forward interpolation."

Interval B has start and end dates both the same as Interval A. Interval B is discarded. The next interval after it will be compared to Interval A. (Note then that the resulting curve will probably not reproduce the input price of the instrument correctly that associated to the discarded interval.)

Interval B has the same start date as Interval A.

←-------A--------→←New→

←-------------B--------------→

Remove Interval B and insert an interval between the end date of Interval A and the end date of Interval B (the previous sorting and cases guarantee a positive time interval). The rate over the new interval is determined by standard arbitrage to retain both forward rates, F(A1, A2)*F(A2,B2) = F(B1, B2). This preserves the pricing of both instruments. (The picture above shows both the original A and B intervals and the newly created interval; after this step, the intervals left for curve building are the contiguous, non-overlapping intervals A and New.)

Interval B has the same end date as Interval A.

←-------------A---------------→

←New→←-------B---------→

Remove Interval A. Add a new interval from the start date of Interval A to the start date of Interval B (sorting and previous cases guarantee a positive time interval). The discount factor over the new interval is determined to preserve both forward rates: F(A1,B1)*F(B1,B2) = F(A1, A2).

Interval B is wholly contained within Interval A.

←-----------------A------------------→

←New→←------B------→←New→

Remove Interval A. Create two new intervals from the start of A to the start of B, and from the end of B to the end of A. Set the forward rate over the two new intervals to be equal, and solve for this rate so as to preserve the forward rates over A and B: F(A1,B1)*F(B1,B2)*F(B2,A2) = F(A1,A2).

Interval B overlaps the end of Interval A.

←-------------A---------------→

←--------------B--------------→

←---New----→←----New--→←---New----→

Remove both Intervals A and B. Replace with three new intervals, covering A1 to B1, B1 to A2, and A2 to B2. To do this there is a different procedure from the other methods, in that first Interval A is added to the earlier intervals and a temporary zero curve is generated out to Interval A's end date, A2. Using this temporary curve the discount factors from the curve's start date to the interval start dates A1 and B1 are found, i.e., F(0,A1), F(0, B1). Then the temporary curve is discarded.

The discount factors over the new intervals, F(A1,B1), F(B1,A2) , and F(B1,A2) intervals are determined by arbitrage: F(0,A1)*F(A1,B1)=F(0,B1); F(A1,B1)*F(B1,A2) = F(A1,A2); and F(B1,A2)*F(A2,B2) = F(B1,B2). Notice that this preserves the forward rates on Intervals A and B; the temporary curve only serves to provide a method for introducing a forward rate on the partial interval from A1 to B1.

Interval B starts after the end date of Interval A.

←----------A---------→←New→←-------B---------→

This is a gap. Create a new interval to fill the gap from A2 to B1. The forward rate over the new interval is set equal to the forward rate over Interval A. Thus this is a "pull forward constant forward rate extrapolation" method.

Summary

Note that all of these methods preserve the forward rates over existing intervals, and do not necessarily preserve the zero rates of the curve built so far. Adding instruments to a curve one by one you have the alternative of finding discount factors using the curve's interpolation method rather than preserving forward rates over intervals. Calypso's methods preserve the forward rates, thus the existing curve is changed, not merely extended, when each successive single-period instrument is added.

Example of Gap Handling for a Single Money Market Underlying

As an example of gaps, consider a simple case where the curve is built with only one money market instrument, a 1M Libor with quoted rate of 3.00%. As will be seen, this creates *two* points on the curve, one each at the start and the end date of the interest rate period.

The generation goes through the following steps:

1. One notes the rate applies to a deposit that begins on the Spot Date, which is the Curve Date plus two business days. The deposit ends on the Spot Date plus 1M. The generator calculates these dates and regards the 3.00% as a forward rate for the interval (Spot Date, Spot Date + 1M).

2. There is now a gap between the Curve Date and the Spot Date. This is the situation described above under First interval added to the curve. To find a rate for the interval between these dates extrapolation must be used. In this case, the steps are:

a) Convert the simple-interest ACT/360 Libor rate to a continuous ACT/365 rate. The formula used is:

$$1+0.03(\frac{T}{360}) = \exp(R\frac{T}{365})$$

where T is the number of actual days in the 1M deposit. This formula is solved for the continuous rate R.

b) Assign the rate R to the interval between the Curve Date and the Spot Date.

3. Find the discount factor at the Spot Date using the forward rate between the Curve Date and the Spot Date. This is:

$$DF(CurveDate, SpotDate) = \exp(-R\frac{S}{365})$$

where S is the number of calendar days to the Spot Date; S = 2 if there are no holidays or weekends. This point is added to the curve.

4. Find the discount factor on Spot Date + 1M using the discount factor on the Spot Date and the forward rate on the interval (Spot Date, Spot Date + 1M). This happens to be the same rate R in this case, because of the extrapolation that had to be performed. Thus:

$$DF(CurveDate, SpotDate+1M) = DF(CurveDate, SpotDate)DF(SpotDate, SpotDate+1M)$$
$$= \exp(-R\frac{S}{365})\exp(-R\frac{T}{365}).$$

This is the second point added to the curve. This gap method has simply applied the 1M rate over the entire period from Curve Date to the maturity date.

Note that the resulting curve has two points, one at Spot Date and one at Spot Date + 1M, and is not the same as would be produced by having only a single point at Spot Date + 1M and none at Spot. The reason is that differing interpolation schemes would result in different discount factors at the Spot Date, which would result in changes to the interest rate between Spot and Spot + 1M; as a consequence the 1M money market would not be guaranteed to price back to the input quote.

Bootstrapping with Forward Interpolation

Linearity Requirement for Forward Rates

In generating zero curves, the standard Calypso bootstrapping algorithm employs market-quoted swaps by solving for the zero rate on the swap maturity date which produces zero net present value for the swap. In doing this calculation, any discount factors and zero rates required for pricing the swap are found by interpolation. The user can choose to interpolate on either zero rates or on discount factors by specifying the environment parameter INT_CURVE_INTERP_RATE_B.

If the user wishes instead to interpolate on forward rates, another bootstrapping algorithm is available. The main feature of this algorithm is **that forward rates used by the swap are required to fall along a line**. This is not quite the same as forcing the curve to be piecewise linear in forward rates, for two reasons:

- Only the actual forward rates needed by the swap fall on a line, not every forward rate within the swap's lifetime.

- Only swaps and basis swaps implement this requirement; cash and futures are solved for as in the standard zero-rate bootstrap.

To employ this bootstrapping method, select **BootStrapForwards** in the generation algorithm field of the CurveZero application.

The BootStrapForwards Generation Procedure

The BootStrapForwards extends a zero curve with a swap using the following procedure:

1. A forward rate for the last floating-rate cashflow of the swap is guessed as a trial rate.
2. For every floating-rate cashflow in the swap, a forward rate is **linearly interpolated** between the last known forward rate from the zero curve and the trial rate. Linear interpolation is used for forwards regardless of the interpolation method defined in the curve application.
3. From the interpolated forward rates, discount factors are found on the start and end of the forward rate periods. In case of gaps, extrapolation across gaps is done using
4. To find the discount factors on all cashflow payment dates, where they differ from forward start and end dates, the interpolation method defined in the curve application is used. The two most commonly used methods are *linear zero rate interpolation* or *compound daily forward interpolation*.
5. Find the NPV of the swap using these discount factors. Adjust the trial forward rate until this NPV is zero.
6. Finally, all of the discount factors used in pricing the swap are stored in the curve. This differs from the zero rate bootstrapping, which only adds one point for each swap. The BootStrapForwards algorithm thus produces curves with many more points on them than the BootStrap algorithm. As a result, it is possible for a user to change interpolation methods after curve generation while still maintaining the swap NPV at zero.

The Algorithm in Detail

In this section, the generation will be described for a curve constructed from money market instruments to 1Y, then two semi-annual swaps of maturity 2Y and 3Y.

<u>Notation:</u>

$i = 0, 1, \ldots, 5$: label for the 6 cashflows of the swaps to 3Y.

FS_i, FE_i : forward period start and end dates for the ith cashflow

fw_i : forward rate of ith cashflow, thus for the interval (FS_i, FE_i)

PD_i : payment date for the ith cashflow

$D(t)$: discount factor for date t

Note that most of the time, but not always, $FE_i = PD_i$.

1. Start with the 1Y money market curve. Add the 2Y swap.

2. Guess fw_3. This is the last needed forward rate of the 2Y swap.

3. The first reset fw_0 of the swap is known from the 1Y curve, but fw_1 may not be known if PD_1 or FE_1 falls after the last date of curve. In this case use *extrapolation* (zero rate or compound daily forward) from the discount factors (not the forwards) of the curve to find the discount factor on PD_1 and/or FE_1 as needed

   a) 1Y swap is added as an underlying to determine fw1 so that the 1Y swap is priced par.

4. Draw a line between the known rate fw_1 and the guessed rate fw_3. Read off fw_2 from this line. Thus fw_1, fw_2 and fw_3 lie on a line. This satisfies the piecewise linear forward requirement.

5. Using fw2 and fw3, calculate discount factors on the forward rate start and end dates FS_2, FE_2, FS_3, and FE_3. For example,

$$D(FE_2) = \frac{D(FS_2)}{1 + fw_2 * (FE_2 - FS_2)}.$$

In this equation, if $D(FS_2)$ is unknown, interpolate it from existing known discount factors (zero rate interpolation or compound daily forward).

6. Using the discount factors computed in Step 5 along with the yield curve, interpolate the discount factors on the payment dates PD_2 and PD_3. In the infrequent case where $PD_3 > FE_3$, there will be extrapolation rather than interpolation.

7. Price the 2Y swap, and repeat the guess of fw_3 until the NPV of the swap is zero. The yield curve has now been extended to the later of FE_3 and PD_3.

Now add the 3Y swap:

8. Guess fw_5.

9. Draw a line between the previously solved fw_3 and the guessed fw_5. Read off fw_4 from this line. Thus fw_3, fw_4, and fw_5 lie on a line. This satisfies the piecewise linear forward requirement.

10. Using fw_3, fw_4, and fw_5, calculate discount factors on FS_4, FE_4, FS_5, and FE_5, using the same type of equation as in Step 5.

11. Using the discount factors computed in Step 10 along with the yield curve, interpolate the discount factors on the payment dates PD_4 and PD_5 (zero rate or compound daily forward interpolation). Again, in the rare case where $PD_5 > FE_5$, there will be extrapolation rather than interpolation.

12. Price the 3Y swap, and repeat the guess of fw_5 until the NPV of the swap is zero. The yield curve has now been extended to the later of FE_5 and PD_5.

Thus the two requirements -- linearity of forward rates and NPV of the swaps being zero -- are satisfied.

Compound Daily Forward Interpolation

The Compound Daily Forward interpolation method is the preferred interpolation method for use with the Bootstrap Forwards generator. It finds discount factors on a date between two dates with known discount factors by assuming the forward rate between the known dates is constant. See Interpolation Methods for details.

1.2.2 Basis Curve

A **basis curve** in Calypso is a curve holding two components:

- An underlying curve, called the **base curve**
- A set of spreads to the zero rates of the base curve

The basis curve can be used as a discount or forward curve for trade valuation, in the same manner as a zero curve. The spreads referred to here are not market quoted spreads, but rather zero rate spreads that have already been generated or manually entered.

A basis curve is generated by specifying a base curve and a set of underlying market quotes. The set of spreads to the base curve is computed so that the market quotes are reproduced when the resulting basis curve is used to value the underlyings.

A **cross-currency (xccy) basis curve** has an additional dependence on a second underlying curve in another currency from the base curve. The resulting basis curve is assigned the same currency of the base curve.

The CurveBasis application is used to generate both single-currency and cross-currency basis curves.

Representation of zero rates

A basis curve computes its zero rates from the zero spreads it holds and its underlying base curve. The following describes how this combination is done. The generation of the spreads from the underlying market quotes is assumed to already have taken place.

Suppose the zero rates of the underlying base curve are Z_i for dates T_i , and the set of zero spreads s_j on dates t_j. The basis curve defines curve points on *each* of the dates T_i and t_j. The zero rate on any date not falling on these points is found from interpolation of the zero rates on these points, using the interpolation method (e.g., linear) specified for the basis curve.

The curve points of the basis curve are defined as follows.

For spread dates:

1. For each spread date t_j, use the base curve to interpolate its zero rate for that date, using the interpolation method defined on that curve. Call this interpolated rate $Z(t_j)$.

2. The combined zero rate of the basis curve at t_j is then defined by adding this interpolated base zero rate to the zero spread:

$$z(t_j) = Z(t_j) + s_j \,.$$

For base curve dates:

1. For each curve point on the base curve at date T_i not equal to some t_j, interpolate $s(T_i)$ from the spreads s_j. Use the method defined on the curve for spread interpolation; the default is linear interpolation on the s_j.

2. The combined zero rate of the basis curve at T_i is then defined by adding this interpolated spread to the base zero rate:

$$z(T_i) = Z_i + s(T_i)\,, \quad \min(t_j) \; < T_i \; < \; \max(t_j),$$

The spread is assumed constant on dates before the first spread date and after the last spread date.

The base zero rates Z_i can be expressed in terms of a day count and compounding frequency different from those of the spreads, in which case a conversion to those of the spread is performed before the addition is made.

From the basis curve $z(t)$ formed in this way on all points T_i and t_j, discount factors and zero rates can be found in the same manner as any other zero curve.

Generation from basis swaps

To extend a basis curve using basis swaps (floating versus floating swaps), the base curve is employed directly in the computation. One leg of the swap pays from an index that is forecast using the base curve; this is called the *base leg*. The index of the other leg is forecast using the basis curve; this will be called the *basis leg*.

Thus the curves used in valuing a single-currency basis swap are as follows:

Base leg:

- Discount curve: base curve
- Forecast curve: base curve

Basis leg:

- Discount curve: base curve
- Forecast curve: basis curve (to be solved for)

The basis curve is solved for in this manner:

1. Find the NPV of the base leg using the base curve. This is the target NPV.

2. Add a trial zero rate spread to the basis curve at the maturity date of the basis leg. Use this amended basis curve to forecast the base leg forward rate. Use the base curve for discounting to find the NPV.

3. Repeat Step 2, varying the trial spread until the NPV of the basis leg matches the target NPV:

NPV (basis leg) = NPV (base leg)

The spread that causes this to be true becomes the zero spread that extends the basis curve.

Basis Curve Generation: Cross-currency

Cross-currency (xccy) basis curve generation differs from that of single-currency basis curves in that a discount curve is solved for, rather than a forward curve, when using cross-currency swaps. The goal is that, given a base curve for discounting one currency, that curve is adjusted with spreads as governed by the swap spreads with respect to another currency. The following explains the practice:

> In theory, a new floating-for-floating swap should involve exchanging LIBOR in one currency for LIBOR in another currency (with no spreads added). In practice, macroeconomic effects give rise to spreads. Financial institutions often adjust the discount rates they use to allow for this. As an example, suppose that market conditions are such that USD LIBOR is exchanged for Japanese yen (JPY) LIBOR minus 20 basis points in new floating-for-floating swaps of all maturities. In its valuations a U.S. financial institution would discount USD cash

> flows at USD LIBOR and it would discount JPY cash flows at JPY LIBOR minus 20 basis points. It would do this in all swaps that involved both JPY and USD cash flows.
>
> -- J. Hull, 5th Edition, *Options, Futures, and Other Derivatives*, p. 598

The CurveBasis application is used for cross-currency basis curve generation. The difference from the single-currency case is that a foreign curve must be chosen in addition to the base curve. The resulting basis curve will be associated with the original base curve currency, not the foreign currency.

Generation from cross-currency basis swaps

To extend a basis curve using cross-currency basis swaps, the curves used in valuing the swap are as follows:

Base leg:

- Discount curve: basis curve (to be solved for)
- Forecast curve: base curve

Foreign currency leg:

- Discount curve: foreign curve
- Forecast curve: foreign curve

Please note: It is important to define the xccy basis swap curve underlying so that there is an Actual exchange of principal. Otherwise the curve generation is likely to fail.

The basis curve is solved for in this manner:

1. Find the NPV of the foreign leg using the foreign curve. This is the target NPV. This is in the quoting currency of the basis swap's currency pair.

2. Forecast the forward rates on the base leg using the base curve.

3. Add a trial zero rate spread to the basis curve at the maturity date of the base leg. Use this amended basis curve to discount the base leg to find the NPV in the quoting currency.

4. Repeat Step 3, varying the trial spread until the NPV of the basis leg matches the target NPV. The spread that causes this to be true becomes the zero spread that extends the basis curve.

Generation from FX forwards (fx swap rates)

FX forward underlyings can also be used in extending a basis curve. The quotes given for these underlyings in the CurveBasis application are interpreted as fx points (swap points), not all-in forward rates. The generation over mixed instruments proceeds by first dealing with the FX forwards, then the money market and futures underlyings, and then the basis swaps.

Generation with the FX forwards is thus done before the other underlyings and proceeds as follows:

1. Use the same technique as CurveZeroFXDerived application to combine the foreign curve with the FX forward underlyings. The result is new zero curve in the base currency.

2. Split the zero rates of the new curve into those of the underlying base curve and a zero rate spread. Store the result as a new basis curve.

The new basis curve is then extended using the remaining non-FX underlyings.

1.3 Inflation Curves

Inflation curves are used to price inflation swaps and inflation caps/floors.

You can create a simple spread inflation curve from a base inflation curve, or a derived inflation curve from underlying instruments.

1.3.1 InflationBasis Generator

Used to generate an inflation curve from a base inflation curve. The points are the spreads over the base curve.

1.3.2 InflationDefault Generator

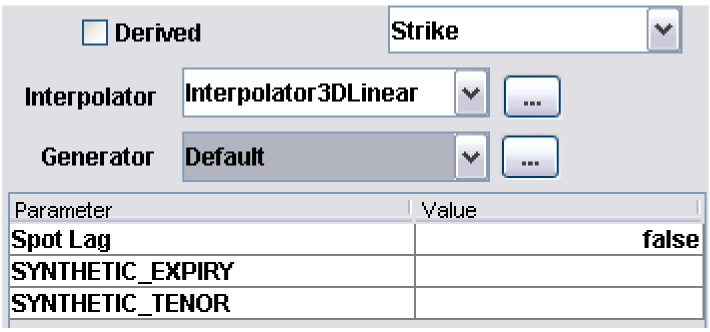
Used to generate an inflation curve from underlying instruments. It only supports bootstrapping from Zero Coupon swaps.

The curve includes parameters for making seasonal adjustments.

1.4 Volatility Surfaces – Simple Generators

1.4.1 Default Generator

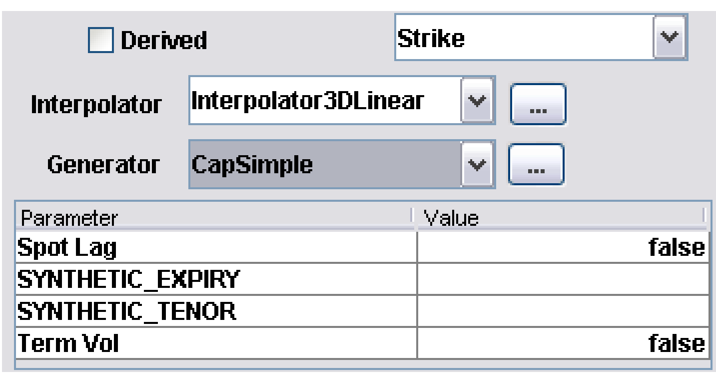
Default Generator — Can be used for caps, swaptions, bond options, etc. So-called simple surface where surface points are defined by the user, not using underlying instruments. If the spot lag parameter is set to true, the generated exercise dates are rolled using the conventions of the definition screen.



Note that SYNTHETIC_EXPIRY and SYNTHETIC_TENOR are not currently used.

1.4.2 CapSimple Generator

CapSimple Generator — Specifically for caps. Same as default generator, except that the user can define if the entered volatilities are term volatilities (i.e. the usual market quotes for caps/floors), Term Vol parameter = true, or regular volatilities (a.k.a. caplet/floorlet “forward” volatilities), Term Vol parameter = false.



Note that SYNTHETIC_EXPIRY and SYNTHETIC_TENOR are not currently used.

1.4.3 CapSABR Generator

Specifically to price caps/floors using the SABR model.

The parameters have the following default values.

Parameter	Value
Default_Beta	1.0
Default_Correlation	0.0
Default_VolofVol	0.01
Tolerance	15
Lambda	0.01
MaxIterations	5000
GetVolUseAlpha	▾ false
GetVolUseATMVol	▾ false
GetVolUseATMVolFromMDI	▾ true
CalibratedWithForecastCurveID	1001
CalibratedWithForecastCurveName	USD LIBOR
CalibratedWithForecastCurveTime	2/13/07 6:17:13.000 PM PST
CalibratedWithForecastCurveDesc	USD/LIBOR3M/USD LIBOR(R)/CLOSE/Feb 7, 2007 9:00:16 AM
CalibratedWithFwdFwdVolID	1401
CalibratedWithFwdFwdVolName	USD LIBOR Volatility
CalibratedWithFwdFwdVolTime	2/13/07 6:19:13.000 PM PST
CalibratedWithFwdFwdVolDesc	USD/RATE/LIBOR/USD LIBOR Volatility/CLOSE/Feb 13, 2007 6:19:13 PM

1.4.4 OFMSimple Generator

The user explicitly defines the volatility and means reversion term structures. This is a so-called simple surface where no generation actually takes place. The user has to specify a list of generation parameters to allow the volatility surface to be defined in terms of the axes (expiry, tenor, strike) as well as the context in which this surface will be used for pricing. The output points of the surface are initialized with “trivial” values (i.e. all zeros) since it is the user’s responsibility to enter the appropriate values for all 3 sets of output point values, i.e. the Black volatilities, the model volatilities and the model mean reversions for each surface vertex point. The look-and-feel as well as the behavior of such a generated volatility surface are EXACTLY the same as those of any other volatility surface in the Calypso system. The only thing (for a software engineer) to remember is that technically speaking the Black volatilities are surface points whereas the model volatilities and mean reversions are surface adjustments, but this is irrelevant for the user.

The OFMSimple generator creates a container for the volatility values which looks similar to the output from the OFMSwaptions generator. However, no values are actually filled in. In this case, we assume the points and calibration are taking place outside of Calypso and they will be manually keyed in by the user.

For this generator, you do not need to specify anything in the Offset panel. You can use the following parameters to define expiration dates, tenors, and strikes.

Parameter	Value
Expiry Start	1Y
Expiry End	5Y
Expiry Step	1Y
Tenor Start	1Y
Tenor End	5Y
Tenor Step	1Y
Strike Start [%]	-1
Strike End [%]	1
Strike Step [%]	1
Distribution	▾ LogNormal
Number of Vertical Nodes	50
Number of Standard Deviations	3

1.4.5 SwaptionSimple Generator

Used for storing volatility points on a surface where the volatilities are constructed outside of Calypso. A basic volatility surface.

1.4.6 CMSBasisAdjSimple Generator

Used to store CMS volatility basis adjustments on the volatility surface.

1.4.7 SABRSimple Generator

Used to create the implied smile along the strike axis. ATM Black volatilities are the input and the model parameters alpha, beta, rho, nu create the implied smile that is applied to the volatility surface.

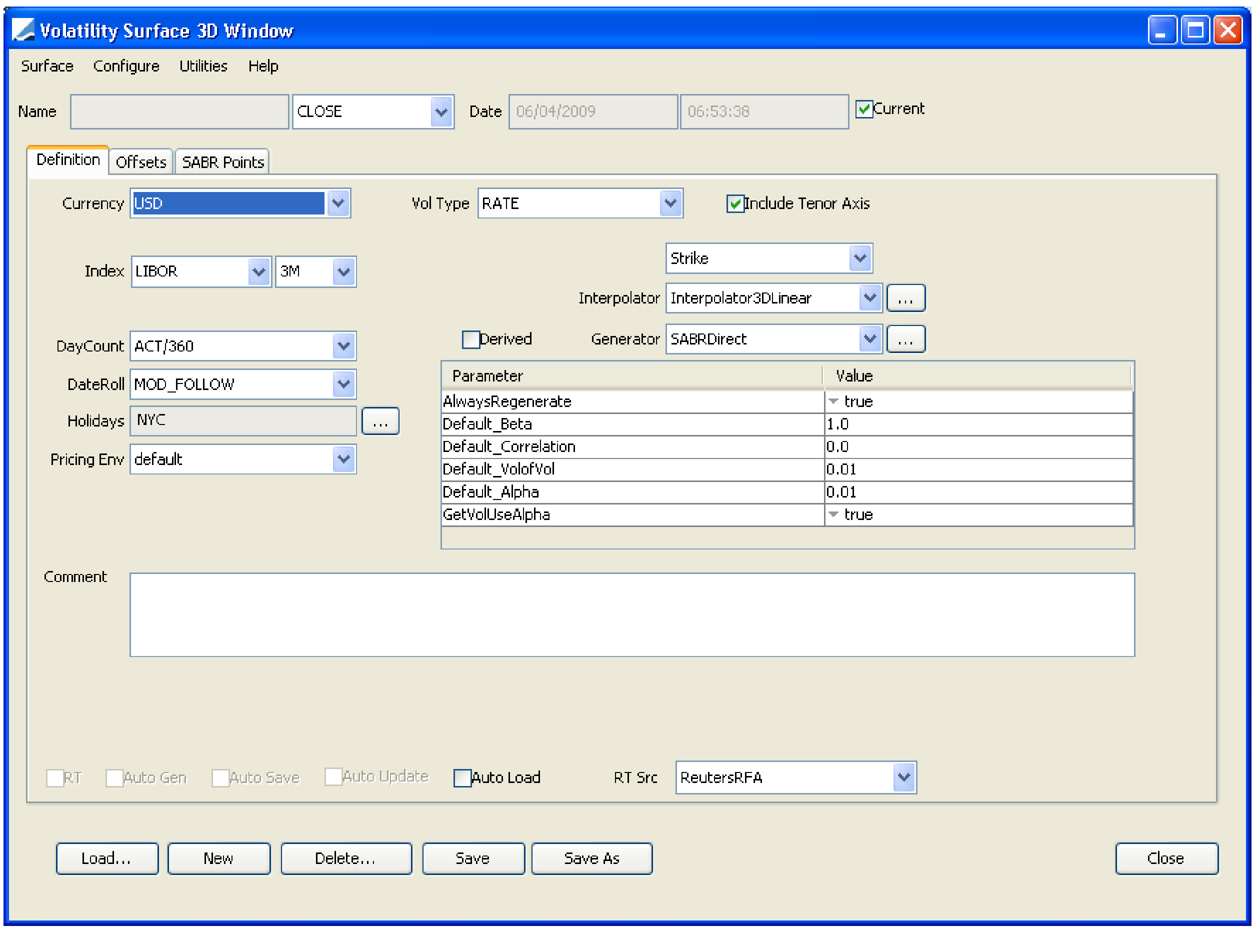
1.4.8 SABRDirect Generator

This is a simple SABR volatility surface. Generally, used in the case when the SABR parameters are calibrated outside of Calypso, and one simply wants to enter the SABR parameters to parametrically define the swaption smile surface. Several steps are required to setup the surface;

Definition

The key points on the definition are;

1. Select Include Tenor Axis
2. Select SABRDirect generator
3. Set the generator parameter GetVolUseAlpha = true, (this is important!)

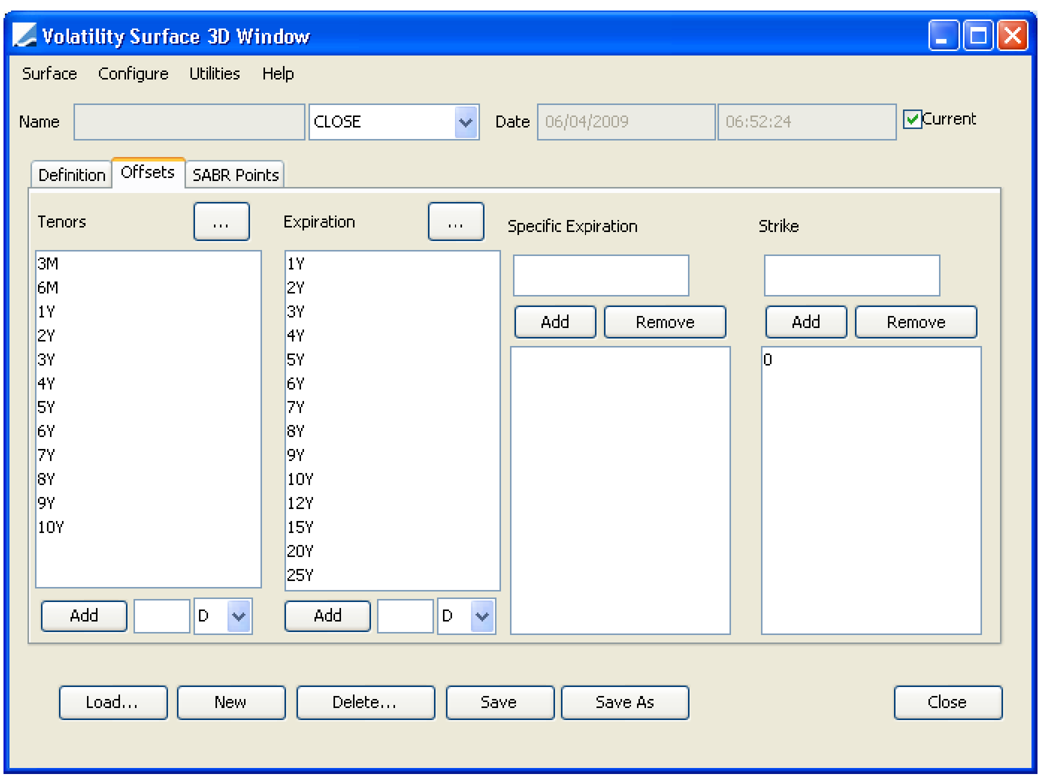


Parameter	Description	Typical Value
AlwaysRegenerate	Superfluous, no longer used	False

Default_Beta	The initial value of beta used when first constructing the beta matrix or when adding a new row or column. The user may edit the surface value to be different from the default value.	1.0
Default_Correlation	The initial value of correlation used when first constructing the correlation matrix or when adding a new row or column. The user may edit the surface value to be different from the default value.	0.0
Default_VolOfVol	The initial value of volatility of volatility used when first constructing the vol of vol matrix or when adding a new row or column. The user may edit the surface value to be different from the default value.	0.01
Default_Alpha	The initial value of alpha of volatility used when first constructing the alpha matrix or when adding a new row or column. The user may edit the surface value to be different from the default value.	0.01
GetVolUseAlpha	This controls the calculation of the implied black volatility return from this volatility surfaces' getVolatility function used by pricers. In the case of GetVolUseAlpha=true, alpha can be read directly from the vol surface point adjustment layer ALPHA, otherwise ATM vol is read from the ATMVOL layer and alpha recomputed (recalibrated) on the fly. This is an extremely important parameter. Generally most users will set this parameter to True.	True

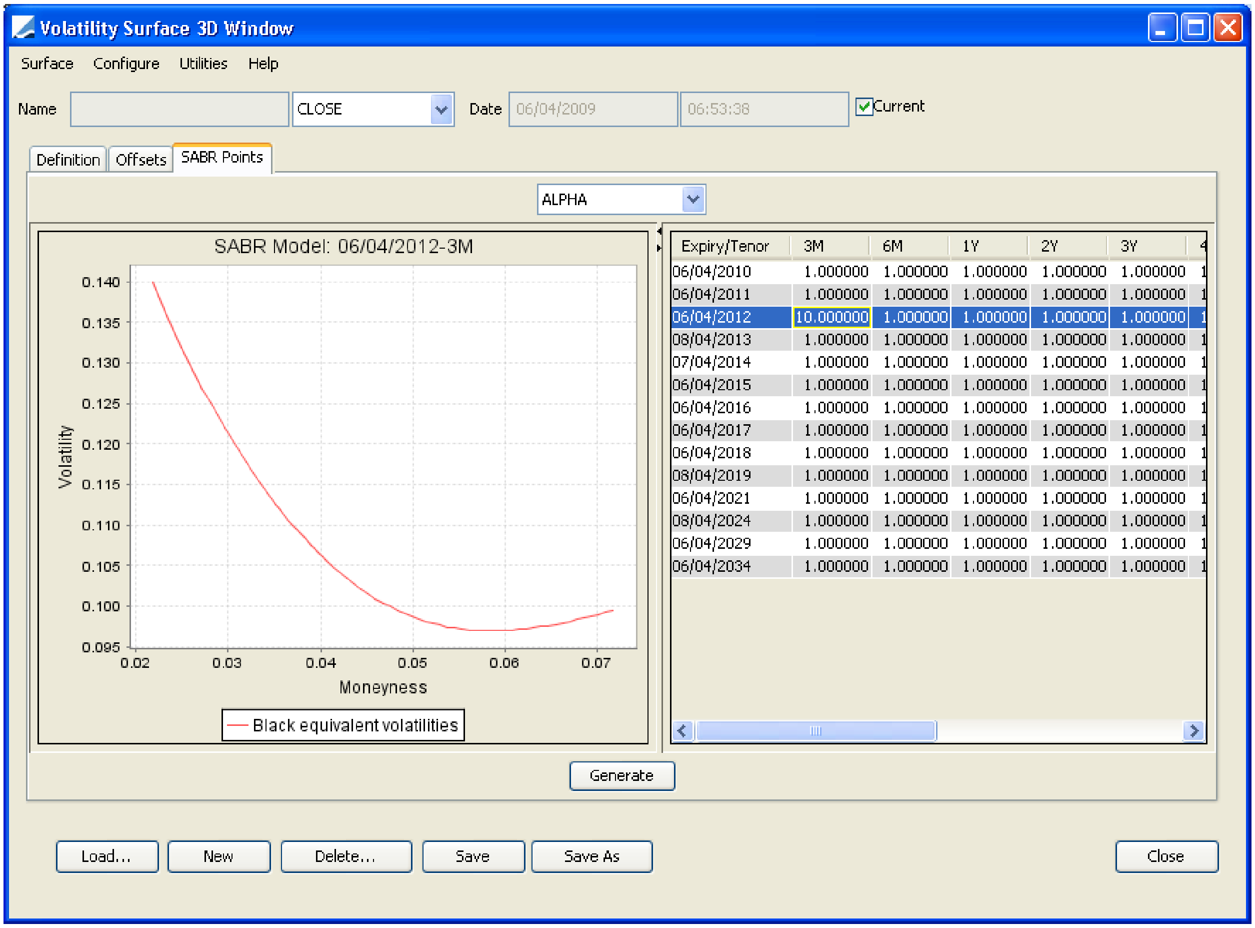
Offsets

Generally, one configures the surface with user defined expiries and tenors, for technical reasons one is required to add one strike to the strike axis. An example is;



SABR Points

Then on the SABR Points tab, one generates to initialize the set of SABR parameters, which can then be edited by the user. When one selects a point on the surface, the SABR implied smile is visualized in a graph for that selected expiry and tenor.



Having configured the SABR surface, it may be used by any pricer that requires swaption volatitlies not just PricerSwaptionSABR, for example PricerSwaption and PricerSwapHagan.

It is worth noting that no attempt has been made within the generator to cutoff the SABR parametric definition in the wings of the surface.

1.4.9 SABRDirectBpVols Generator

Similar to SABRDirect, except that you input ATM BP volatilities instead of ATM Black volatilities.

1.4.10 LGMMeanRev Generator

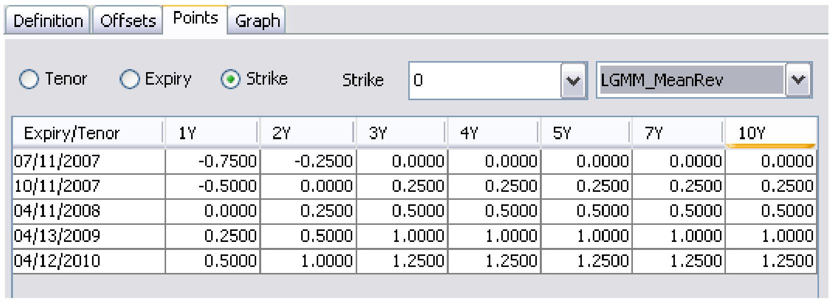
The Linear Gauss Markov Model is really the Hull-White one factor model reset in the Heath-Jarrow-Morton framework for **Bermudan Swaptions**. This alternative characterization greatly helps with calibration and trade valuation.

Calibration

The calibration is stored in a layer of a volatility surface on the Points panel. The generator takes a list of ATM swaption volatilities as inputs and creates an empty container for the mean reversion values. In the case of this model, these values change very infrequently and so it is reasonable that they would remain their original values for the better part of a year. For this reason we have separated the volatility surface that the model requires from the mean reversion parameters that it also requires. This enables the volatility surface to change and be regenerated daily whereas the mean reversion values can remain constant.

Calibration Matrix

In the Points panel of a volatility surface generated with the (simple) LGMMMeanRev generator, there is a layer created titled LGMM_MeanRev. This is where the user is required to key in the values, possibly calculated on a spreadsheet, and then save the surface.



Expiry/Tenor	1Y	2Y	3Y	4Y	5Y	7Y	10Y
07/11/2007	-0.7500	-0.2500	0.0000	0.0000	0.0000	0.0000	0.0000
10/11/2007	-0.5000	0.0000	0.2500	0.2500	0.2500	0.2500	0.2500
04/11/2008	0.0000	0.2500	0.5000	0.5000	0.5000	0.5000	0.5000
04/13/2009	0.2500	0.5000	1.0000	1.0000	1.0000	1.0000	1.0000
04/12/2010	0.5000	1.0000	1.2500	1.2500	1.2500	1.2500	1.2500

1.5 Volatility Surface – Derived Generators

1.5.1 Cap Generator

Cap Generator — Specifically for caps and floors. Generates a (forward) volatility term structure for given cap/floor term volatilities or given caplet/floorlet volatilities (a.k.a. forward volatilities).

The strike can be defined as relative (ATM or off-ATM, in % or bp) or absolute. Any strike dependent adjustments (OTM or ITM volatility adjustments) are computed into the generated volatilities.

On the client side, if a volatility is polled from a surface generated by this generator with relative strikes defined, the forward rate for the particular caplet/floorlet for the given expiry/tenor vertex point is taken as the ATM strike. Subsequently, the strike offset (difference between the actual (absolute) strike of the caplet/floorlet and the ATM strike for the given expiry/tenor vertex point) is passed to the surface to retrieve the required volatility. For absolute volatilities, there are no adjustments necessary.

If caplets/floorlets are the volatility surface underlying instruments, then generation means simply putting the input volatilities on the respective vertex point (expiry/tenor/strike) of the volatility surface.

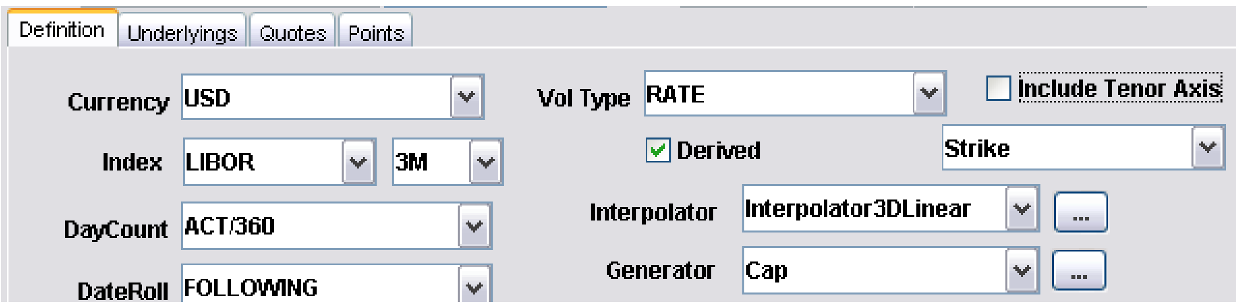
If caps/floors are volatility surface underlying instruments, then the following steps are followed to bootstrap volatilities given term volatilities:

1. A preliminary volatility surface is built for the expiry/tenor/strike points using the surface underlying instruments and their term quotes.
2. Each cap/floor is decomposed into its equivalent caplets/floorlets. Synthetic caps/floors matching the end date of each caplet/floorlet are created and priced using interpolated term volatilities (i.e. same volatility is used for all caplets/floorlets of the given synthetic cap/floor). This produces a set of target present values for a series of caps/floors. For example: we are given a 1 year 3M Libor cap and a 2 year 3M Libor cap. The described procedure produces a 3M, 6M, 9M, 12M, 15M, 18M, 21M and 2Y cap where the associated term volatilities are interpolated on the quotes provided by the original surface underlying instruments.
3. Now we have to solve for the volatility of the last caplet of each synthetic cap such that we re-price it exactly. In the example above, the (synthetic) cap 3M volatility is the same as the volatility of the first 3M caplet. Taking the 6M (synthetic) cap target present value, we find the implied volatility of the 3M into 3M caplet such that its NPV plus the sum of the the3M (spot) caplet equals the target present value. Taking the 9M (synthetic) cap we compute the implied volatility for the 6M into 9M caplet by summing its own NPV with the NPVs of all the previous caplets (in this case the3M (spot) caplet and the 3M into 3M caplet). And we continue this procedure until the last caplet that spans from 21M to 2Y.

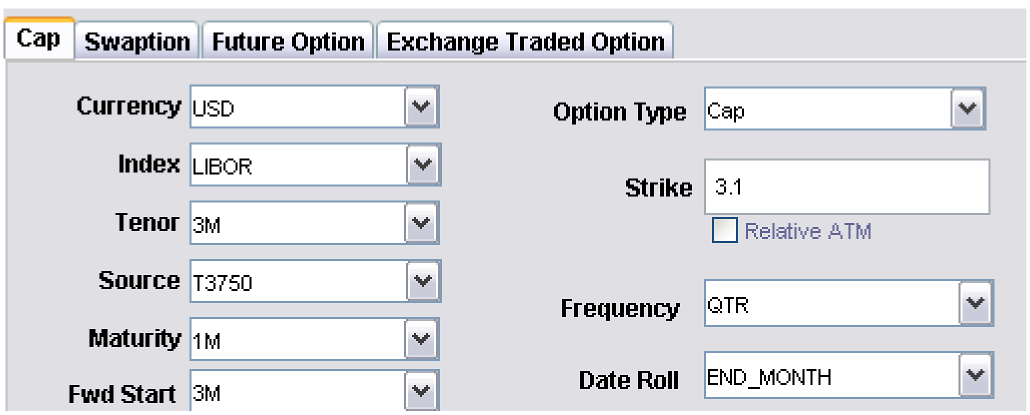
4. It should be noted that we transform from cap/floor maturity dates on the expiry axis to true expiry dates by taking the expiry dates of the generated caplets/floorlets.
5. It has to be emphasized that at-the-money means that every caplet/floorlet of a cap/floor is struck at its own forward rate, rather than on the “average” par swap rate of the equivalent swap (a.k.a. zero cost collar method). The reason is that when using the zero cost collar (ZCC) method, one cannot re-price for example a 2Y/3Y ZCC-ATM forward cap (2Y ZCC-ATM cap starting in 3Y) on a volatility surface built using both a 2Y ZCC-ATM cap as well as a 5Y ZCC-ATM cap as volatility surface underlying instruments. Depending on the shape of the interest rate curve, the ZCC-ATM rates for the equivalent 2Y swap, 2Y/3Y forward swap and 5Y swap are usually not the same. Therefore different strike offsets (i.e. actual absolute caplet strike – ZCC-ATM strike) would apply to the same caplet (e.g. 24M-27M caplet) and thus for the same caplet different volatilities would be polled from the volatility surface. Using the reset rate for each caplet as its ATM strike solves this problem.

The Cap generator is only available for a derived curve. The following combinations can be specified.

Strike Definition

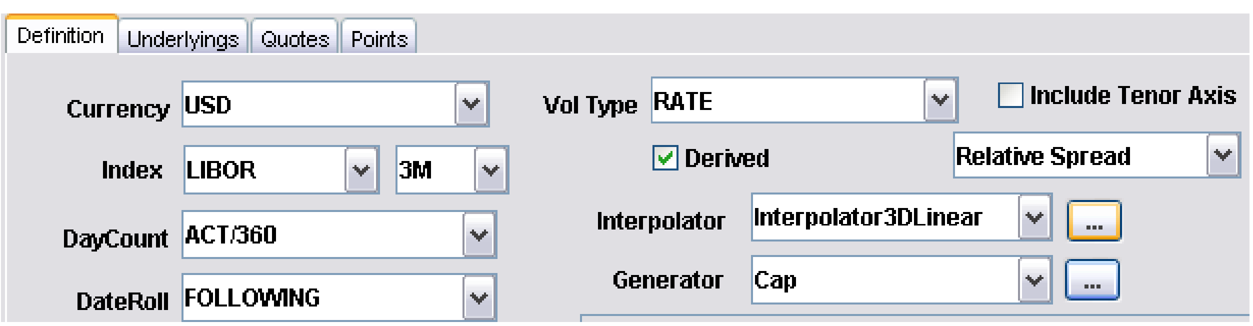


The underlying instruments must be specified using an absolute strike as shown below.

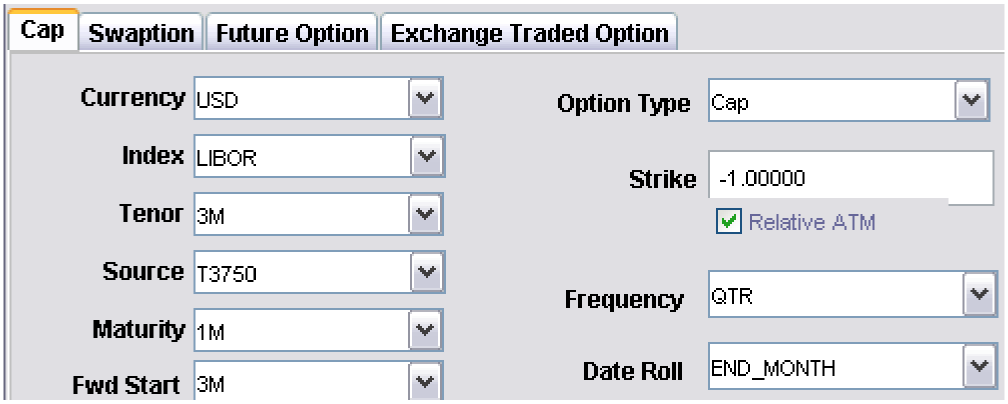


Then, you can enter the quotes and generate the points using the Quotes and Points panels.

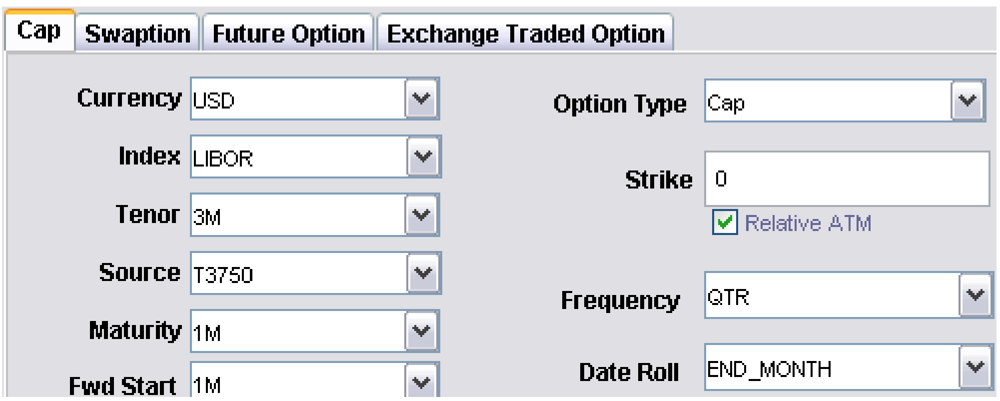
Relative Spread Definition



The underlying instruments must be specified using a relative strike as shown below. In this example the relative strike is -1%.

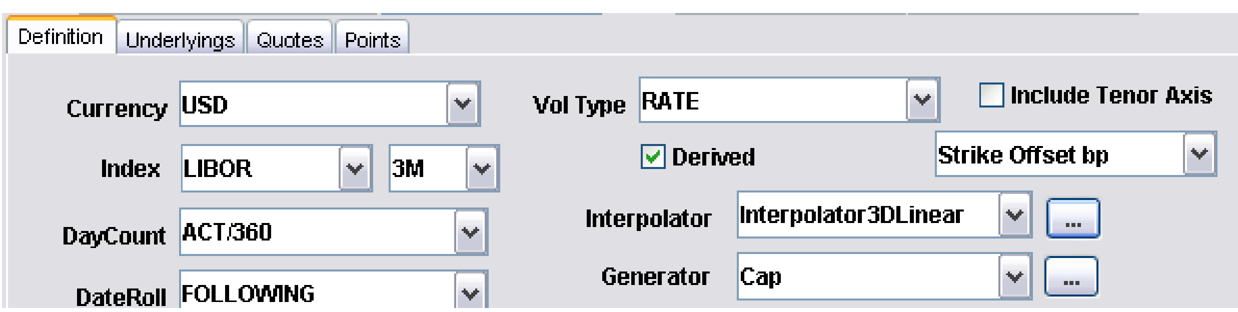


ⓘ Note that when using a relative strike, one of the underlying instruments must be defined with a strike of 0 as shown below.

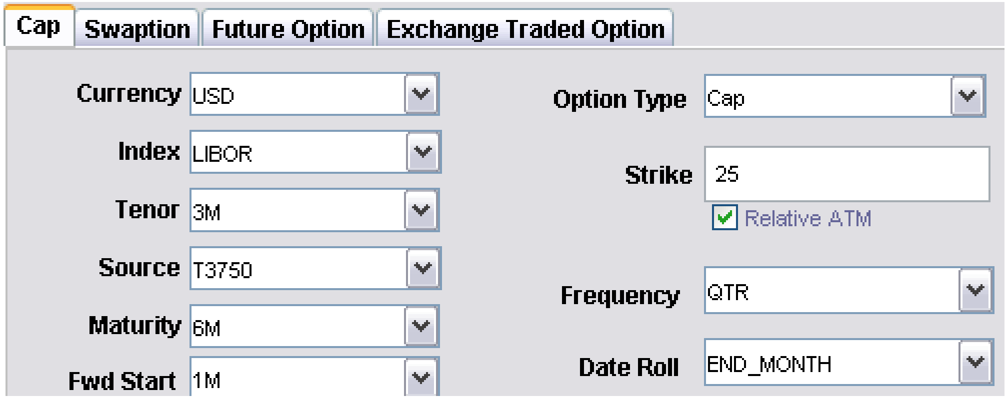


Then, you can enter the quotes and generate the points using the Quotes and Points panels.

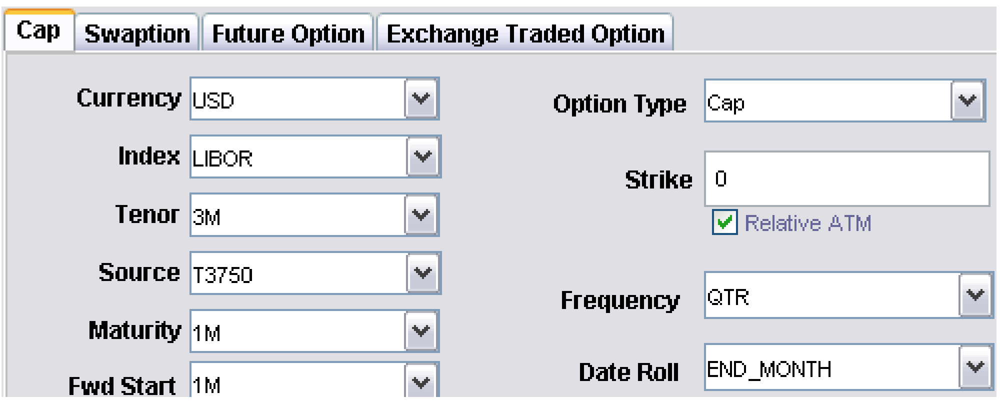
Strike Offset bp Definition



The underlying instruments must be specified using a relative strike as shown below. In this example the relative strike is +25bp.



ⓘ Note that when using a relative strike, one of the underlying instruments must be defined with a strike of 0 as shown below.



Then, you can enter the quotes and generate the points using the Quotes and Points panels.

1.5.2 CapTerm Generator

Used for storing cap volatilities in a surface. Similar to the CapSimple generator except that the points are linked to specific underlying volatilities. The surface points have the same maturity, strike, rate index tenor, and volatility as the underlying cap.

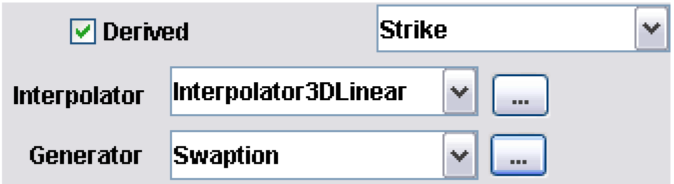
1.5.3 Swaption Generator

Swaption Generator — Specifically for swaptions; generates a swaption volatility surface. Strike can be defined as relative (ATM or off-ATM, in % or bp) or absolute. Any strike dependent adjustments (OTM or ITM volatility adjustments) are part of the generated volatilities.

Generation in context of this generator means simply putting the input swaption volatility on the respective vertex point (expiry/tenor/strike) of the volatility surface.

On the client side, if a volatility is polled from a surface generated by this generator with relative strikes defined, the forward par swap rate is computed for the given expiry/tenor vertex point (ATM strike). Subsequently, the strike offset (difference between the actual (absolute) strike and the ATM strike for the given expiry/tenor vertex point) is passed to the surface to retrieve the required volatility. For absolute volatilities, there are no adjustments necessary.

All other quoted strikes are converted to relative strikes by computing a strike offset which is the difference between the actual (absolute) strike and the ATM strike for the given expiry/tenor vertex point.



1.5.4 OFMSwaptions Generator

Generates an OFM volatility surface from swaptions, and calibrates the mean reversion and volatility term structure to those underlying instruments.

A set of swaption underlyings with their market observed volatilities are used to bootstrap a volatility term structure to reprice those market instruments. Currently the mean reversion is a generator parameter (assumed to be constant) for the whole surface but which of course has a determining effect on the calibrated model volatilities.

First, the user has to specify again various generation parameters that define the surface structure as well as the purpose of the generated surface, i.e. if the surface is to be used using a lognormal or normal process. It is also adviced (but not absolutely necessary) to specify some initial starting values for the multi-dimensional BFGS solver. Last but not least, an interest rate curve has to be specified (currently through its curve ID) which of course should be in line with the interest rate curve later being used for pricing those trades that use this calibrated model volatility surface.

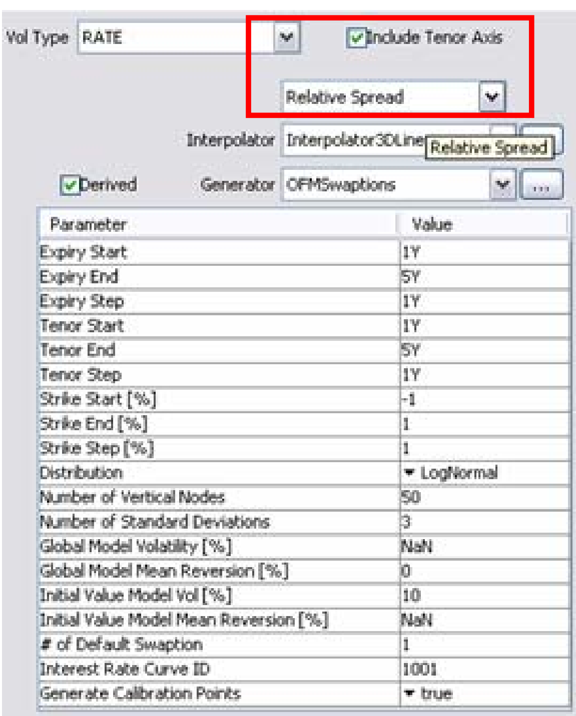
Second, the user has to specify one or more surface underlyings (currently swaptions) on tab 2 and the corresponding market volatilities on tab 3.

With this information, the generator computes the npv's of all underlying instruments. Then via an iterative multi-dimensional solving algorithm (BFGS) the model volatility term structure is computed such that every underlying instrument is repriced on the lattice matching the "Black" npv.

Everything said above for OFMSimple regarding the axes (expiry, tenor, strike) and the generated sets of Black volatilities, model volatilities and mean reversion values also applies to this generator. But this time the output points are (of course) calculated and ready to be used by the pricer.

In general, the look-and-feel as well as the behavior of such a generated volatility surface is again EXACTLY the same as those of any other volatility surface in the Calypso system. The only thing (for a software engineer) to remember is that technically speaking the Black volatilities are surface points whereas the model volatilities and mean reversions are surface adjustments, but this is irrelevant for the user.

Following is an example of the generator parameters.



1.5.5 CMSBasisAdjSimple Generator

Used to store CMS volatility basis adjustments on the volatility surface.

1.5.6 SwaptionBpVols Generator

The generator SwaptionsBpVols uses underlying swaptions to produce a volatility surface. The main advantage of this generator is that it can take a mixture of points, some quoted in Black volatility (or **Yield** Quote Type) and others quoted in basis point volatility (or **BpVol** Quote Type) and then the generator will perform all the necessary conversions to get a surface layer all in Black Volatilities and another surface layer all in basis point volatilities. Now the pricers are free to use the black layer if they require Black volatilities, or use the basis point layer if they require basis point volatilities and both layers are on the same surface.

There are several conversion methods available for converting Black volatilities to basis point volatilities and vice versa. The conversion method is selected as a parameter of the generator.

Conversion Methods	Description
EXACT	$\sigma = f^{-1}(\upsilon)$
HAGAN_APPROX	$\upsilon = \frac{2\sigma}{(F+K)}\left(1+\frac{1}{3}\left(\frac{F-K}{F+K}\right)^2+\frac{1}{6}\left(\frac{\sigma^2 T}{(F+K)^2}\right)+\cdots\right)$
STREET_PROXY1	$\sigma = \sqrt{FK}\upsilon$
STREET_PROXY2	$\sigma = \upsilon\sqrt{\frac{1}{2}(F^2+K^2)}$
STREET_PROXY3	$\sigma = \upsilon F(1-\frac{1}{24}\upsilon^2 T)$
STREET_PROXY4	$\sigma = F\upsilon$
STREET_PROXY5	$\sigma = K\upsilon$

where, σ denotes the bpvol and ν denotes the black volatility.

1.5.7 SwaptionSABR Generator

Used to create the implied smile on a set of Swaption underlyings by setting the model parameters alpha, beta, rho, and nu.

1.5.8 SwaptionSABRDirect

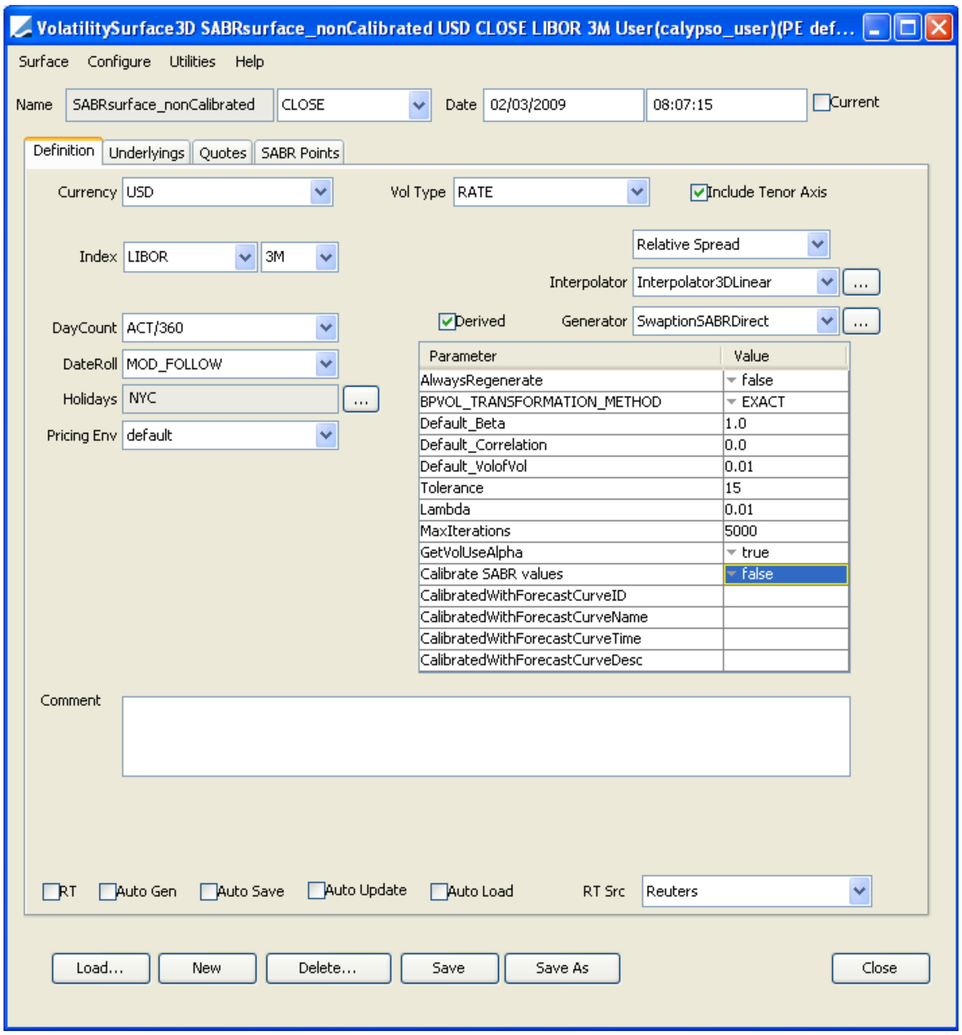
The SABR surface has a dual purpose. One is that it can form a standard black volatility surface which any pricer can use without knowing in advance how the volatilities are computed. The pricers simply ask the volatility surface for the volatility and it delivers. The second purpose of the SABR surface is to feed in model parameters to the SABR model. These are the other layers of points on the volatility surface. Several steps are required to setup the surface using the SwaptionSABRDirect generator;

Definition

The key points on the definition are;

1. Select Derived surface
2. Select Include Tenor Axis
3. Select SwaptionSABRDirect generator
4. Set the generator parameter GetVolUseAlpha = true, (this is important!)

There are two alternative approaches for the user to take if using the SwaptionSABRDirect generator, this is controlled by the Calibrate SABR values flag.



If the Calibrate SABR values flag is set to true the user is able generate a derived SABR volatility surface from ATM swaptions and non-ATM swaptions.

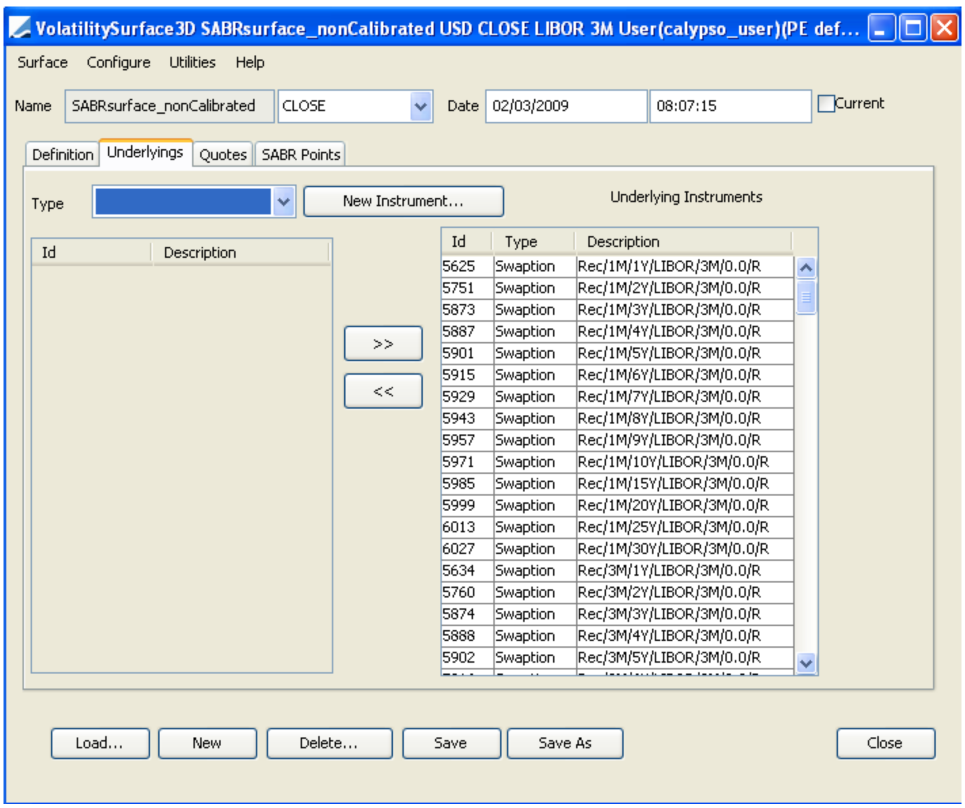
If the Calibrate SABR values flag is set to false, the generator calculates a derived SABR surface from a combination of ATM swaptions and user defined SABR parameters; beta, rho and nu.

Underlyings

Details of the underlying swaptions are entered into the underlyings tab and the quotes updated as required. Careful consideration on the choice of swaptions is needed to ensure that data is available for each point in the expiry vs tenor matrix.

If the Calibrate SABR values flag is set to true, ATM and non-ATM swaptions should be entered. When selecting the underlyings to be used in generating the surface the user must select two non ATM swaptions for each ATM swaption.

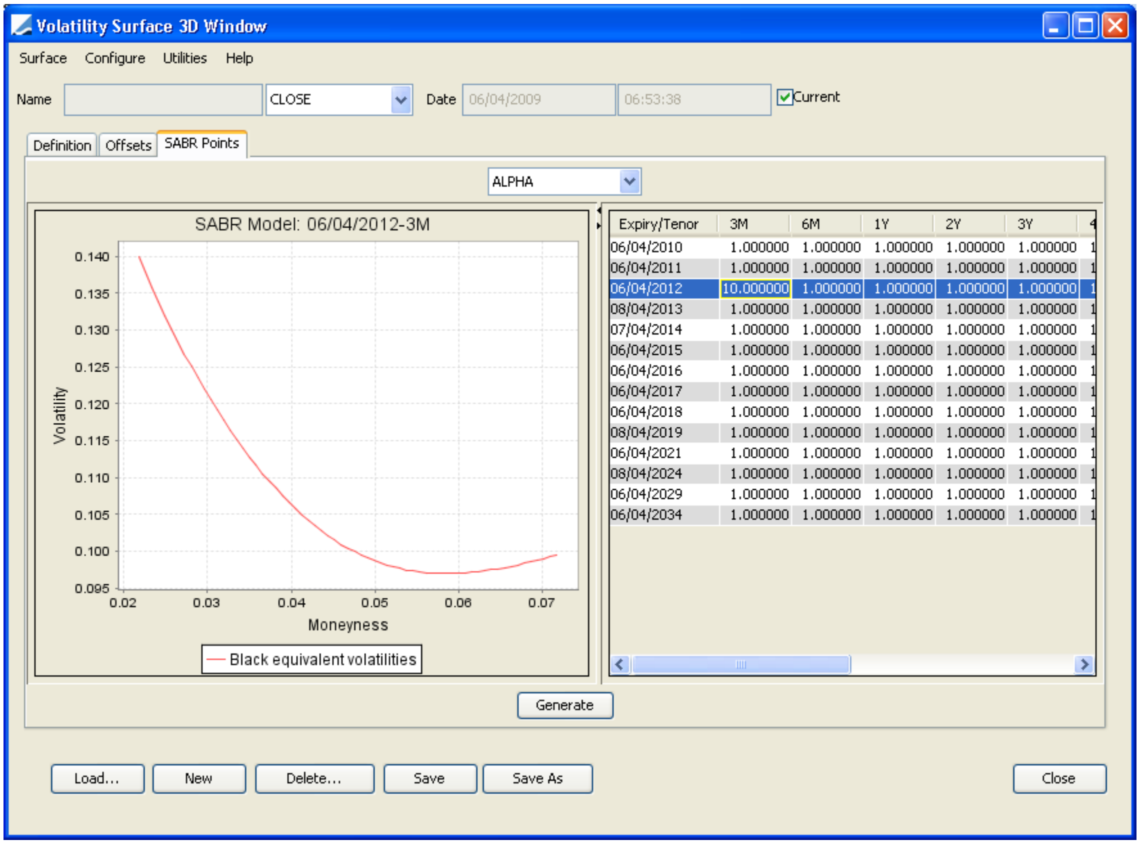
If the Calibrate SABR values flag is set to false, ATM swaptions only should be entered, see below for example configuraton. When the Calibrate SABR values flag is set to false, the user is also required to enter the SABR parameters, beta, rho and nu which parametrically define the swaption smile surface. These can be calibrated outside of Calypso. Typically the ATM swaption data is updated more frequently than the SABR parameters.



On the quotes tab one can enter two vol types, Yield meaning Black vol and BpVol meaning BpVol (normal vol). The generator then uses this information accordingly to create the surface. IF BPvol is selected then the transformation method is chosen from the BPVOL_TRANSFORMATION_METHOD setting on the definition tab (see previous sections for further details).

SABR Points

On the SABR Points tab, one generates to initialize the set of SABR parameters. If the Calibrate SABR values flag is then set to false the user can then edit the SABR parameters beta, rho and nu then re-generate. When one selects a point on the surface, the SABR implied smile is visualized in a graph for that selected expiry and tenor.



1.5.9 CAPSABRDirect

Definition

The key points on the definition are;

1. Select Derived surface
2. Select Include Tenor Axis
3. Select CAPSABRDirect generator
4. Set the generator parameter GetVolUseAlpha = true, (this is important!)
5. Set Default Beta

Underlyings

The CAPSABRDirect generator creates a SABR surface, including alpha, rho and nu, by extracting data from the entered Caps data and grafting onto the swaption vols. The underlyings required for this generator are Caps and ATM swaptions. The user is also required to define beta on the definitions tab (Default_Beta).

1.5.10 FutureOption Generator

Generates and stores volatilities where the underlyings are options on futures.

This only applies if the quote type of the future is rate. If the quote type of the future is price, use instead the MMFUTUTE volatility type and the FutureOption generator.

1.6 Covariance Matrices

In contrast to the One Factor Models which assume all the interest rate movements a function of the (unobservable) short rate, the Multi Factor Model assumes the future of the interest rate movements depend on several observable forward rates which are allowed to move with different random behavior. This lends the model to be calibrated more accurately to the market and is why this model is also known by the name Libor Market Model.

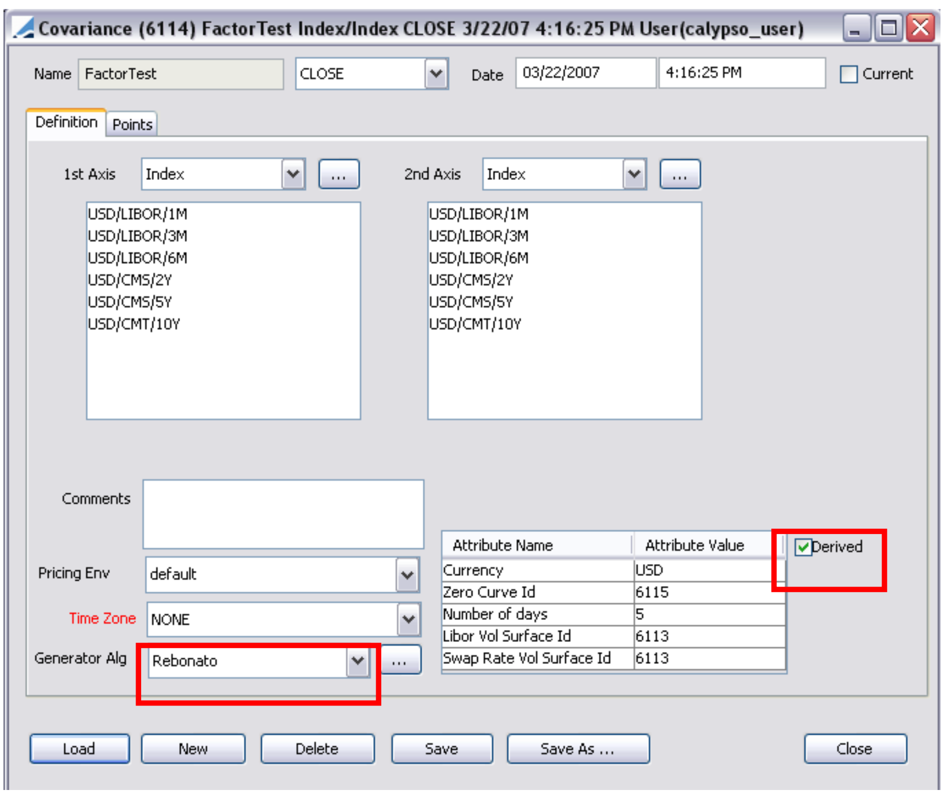
Calypso's implementation of the model uses Monte Carlo simulation to repeatedly walk down possible paths and then average out the total collection of values to determine an expected value of a given trade.

The products supported by the MultiFactorModel are vanilla Swaps with floating rate LIBOR-like, CMS, and CMT; Swaptions with the same floating rate types as the swaps; and CapFloors with the same floating rate type as the swaps.

The model is calibrated to specific tenors on the interest rate curve chosen by the user. These points are allowed to have different volatilities which are specified by a volatility surface. Besides identifying the tenors on the curve which will correspond to the forward rates used in the model, we must also determine the correlations between these. This is achieved by taking a daily history of a given interest rate curve as a sample and computing the correlation from this set.

1.6.1 Rebonato Generator

This calibration is captured as a CovarianceMatrix in Calypso. Only derived covariance matrices are supported. Once the Derived checkbox is ticked, one can select either the MFMDefault generator or Rebonato generator. The Rebonato generator is the suggested generator to use.



Section 2. IRD Trades

Types of Trades	Calypso Product	Pricer
Cancelable Cross Currency Swap	CancellableXCCySwap	PricerCancellableXCcySwap
Cancelable Swap - Single and multiple cancellation dates.	CancellableSwap	PricerCancellableSwap PricerCancSwapOneFactorModel
Cancelable Swap – Vanilla Swap using the Cancelable feature. Single and multiple cancellation dates.	Swap	PricerSwap PricerSwapOneFactorModel
Credit Contingent Swap – Vanilla Swap using the CreditContingent feature. Single name or Basket contingency.	Swap	PricerSwap PricerSwapCreditContBasket PricerSwapCreditContingent
Cross Currency Swap	XCCySwap	PricerXCCySwap
Exotic Swap – Vanilla Swap with exotic structure.	Swap	PricerExoticSwap
Exotic Swap Leg – Vanilla Swap Leg with exotic structure	SwapLeg	PricerExoticSwapLeg
Vanilla CapFloor Basis CapFloor Inflation CapFloor	CapFloor	PricerCapFloor PricerCapFloorMultiFactorModel
Digital CapFloor - Cash or nothing. Single exercise date which decides if holder gets predefined payoff (cash) or not (European style).	CapFloor	PricerCapFloor PricerCapFloorHagan PricerCapFloorMultiFactorModel
Capped, Floored, Collared Swap - Each leg can be individually capped, floored or collared.	CappedSwap	PricerCappedSwap
Exotic Cap Floor Barrier CapFloor Single monitoring of the barrier (European style). Upon exercise, the holder gets the payoff of a vanilla European option.	ExoticCapFloor	PricerExoticCapFloor
Extendible Swap - Single and multiple extension dates.	ExtendibleSwap	PricerExtendibleSwap
FRA	FRA	PricerFRA
Spread Lock - Allows user to enter a swap at a predefined date at a fixed spread over an agreed upon reference asset (bond).	SpreadLock	PricerSpreadLock
Vanilla Swap Basis Swap Yield Curve Spread Swap Inflation Swap	Swap	PricerSwap PricerSwapMultiFactorModel

Types of Trades	Calypso Product	Pricer
Vanilla Swaption Barrier Swaption (Trigger Swaption)	Swaption	PricerSwaption PricerSwaptionBpVol PricerSwaptionMultiFactorModel PricerSwaptionOneFactorModel PricerSwaptionSABR PricerSwaptionCEV
Bermudan Swaption	Swaption	PricerSwaption PricerSwaptionLGMM PricerSwaptionMultiFactorModel PricerSwaptionOneFactorModel
CMS/InAdvance/InArrears Swap	Swap	PricerSwap PricerSwapHagan
CMS/InAdvance/InArrears Leg	SingleSwapLeg	PricerSwap PricerSingleSwapLegHagan
CMS/InAdvance/InArrears/Digital Cap - Digitals valued by call spread.	CapFloor	PricerCapFloorHagan
Spread Cap Floor	SpreadCapFloor	PricerSpreadCapFloorGBM2F – Computes the index forwards using the classic convexity correction and timing adjustments described in Hull's textbook. This pricer is consistent with PricerSwap. PricerSpreadCapFloorGBM2FHagan – Computed the index forwards using the methodology of Hagan (2003). This pricer is consistent with PricerSwapHagan.

2.1 SpreadLock Swap

A spread-lock swap is an agreement that fixes the spread between the forward price of an interest swap rate and its underlying government bond yield (a non-standard benchmark rate in general).

It is priced as a normal swap where the fixed rate is set via the reference bond yield calculation.

2.2 FRA

A Forward Rate Agreement (or FRA) is very similar to a futures contract. It is an agreement between two parties regarding the value or level of a financial instrument at a future date. Unlike futures, FRAs are not traded on an exchange. FRAs are infinitely more flexible, as they can be structured to mature on any date. In general FRAs are traded on the future level of 3 or 6 month Libor.

The FRA does not involve any transfer of principal. It is settled at maturity in cash, representing the profit or loss resulting from the difference in the agreed rate (FRA rate) and the settlement rate at maturity.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

$$\varepsilon = \begin{cases} +1 & for \;\; buying \;\; a \;\; FRA \\ -1 & for \;\; selling \;\; a \;\; FRA \end{cases}$$

- N is the notional

- T_1 is the start date of the FRA
- T_2 is the maturity of the FRA
- $\tilde{T}$ is the fixing date of the FRA
- $\overline{T}$ is the settlement date of the FRA
- yf is the year fraction between T_1 and T_2
- K is the fixed rate (agreed) for the FRA
- $\overline{R}$ is the realized forward rate over the period (or fixing rate)

The FRA amount is first calculated as a cash-flow at the end of the period: $\varepsilon \cdot N \cdot yf \cdot (K - \overline{R})$

Then, it is discounted back to the fixing date, using different types of discounting (NONE, FWD_DISC and DUAL_DISC). FWD_DISC and DUAL_DISC only apply when the FRA is not settled "In Arrear".

On FWD_DISC, Calypso discounts the payment/receipt amount from the end date to the start date using the fixing rate. On DUAL_DISC, Calypso discounts the payment/receipt amount from the end date to the start date using both the fixing rate and the fixed rate. On NONE, no discount is performed.

The amount is finally paid at the payment date (settlement date), so it is multiplied by the discount factor at such date to provide its value at valuation date.

2.3 Vanilla Swap

2.3.1 Standard

A standard (or vanilla) swap could be one of the following types: fixed-floating, fixed-fixed, floating-fixed, floating-floating.

Its characteristics are the following: no amortization structure, the floating leg frequency matches the index frequency.

We will focus on the fixed-floating type of standard swap, since we have both types of swap legs to valuate.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

Each cash-flow i starts at date T_i, ends at date T_{i+1}, has a reset date $\tilde{T}_i$ and a payment date $\overline{T}_i$

- N_i is the notional
- yf_i is the year fraction between T_i and T_{i+1}
- df_t is the discount factor between valuation date and t.
- F_t is the forward index rate projected for reset date t.
- K is the fixed rate.

For the fixed leg, the NPV is the following $NPV_{fix} = \sum_{i=0}^{i=N-1} N_i \cdot yf_i \cdot df_{\overline{T}_i} \cdot K$

For the floating leg, the NPV is the following $NPV_{flt} = \sum_{i=0}^{i=N-1} N_i \cdot yf_i \cdot df_{\overline{T}_i} \cdot F_{\tilde{T}_i}$

For each reset date $\tilde{T}_i$ the rate index has an associated deposit period over which a loan at the reset rate can be made. This period is defined by start and end "deposit period dates," which are also referred to as "forward start" and "forward end" dates. These dates need not be the same as the interest accrual period of the associated swap cashflow. Define:

- s_i the forward start date for the ith reset date
- e_i the forward end date for the ith reset date

If the discount curve that produces the df_t factors is the appropriate curve for projecting the rate index, then an ACT/360 forward rate is found as follows:

$$F_{\tilde{T}_i} = (\frac{df_{s_i}}{df_{e_i}} - 1) \cdot \frac{360}{e_i - s_i}$$

In certain cases, this leads to a commonly-used expression for the NPV of the floating leg:

$$NPV_{flt} = \sum_{i=0}^{i=N-1} N_i \cdot (df_{T_i} - df_{T_{i+1}})$$

This expression is valid if the forward index is the underlying index of the discount curve and if the forward start and end dates of the index deposit period correspond exactly with the start and end dates of the swap's interest accrual period. In practice these are frequently not aligned, due to the different market conventions for deposit periods (which can overlap each other) and consecutive swap leg cashflows (whose interest periods are not allowed to overlap). Calypso therefore does not employ this expression for the NPV.

2.3.2 Amortizing

Swaps where the notional principal is an increasing function of time are known as "*step-up swaps*". Swaps where the notional principal is a decreasing function of time are known as "*amortizing swaps*".

An amortizing product is any financial instrument with a declining (general case) notional principal or with repayments of principal on a predetermined or contingent schedule prior to last maturity.

Different types of amortization structures are implemented, all schedules take place in between a start date and end date that can be different than the start date and maturity dates of the product itself:

- Bullet — default. No amortization is done, each cash-flow notional is the leg's principal notional.
- Annuity — depending on a user-defined fixed rate.
  - If no fixed rate is given (or 0.0), it becomes an equal structure of amortization.
  - If a fixed rate is given, then the amortization annuity is calculated from it. This annuity is then the amount that multiplied by the fixed rate is subtracted to the notional principal at each amortization reset date, so that the outstanding notional principal decrease over time.
- Equal — An equal amount is subtracted to the notional principal at each amortization reset date. This amount is equal to the leg's principal notional divided by the number of amortization periods. This is also a decreasing structure over time.
- Step (down) — the structure can here be a subtraction, an addition (by a certain user-defined amount at each amortization reset date), or a multiplication, a division (by a certain user-defined ratio at each amortization reset date).
- Mortgage — You can use a rate that is different from the fixed rate of the trade to amortize the principal. You can determine how long and how frequently the principal is amortized, with or without a residual amount. A mortgage amortization structure is similar to an annuity structure, but supports all daycount conventions and periods of different length. It is a decreasing amortization structure over time.
- Schedule/Custom — user-defined structure. The pricing just checks its validity (no negative notional is allowed).

2.3.3 Compounding

The compound method does provide a period of compounding, a compounding type structure and the compounding index definition (including its index factor and spread).

On a compound cash-flow, interests that might have been earned (or will be owed) accumulate on a compound basis. That is, additional interest will be earned (or owed) on the interest that remains outstanding or is reinvested.

For a compounding swap, there is only one payment date for both floating-rate and fixed-rate payments, and it is at the maturity of the swap.

2.3.4 Averaging

The averaging method does provide a period of averaging, a frequency for the samplings during that period, the samplings weight structure and the averaging index definition (including its index factor and spread).

At each sample date during the averaging period, an implied reset rate is calculated using the index forecast curve, multiplied to its weight and then summed up to build the resultant index rate.

- Period of averaging — This period could be any one-period. The implementation does not allow having two averaging periods within one cash-flow period.
- Frequency of averaging — The frequency of averaging could be almost any one shorter than one year and shorter than the cash-flow frequency. The pricing offers the ability to manually reset the first averaging rate sample and a “cut-off” property to stop the averaging at a particular date.
- Samplings weight structure — For each sampling period, the sum of all weights is 1.0. Two types are available:
    - Equal — all samplings receive the same weight. The average rate is a simple average of all observed rates.
    - Weighted — the samplings receive different weights. The average rate is a weighted average of the observed rates, weighting them by the number of days for which the rate applies.

2.3.5 Convexity Correction

Based on implementation of rate index, e.g. CMS.

This type of correction is applied to the index valuation in its own currency.

The most common and quasi only use is for CMS/CMT indices.

A CMS product is a product that resets according to a constant-maturity swap (CMS) index, i.e. an index representing the fixed rate of a par swap of predefined tenor that is entered into on each reset date. By definition, this kind of index is paid at the end of the reset period (“in arrears”), and therefore a timing adjustment will be performed. And because it is a swap rate (and generally of maturity superior to 1Y), another adjustment is made in order to cope with the equivalent bond rate (of same maturity) expected level.

The pricing of a CMS swap differs from the pricing of a vanilla swap by overriding the reset rate calculation for each cash-flow in order to calculate the underlying forward swap rate and apply the adjustments to such rate. The rest of the pricing relies on the vanilla swap pricing.

We will present the formula of the CMS adjustment when volatilities and correlations are constant during a cash-flow period.

Let’s introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve’s day count).

Each cash-flow i starts at date T_i ,ends at date T_{i+1} , has a reset date $\tilde{T}_i$ (expiration date) and a payment date $\overline{T}_i$

- $L(t,T_i,T_{i+1})$ is the forward Libor rate between T_i ,ends at date T_{i+1} , valuated at date t .
- $\sigma_L(t)$ is the volatility of this forward Libor rate at date t .

- F_t is the forward index rate valuated at date t, here the CMS index rate (non-adjusted)
- $\sigma_F(t)$ is the volatility of this forward index rate at date t.
- $\rho_{LF}(t)$ is the correlation between the forward index rate and the Libor rate at date t.
- $g(y)$ is the function calculating the bond price given its yield rate y
- ($g'(y)$ its one-time derivative with respect to y and $g''(y)$ its two-times derivative with respect to y).

At reset date $\tilde{T}_i$, the pure convexity correction is often presented as a parameter $Conv_i$ and the timing adjustment by a parameter Adj_i as shown in the following equation, giving the expectation at reset date of the forward CMS rate:

$E^{Q_{T_{i+1}}}(F_{T_i}) = (F_{\tilde{T}_i} \cdot (1+Conv_i)) \cdot (1+Adj_i^{IA})$ where $F_{\tilde{T}_i}$ represents the forward price of the CMS non adjusted rate at reset date.

$$Conv_i = -F_{\tilde{T}_i} \cdot \frac{\sigma_F(\tilde{T}_i)^2 \cdot \tilde{T}_i}{2} \cdot \frac{g''(F_{\tilde{T}_i})}{g'(F_{\tilde{T}_i})}$$

and $Adj_i^{IA} = -\rho_{LF}(\tilde{T}_i) \cdot \sigma_F(\tilde{T}_i) \cdot \sigma_L(\tilde{T}_i) \cdot \dfrac{yf_i \cdot L(\tilde{T}_i, T_i, T_{i+1})}{1+yf_i \cdot L(\tilde{T}_i, T_i, T_{i+1})} \cdot \tilde{T}_i$ (as first approximation).

2.3.6 Differential Correction

This type of correction is applied when the leg currency is different to the rate index currency. It is performed after any convexity correction for the rate index in its own currency. The result is a corrected rate forward index value in the leg's currency.

A Differential Swap (or Quanto Swap) is a fixed-floating or floating-floating interest rate swap, where one of the floating rates is a foreign interest rate, but applied to a notional amount in the domestic currency.

With respect to the standard leg pricing, we do modify the reset rate on each cash-flow in order to take into account the diff feature of the index payment (different currencies). The adjustment we make comes from a change of numeraire. The rest of the pricing relies on the vanilla leg pricing.

We will present the formula of the differential adjustment when volatilities and correlations are constant during a cash-flow period. This adjustment does only depend on the unadjusted forward foreign rate, its volatility, the volatility of the foreign exchange rate (FX rate) and the correlation between the FX rate and the forward foreign rate. It does not depend on the current FX spot rate or on the forward FX rate.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

Each cash-flow i starts at date T_i ,ends at date T_{i+1}, has a reset date $\tilde{T}_i$ (expiration date) and a payment date $\overline{T}_i$

- yf_i is the year fraction between T_i and T_{i+1}
- X_t is the forward FX rate at date t.
- $\sigma_X(t)$ is the volatility of this forward FX rate at date t.
- F_t^f is the forward foreign interest rate at date t.

- $\sigma_{F^f}(t)$ is the volatility of the forward foreign interest rate at date t.
- $\rho_{XF^f}(t)$ is the correlation between the forward FX rate and the forward foreign interest rate at date t.

At reset date $\tilde{T}_i$, the adjustment is often presented as a parameter Adj_i^{Diff} as shown in the following equation, giving the expectation at reset date of the forward foreign interest rate in domestic currency:

$$E^{Q^d_{T_{i+1}}}(F^f(T_i)) = F^f_{\tilde{T}_i} \cdot Adj_i^{Diff}$$

The adjustment is then given by the following formula: $Adj_i^{Diff} = e^{\sigma_X(\tilde{T}_i)\cdot\sigma_{F^f}(\tilde{T}_i)\cdot\rho_{XF^f}(\tilde{T}_i)\cdot\tilde{T}_i}$

2.3.7 In Arrears

This type of correction is applied to the payment amount per cash-flow, so it is applied to the corrected rate index in the leg's currency, that is why it is performed at the end of the cash-flow reset rate calculation.

Under an In Arrears product, instead of setting the floating rate at the beginning of the rollover or reset period, we set it at the end of the period. The payment is made as normal at the end of the period, at the setting date.

We will present the formula of the in arrears adjustment when volatilities and correlations are constant during a cash-flow period.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

Each cash-flow i starts at date T_i ,ends at date T_{i+1}, has a reset date $\tilde{T}_i$ (expiration date) and a payment date $\bar{T}_i$

- yf_i is the year fraction between T_i and T_{i+1}
- $L(t, T_i, T_{i+1})$ is the forward Libor rate between T_i ,ends at date T_{i+1}, valuated at date t.
- $\sigma_L(t)$ is the volatility of this forward Libor rate at date t.
- $F(t) = F_t$ is the forward index rate at date t.
- $\sigma_F(t)$ is its volatility of the forward index rate at date t.
- and $\rho_{LF}(t)$ the correlation between the forward index rate and the Libor rate at date t.

At reset date $\tilde{T}_i$, the adjustment is often presented as a parameter Adj_i^{IA} as shown in the following equation, presenting the expectation of the forward rate,

$$E^{Q_{T_{i+1}}}(F_{T_i}) = F_{\tilde{T}_i} \cdot (1 + Adj_i^{IA})$$

The adjustment is then given by

$$Adj_i^{IA} = \frac{yf_i \cdot L(\tilde{T}_i, T_i, T_{i+1}) \cdot (e^{\sigma_L(\tilde{T}_i)\cdot\sigma_F(\tilde{T}_i)\cdot\rho_{LF}(\tilde{T}_i)\cdot\tilde{T}_i} - 1)}{1 + yf_i \cdot L(\tilde{T}_i, T_i, T_{i+1})}$$

2.4 Basis Swap

A basis swap (a floating/floating cross-currency swap) is a swap in which are exchanged two streams of money market floating rates of two different currencies. A notional is also exchanged at the starting of the swap and exchanged back at termination.

A basis swap should not be confused with:

- general cross-currency swap: a basis swap is not necessarily based on two currencies, while a cross-currency swap is not necessarily floating/floating, but can be fixed/floating, floating/fixed and fixed/fixed.
- diff-swap (or quanto swap) which has no exchange of notional.

With respect to a pricing of a vanilla floating/floating swap, the basis swap differs only in the assignment of the discount and forecast curves. The rest of the pricing relies on the vanilla swap pricing.

The quote given for a specific basis swap is the spread such that, when added to the base index rate, this basis swap price is nil (valuation accordingly to the proper curve assignments).

2.5 Cross-Currency Swap

A cross-currency swap valuation is mainly a vanilla swap valuation where the payment amounts in foreign currency are converted to the domestic currency accordingly (using the spot foreign exchange rate defined in the trade).

A cross-currency swap can be floating/floating, floating/fixed (the most common case), fixed/floating or fixed/fixed.

A floating/floating actual exchange cross-currency swap can require that whenever the floating index is reset the notional on one leg be marked to market setting it to the FX rate as of the reset date multiplied by the notional for the other leg. Interest for the current and all subsequent periods will be based on the new notional amount. This adjustment of the notional will produce a net positive or negative payment equal to the original notional amount minus the marked to market notional amount.

A Cross Currency Swap where both legs are floating rate is part of the Basis Swap product family. Cross Currency Swaps are also known as a CIRCA (a Currency and Interest Rate Conversion Agreement).

2.6 Cancelable Swap

A cancelable swap is priced as a swap plus the Bermudan swaption to enter into the reverse swap at the specific reset dates. It is also called putable swap.

2.7 Extendible Swap

An extendible swap is merely priced as the combination of a swap plus a Bermudan swaption to enter into a new equivalent forward swap of a pre-determined maturity at the specific reset dates.

2.8 Capped, Floored, Collared Swap

A capped swap is an interest rate swap where one or both of the floating legs reset a floating rate plus a cap, a floor or collar which exercises at a predetermined trade-specific strike rate and has a maturity equal to the maturity of floating rate. Calypso allows the cap, floor or collar to be of binary type.

2.9 Yield Curve Spread Swap

An yield curve spread swap product is a standard swap (generally a fixed-floating swap) where the index is in fact a combination of two indices I_1 and I_2: $a*I_1 + b*I_2 + c$, where a, b and c are real numbers.

Calypso defines the combination of the two indices as: $I_1 + bI_2 + c_2$, where b and c_2 are real numbers.

2.10 Vanilla CapFloor

2.10.1 Standard

The price of an interest-rate cap is computed as the sum of the prices of each of its caplets, and the price of an interest-rate floor is computed as the sum of the prices of each of its floorlets.

The price of a standard caplet is valued using a Black formula, which assumes that the underlying follows a log-normal process in a complete market.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

$N(x)$ represents the cumulative normal distribution function at x (numerical approximation (Hull, p. 252)).

$N'(x)$ represents its derivative at x (density function).

$$\varepsilon = \begin{cases} +1 \ \ for \ \ a \ \ Cap \\ -1 \ \ for \ \ a \ \ Floor \end{cases}$$

Each cash-flow i starts at date T_i, ends at date T_{i+1}, has a reset date $\tilde{T}_i$ (expiration date) and a payment date $\overline{T}_i$

- N_i is the notional
- yf_i is the year fraction between T_i and T_{i+1}
- df_t is the discount factor between valuation date and t.
- K_i is the strike of the caplet (or floorlet)
- F_t is the forward index rate valuated at date t.
- $\sigma_F(t)$ is the volatility of this forward index rate at date t.
- r_i is the continuous discount rate between valuation date and T_i, $\left(-\left.\frac{\partial df_i}{\partial t}\right|_{T_i}\right)$
- $r_{\tilde{i}}$ is the continuous discount rate between valuation date and $\tilde{T}_i$, $\left(-\left.\frac{\partial df_i}{\partial t}\right|_{\tilde{T}_i}\right)$

The caplet's (floorlet's) formula for a call (put) option on the forward rate (Hull, pp. 537ff) is:

$$caplet_i = df_{\overline{T}_i} \cdot N_i \cdot yf_i \cdot [F_{\tilde{T}_i} \cdot N(d_1) - K_i \cdot N(d_1 - \sigma_F(\tilde{T}_i) \cdot \sqrt{\tilde{T}_i})]$$

$$caplet_i = df_{\overline{T}_i} \cdot N_i \cdot yf_i \cdot [-F_{\tilde{T}_i} \cdot N(-d_1) + K_i \cdot N(-d_1 + \sigma_F(\tilde{T}_i) \cdot \sqrt{\tilde{T}_i})]$$

where $d_1 = \dfrac{Ln(\frac{F_{\tilde{T}_i}}{K_i}) + \frac{\sigma_F^2(\tilde{T}_i) \cdot \tilde{T}_i}{2}}{\sigma_F(\tilde{T}_i) \cdot \sqrt{\tilde{T}_i}}$

The forward rate $F_{\tilde{T}_i}$ is projected from the zero curve associated to the cap interest rate index. The method for projecting the rate is the same as that for swap floating cash-flows.

Note this form explicitly maintains the put-call parity relationship, as the combination of a long position in a caplet and a short position in a floorlet provides the same payment as having a standard simple swap cash-flow paying fixed (the fixed rate K_i) and receiving floating (the interest index rate $F_{\tilde{T}_i}$).

$$caplet_i - floorlet_i = df_{\tilde{T}_i} \cdot N_i \cdot yf_i \cdot [F_{\tilde{T}_i} - K_i]$$

When the option is on an asset paying a continuous dividend at constant rate q, the formulas do not need to be modified as long as the forward index rate incorporates the dividend, $F_{\tilde{T}_i} = S_{\tilde{T}_i} \cdot e^{(r_i - q)\cdot \tilde{T}_i}$ where $S_{\tilde{T}_i}$ is the spot rate at reset date $\tilde{T}_i$.

2.10.2 Amortizing

See Amortizing structures for a vanilla Swap under Amortizing.

2.10.3 Compounding

See Compounding structures for a vanilla Swap under Compounding.

2.10.4 Averaging

See Averaging structures for a vanilla Swap under Averaging.

2.10.5 Convexity correction

See Convexity correction for a vanilla Swap under Convexity correction.

2.10.6 Differential Correction

A Differential Cap/Floor (or Quanto Cap/Floor) is an interest rate Cap/Floor, where the floating rate is a foreign interest rate, but applied to a notional amount in the domestic currency.

See Differential Adjustment for a vanilla Swap under Differential Correction.

2.10.7 In Arrears

See In Arrears Adjustment for a vanilla Swap under In Arrears.

2.10.8 Collar

A Collar CapFloor is the sum of the Cap and a Floor of same characteristics.

2.10.9 Straddle

A Straddle CapFloor is the difference of the Cap minus a Floor of same characteristics.

2.11 Yield Curve Spread CapFloor

See Yield Curve Spread Swap for a definition. The CapFloor has a unique leg. The pricing formula relies on the Kirk (1995) method ("The complete guide to option pricing formulas", p. 59).

It is a Black-Scholes type formula for a European style spread option.

Let's introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

$N(x)$ represents the cumulative normal distribution function at x (numerical approximation (Hull, p. 252)).

The option period starts at valuation date and ends at maturity date T, with expiration date $\tilde{T}$,

- df_t is the discount factor between valuation date and date t.
- $S_{1,t}$ is the first forward index valuated at date t.
- $\sigma_1(t)$ is the volatility of this first forward index at date t.
- $S_{2,t}$ is the second forward index valuated at date t.
- b is the factor for that second forward index.
- $\sigma_2(t)$ is the volatility of this second forward index at date t.
- $\rho_{12}(t)$ is the correlation between the forward first index rate and the forward second index rate at date t.
- c is the spread.
- K is the strike rate.

The pay-off is for a call option $SpreadOpt_{call}(T) = Max(0, S_{1,T} - b \cdot S_{2,T} - c - K)$

We can introduce a modified strike rate: $\overline{K} = K + c$

(We do not consider a factor for the first index, we can always divide the pay-off by that factor to be in the case considered here).

The formula for the call option is then

$$SpreadOpt_{call}(T) = (E^{Q_T}(b \cdot S_{2,T} + \overline{K}) \cdot df_{\tilde{T}} \cdot [F_T \cdot N(d_1) - N(d_1 - \sigma_F(T) \cdot \sqrt{T})]$$

where $F_T = E^{Q_T}\left(\dfrac{S_{1,T}}{b \cdot S_{2,T} + \overline{K}}\right)$

its volatility is

$$\sigma_F(T) = \sqrt{\sigma_1^2(T) + [(b \cdot \sigma_2(T)) \cdot \frac{b \cdot S_{2,T}}{b \cdot S_{2,T} + \overline{K}}]^2 - 2 \cdot \rho_{12}(T) \cdot \sigma_1(T) \cdot (b \cdot \sigma_2(T)) \cdot \frac{b \cdot S_{2,T}}{b \cdot S_{2,T} + \overline{K}}}$$

and $d_1 = \dfrac{\ln(F_T) + \dfrac{\sigma_F^2(T) \cdot T}{2}}{\sigma_F(T) \cdot \sqrt{T}}$

We have the same type of formula for a put option:

$$SpreadOpt_{put}(T) = (E^{Q^T}(b \cdot S_{2,T} + \overline{K}) \cdot df_T \cdot [-F_T \cdot N(d_1) + N(d_1 - \sigma_F(T) \cdot \sqrt{T})]$$

2.12 Basis CapFloor

A basis capfloor, as defined by Calypso, is a vanilla capfloor where the forecast discount curve is a basis curve.

2.13 Digital CapFloor

A digital cap is a strip of digital caplets, each of which is a digital call (or binary call) on the underlying Libor rate.

A digital call is a binary option that pays out a fixed amount if the underlying satisfies a predetermined trigger condition and nothing otherwise.

We distinguish two forms of binary options, and both can be European or American:

- cash-or-nothing
  - A European cash-or-nothing binary pays a fixed amount of money if it expires in the money and nothing otherwise.
  - An American cash-or-nothing binary is issued out the money and makes a fixed payment if the underlying's value ever reaches the strike. The payment can be made immediately or deferred until the option's expiration date.
- asset-or-nothing
  - A European asset-or-nothing binary pays the value of the underlying (at expiration) if it expires in the money and nothing otherwise.
  - An asset-or-nothing binary might be structured as an American option with deferred payment.

Calypso only treats the European style of binary options.

Let introduce the notations first:

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

$N(x)$ represents the cumulative normal distribution function at x (numerical approximation (Hull, p. 252)).

$N'(x)$ represents its derivative at x (density function).

$\hbar(x)$ represents the heaviside function at x. Its definition is the following $\hbar(x) = \begin{cases} 1 \; if \;\; x \geq 0 \\ 0 \; if \;\; x < 0 \end{cases}$.

$$\varepsilon = \begin{cases} +1 \;\; for \;\; a \;\; Cap \\ -1 \;\; for \;\; a \;\; Floor \end{cases}$$

Each cash-flow i starts at date T_i ,ends at date T_{i+1} , has a reset date $\tilde{T}_i$ (expiration date) and a payment date $\overline{T}_i$

- N_i is the notional.
- yf_i is the year fraction between T_i and T_{i+1} .
- df_t is the discount factor between valuation date and t .
- $L(t, T_i, T_{i+1})$ is the forward Libor rate between T_i and T_{i+1} valuated at date t .
- $\sigma_L(t)$ is the volatility of this forward Libor rate at date t .
- F_t is the forward index rate valuated at date t .
- $\sigma_F(t)$ is the volatility of this forward index rate at date t .
- K_i is the strike rate of the caplet.

The pay-off from the asset-or-nothing caplet maturing at time $\tilde{T}_i$ and received at the end of the accrual period $\overline{T}_i$ is $\delta_{\overline{T}_i} = yf_i \cdot \hbar(L(\tilde{T}_i, T_i, T_{i+1}) - K_i)$

The price of an asset-or-nothing binary option is then

$$\delta_{T_n} = \sum_{i=0}^{n-1} N_i \cdot yf_i \cdot df_{\overline{T}_i} \cdot F_{\tilde{T}_i} \cdot N(\varepsilon \cdot d_1)$$

The price of a cash-or-nothing binary option paying an amount A_i for each cash-flow is

$$\delta_{T_n} = \sum_{i=0}^{n-1} N_i \cdot yf_i \cdot df_{\overline{T}_i} \cdot A_i \cdot N(\varepsilon \cdot (d_1 - \sigma_F(\tilde{T}_i) \cdot \sqrt{\tilde{T}_i}))$$

where d_1 is defined as before for a standard caplet.

Calypso multiplies each binary option by an “index factor” (user-input), and the asset-or-nothing binary by a “digital factor” (also user-input).

2.14 Barrier CapFloor

Calypso names “Exotic” the barrier type products. All barrier products do not pay a rebate.

Two distinctive types of barrier options exist:

- Knock-Out — the option becomes worthless in the event that the underlying crosses a certain level.
    - - Up & Out — the underlying spot value starts below the barrier level and crosses it.
    - - Down & Out — the underlying spot value starts above the barrier level and crosses it.
- Knock-In — the option do not exist until a predetermined level is triggered in the underlying.
    - - Up & In — the underlying spot value starts below the barrier level and crosses it.
    - - Down & In — the underlying spot value starts above the barrier level and crosses it.

As Calypso defines a barrier CapFloor, it can only be an Up and Out or an Up and In CapFloor.

When no rebate is paid, the strategy consisting being long a CapFloor Down and Out plus a CapFloor Down and In with the same barrier level and the same strike is equivalent to a simple Cap with the same strike. We have the same result for Floors. This allows computing, in this case, the prices of a CapFloor Down and Out, as well as a CapFloor Down and In.

2.15 European Swaption

A standard or vanilla swaption is an option on a swap interest rate. Its pricing is commonly based in the assumption that the underlying (here the swap interest rate) follows a log-normal process in a complete market.

The option itself can be one of the following types:

- Call — the option to buy the underlying at a specific value at maturity or exercise date.
- Put — the option to sell the underlying at a specific value at maturity or exercise date.

Once exercised, the user enters into the corresponding underlying swap, which is in general a vanilla type swap.

2.15.1 PricerSwaption

In the case of a European swaption, a Black formula can be used. Let’s introduce the notations first:

- $N(x)$ represents the cumulative normal distribution function at *x*.
- N is the notional.

- $D(t)$ is the discount factor between valuation date and t.
- T is the exercise date of the option.
- K is the strike of the option (in Calypso, entered as fixed rate for the underlying swap).
- S is the forward swap rate valuated at the valuation date (the BREAK_EVEN_RATE of the underlying swap).
- σ is the volatility of the forward swap rate.
- a_i is the accrual period length in years corresponding to the i^{th} accrual period on the fixed leg.
- t_i is the pmt date of the i^{th} interest flow on the fixed leg.

Then first one determines the so-called *annuity factor*, *A*, as

$$A = \sum_i a_i D(t_i)$$

For a payer swaption, the valuation formula is:

$$V = NA[SN(d_1) - KN(d_2)]$$

For a receiver swaption, the valuation formula is:

$$V = NA[-SN(-d_1) + KN(-d_2)]$$

where $d_1 = \dfrac{\ln\left(\frac{S}{K}\right) + \frac{\sigma^2 T}{2}}{\sigma\sqrt{T}}$ and $d_2 = d_1 - \sigma\sqrt{T}$

The above valuation applies to the most vanilla cases. For various deviations away from the vanilla case, a variety of ad-hoc adjustments are made by practitioners; for example, the cases of, non-zero spread on the float leg, step-up/down coupons and amortizing notionals. The following section describes the ad-hoc adjustments we apply.

Adjustment for floating spread, step-up/down coupon and/or notionals

Suppose a spread of *s*, is paid on the float leg of the swap. We abuse the notation used above for convenience and index with *j* to mean a quantity on the float leg, whilst index with *i* means the same quantity on the fixed leg. We do not assume that the notionals or the accrual period lengths need be the same, generally they are not. The value of each leg of the swap is computed as;

$FloatLegPV = \sum_j a_j F_j N_j D(t_j) + \sum_j a_j s_j N_j D(t_j)$, where *F* denotes the forecast rate of the floating index

$$FixedLegPV = \sum_i a_i K_i N_i D(t_i)$$

Then the forward swap rate, *S*, is determined as;

$$S = \frac{FloatLegPVNoSpread}{FixedLegAnnuity} = \frac{\sum_j a_j F_j N_j D(t_j)}{\sum_i a_i N_i D(t_i)}$$

Whilst the so-called *effective strike*, K^*, is determined as;

$$K^* = \frac{FixedLegPV - FloatSpreadPV}{FixedLegAnnuity} = \frac{\sum_i a_i K_i N_i D(t_i) - \sum_j a_j s_j N_j D(t_j)}{\sum_i a_i N_i D(t_i)}$$

Now the PV of the underlying swap of a payer swaption can be written evocatively as;

$$\begin{aligned} A.[S - K^*] &= \sum_j a_j F_j N_j D(t_j) + \sum_j a_j s_j N_j D(t_j) - \sum_i a_i K_i N_i D(t_i) \\ &= FloatPV - FixedPV \end{aligned}$$

where, *A*, abuses the notation above, and is now defined as $A = \sum_i a_i N_i D(t_i)$.

The swaption is then valued by computing the expectation;

$$V = A.\mathrm{E}[\max(S - K^*, 0)]$$

Volatility for amortizing notional European swaptions

Swaptions whose underlying swaps are defined with varying notionals; for example, amortizing, accumulative or ‘roller-coaster’ notionals, are priced using a volatility derived the market volatility of vanilla swaptions. This volatility is calculated by decomposing the swaption into multiple vanilla swaptions (with constant notionals) and then calculating a weighted average of their respective volatilities. Two methods are available and are defined on the floating leg notionals N_j with projected cashflows $c_j = a_j F_j$. The Boolean pricing parameter AMORT_VOL_OVLP controls which method is to be used (AMORT_VOL_OVLP=true, means used the overlapping method).

Overlapping Method

In the overlapping method, the cashflows on the underlying amortizing swap are replicated by the cashflows on vanilla swaps that share cashflow dates. For example, a decreasing-notional amortized swap, the vanilla swap with label *k* has notional amount

$$H_k = \begin{cases} N_k - N_{k+1} & k \neq n \\ N_n & k = n \end{cases}$$

Let *b(k)* denote the index of the original cashflow date with which the vanilla swap begins, and *e(k)* the index of the cashflow with which it ends. For example, a decreasing-notional swap, the k^{th} vanilla swap has:

b (k) = 1

e (k) = index of last cashflow with notional N_k .

For more general roller-coaster amortized swaps, where the notionals can alternately increase or decrease, the calculations of H_k, *b(k) and e(k)* are more complex. In all cases, the approach is to find the longest fixed-notional swaps possible that, when combined, equal the amortized swap.

Non-overlapping Method

In the non-overlapping method, the replicating vanilla swaps share no cashflow dates. The end date of one swap is essentially the start date of the next swap. The swap with label *k* has notional amount

$$H_k = N_j$$

and has start and end cashflow indexes

b (k) = index of first cashflow with notional N_j
e (k) = index of last cashflow with notional N_j

Weighted Volatility

In either method the volatility used to plug into the standard Black76 formula is

$$\sigma^2 = \sum_{i,j} w_i \sigma_i w_j \sigma_j$$

where, the weighting factor w_k associated with the k^{th} component swaption is determined as

$$w_k = \frac{H_k \sum_{j=b(k)}^{e(k)} c_j D(t_j)}{\sum_l \left\{ H_k \sum_{j=b(k)}^{e(k)} c_j D(t_j) \right\}}$$

whilst the σ_k is of course the market volatility associated with the k^{th} component swaption. In the case of the non-overlapping method the swaps are typically forward starting with respect to the expiry date. This delayed start is ignored when identifying the market volatility.

Cash Settled Swaption

Cash settled swaptions can be settled using several different methods. The methods we consider are

- Zero Coupon,
- Par Curve Adjusted,
- Par Curve Unadjusted

The details are well defined in the ISDA 2000 definitions. In the case of zero coupon cash settlement, the valuation method is the same at that described above. In the cases of the par curve methods we account for the different payoff by making a simple adjustment to the annuity factor in a similar fashion to Brigo and Mercurio (2001)[1].To avoid confusion we introduce the notation of *forecasting annuity factor*, A_F , and *discounting annuity factor*, A_D. The forecasting annuity discount factor is used to determine the forecast swap rate, namely

$$A_F = \sum_i a_i N_i D(t_i)$$

The key step in the approximation is to alter the discounting factor by discounting using the forecast swap rate, until the delivery date, and then discount from the deliver date back to the valuation date using the current yield curve; that is,

$$A_D = D(t_D)\left(\frac{N_0\tau_0}{1+S\tau_0} + \frac{N_1\tau_1}{(1+S\tau_0)(1+S\tau_1)} + \ldots + \frac{N_n\tau_n}{(1+S\tau_0)(1+S\tau_1)\ldots(1+S\tau_n)}\right)$$

where, t_D, denotes the delivery date of the option, whilst τ_i, denotes the accrual period length. In the case, of *par curve adjusted*, τ_i, denotes the accrual period length of the adjusted period, whilst in the case of *par curve unadjusted*, τ_i, denotes the accrual period length of the unadjusted period. Then the cash settled swaption is valued as,

$$V = A_D.\mathrm{E}[\max(S - K^*, 0)]$$

[1] Brigo, D. and Mercurio, F. (2001), “Interest Rate Models: Theory and Practice”, Springer-Verlag, Berlin Heidelberg New York

2.15.2 PricerSwaptionBpVol

This pricer is similar to the functionality in PricerSwaption except the underlying swap rate is modeled as following a normal process (as opposed to lognormal in Black's model), specifically the swap rate *S* is modeled with the following S.D.E.

$$dS = \sigma dW$$

The volatility parameter is often referred to as basis point volatility, or just BpVol. Generally, the volatility is different in magnitude, and can be related using the following rule of thumb,

$$\sigma_{BPVOL} \approx S\upsilon_{BLACK}$$

So, for example if the Black's vol is 20% and the spot swap rate is 5%, then the equivalent BPVol is approximately 1% or 100 basis points.

The following pricer parameters are new compared with PricerSwaption.

Pricing Parameters	Type	Description	Typical Value
BP_VOL_TRANSFORMATION	String	This field controls which method is used by the pricer measure BLACK_EQUIV_VOL when converting from the basis point volatility used in this pricer to a Black equivalent volatility. Current methods are EXACT, HAGAN_APPROX,STREET_PROXY1, STREET_PROXY2, STREET_PROXY3, STREET_PROXY4, STREET_PROXY5 Details of each method are found in Calypso's dedicated model document "Basis point volatility"	EXACT
VOLATILITY	Rate	In this pricer the volatility means the basis point although the display is in Rate terms, so for example a typical value is 100bp, one enters a rate of 0.01	0.01

In addition to the usual pricer measures of PricerSwaption, the following additional pricer measures are available.

Pricer Measure	Description	Caveats
BLACK_EQUIV_VOL	Given the current basispoint volatility used in pricing within PricerSwaptionBpVol, this measure is the equivalent volatility one would need to plug into Black's model in order to have the same NPV.	Not supported on straddles
BP_VOL	The current model volatility used in pricing	Not supported on straddles
NVEGA_BP_VOL	The change in NPV given a 1b.p. shift in the basis point volatility	

Volatility Surface Configuration

The pricer expects a volatility surface with point adjustments ASK_BPVol, MID_BPVol, BID_BPVol. An example of a generator which creates such a surface is SwaptionBpVols.

2.15.3 PricerSwaptionCEV

This pricer is similar to the functionality in PricerSwaption except the underlying swap rate is modeled as following a so-called constant elasticity of variance (CEV) process (as opposed to lognormal in Black's model), specifically the swap rate *S* is modeled with the following S.D.E.

$$dS = \alpha S^{\beta} dW$$

The volatility parameter alpha (α) is often referred to as basis point volatility, or just BpVol. Generally, the volatility is different in magnitude, and can be related using the following rule of thumb,

$$\upsilon_{BLACK} \approx \frac{\alpha_{CEV}}{S^{1-\beta}}$$

So, for example if the cev model volatility 3%, beta = 0.5 and the spot swap rate is 4%, then the equivalent Black is approximately 15%.

The following pricer parameters are new compared with PricerSwaption.

Pricing Parameters	Type	Description	Typical Value
BP_VOL_TRANSFORMATION	String	This field controls which method is used by the pricer measure BLACK_EQUIV_VOL when converting from the basis point volatility used in this pricer to a Black equivalent volatility. Current methods are EXACT, HAGAN_APPROX,STREET_PROXY1, STREET_PROXY2, STREET_PROXY3, STREET_PROXY4, STREET_PROXY5 Details of each method are found in Calypso's dedicated model document "Basis point volatility"	EXACT
CEV_BETA	Amount	This is the beta parameter of the CEV model.	0.5
CEV_ALPHA	Rate	This is the volatility parameter of the CEV model. Use pricer measure BLACK_EQUIV_VOL to get a sense of the model parameter for a given value of beta.	0.01- 30%, depending on beta.
CEV_VALUATION_METHOD	String	Choices are EXACT, HAGAN_WOODWARD_HIGHORDER, HAGAN_WOODWARD_LOWORDER, ANDERSON_RATCLIFFE_DD. Several approximations are available as well the exact valuation. Generally, these approximations are very accurate so little difference is noticed between each method. More details are available in the dedicated document "CEV and displaced diffusion models"	HAGAN_WOODWARD_HIGHORDER

In addition to the usual pricer measures of PricerSwaption, the following additional pricer measures are available.

Pricer Measure	Description	Caveats
BLACK_EQUIV_VOL	Given the current cev model used in pricing within PricerSwaptionCEV, this measure is the equivalent volatility one would need to plug into Black's model in order to have the same NPV.	Not supported on straddles
BP_VOL	Given the current cev model used in pricing within PricerSwaptionCEV, this measure is the equivalent volatility one would need to plug into the normal (or bpvol) model in order to have the same NPV.	Not supported on straddles
D_CEV_ALPHA	The sensitivity to a change in alpha	
D_CEV_BETA	The sensitivity to a change in beta	

Volatility Surface Configuration

The pricer expects a volatility surface with point adjustments CEV_ALPHA, CEV_BETA. An example of a generator which creates such a surface is SwaptionCEV.

2.16 Bermudan Swaption

A Bermudan swaption is an option on a swap that can be exercised at predetermined exercise dates. The underlying swap can be defined in two ways;

- Fixed End Date – the underlying swaps corresponding to the exercise dates all shared the same end-date
- Fixed Tenor – the underlying swaps corresponding to the exercise dates all have the same relative maturity

2.16.1 PricerSwaptionLGMM

PricerSwaptionLGMM supports the valuation of Bermudan swaption, either fixed end date or fixed tenor, using the so-called LGM model, a term coined by P.Hagan in an unpublished, but widely known, working paper[2]. The LGM model is precisely the Hull-White one factor model expressed as an HJM model. Further details on the Calypso are found in a dedicated Calypso analytics document analytics document, “Linear Gauss Markov model”.

Pricing Parameters

Pricing Parameter	Type	Description	Typical Value
LGMM_MEAN_REV	Rate	A transient override for the mean reversion parameter.	-1% to 5%
LGMM_MODEL_VOL	Rate	A transient override for the model's volatility parameter.	1%
LGMM_IR_RATE	Rate	A transient override for the yield curve.	1%-6%

[2] The paper is “Methodology for callable swaps and Bermudan ‘exercise into’ swaptions”, P.S.Hagan circulated and discussed on the Wilmott forums (www.wilmott.com).

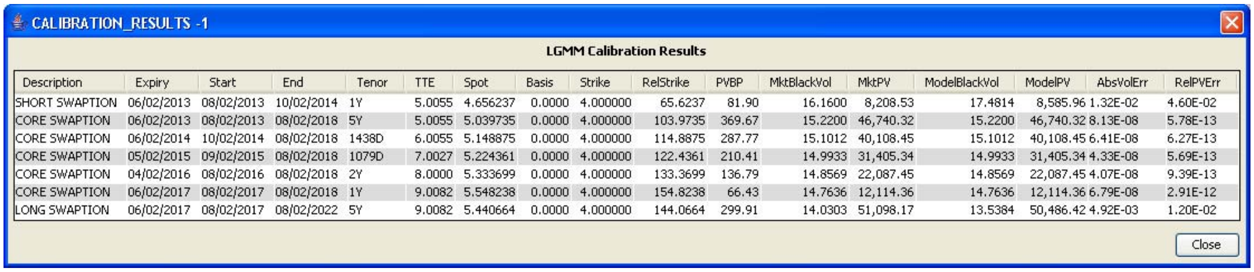
Pricing Parameter	Type	Description	Typical Value
LGMM_CALIBRATION_INSTRUMENTS	Choice	CORE_SWAPTION – the calibration instrument are European swaptions defined by the exercise dates and the swap related to the exercise date. On a typical 10Y NC 5Y, the core swaption are 5Yx5Y, 6Yx4Y, 7Yx3Y, 8Yx2Y, 9Yx1Y. On fixed-tenor (or trombone) swaption, the core swaptions would be 5Yx5Y, 6Yx5Y, 7Yx5Y, 8Yx5Y, 9Yx5Y Also the strikes are at the strike of the current Bermudan swaption. CORE_AND_SHORT_SWAPTION – the calibration instruments are the core swaptions described above and one additional swaption, defined by the first (alive)exercise date of a maturity the same length as the fixed coupon length. In the previous example of a 10Y NC 5Y it would correspond to a 5Yx1Y swaption. CORE_SWAPTION_ATM – Same as CORE_SWAPTION except the strikes are chosen to be at the money, rather than the same as the trade. CORE_AND_SHORT_SWAPTION_ATM- Same as CORE_AND_SHORT_SWAPTION except the swaptions are at the money.	CORE_SWAPTION
LGMM_CALIBRATION_SCHEME	Choice	EXACT_STEP_SIGMA – the model volatility function is a step function, chosen so as to match the calibration instruments exactly BEST_FIT_LM – the model mean reversion and volatility are constant and chosen by a Levenberg-Marquardt best fit routine applied to the calibration instruments. APPROX_STEP_SIGMA – same as EXACT_STEP_SIGMA except and using a faster but approximate method.	EXACT_STEP_SIGMA
LGMM_CONTROL_VARIATE	Boolean	When pricing the Bermudan, also price the first European numerically and use it as a control variate.	FALSE
LGMM_LATTICE_NODES	Integer	The number of nodes in the discretisation of the state space of the markov process.	35
LGMM_QUAD_ORDER	Integer	The number of point in the local quadrature rule used in the roll-back routine.	20
LGMM_LATTICE_CUTOFF	Double	The number of deviations to the outer model node in the state space	6

Pricing Parameter	Type	Description	Typical Value
		discretisation.	
LGMM_RISK_OPTIMISE	Boolean	Controls whether or not optimization techniques are used within scenario analysis, in particular for shift and revalues of the volatility surface.	TRUE
LGMM_MIN_MEAN_REVERSION	Rate	When using CALIBRATION_SCHEME=BEST_FIT_LM the user can control the minimum level of mean reversion permitted within the calibration.	-1%
LGMM_MAX_MEAN_REVERSION	Rate	When using CALIBRATION_SCHEME=BEST_FIT_LM the user can control the maximum level of mean reversion permitted within the calibration.	5%
LGMM_MIN_SIGMA	Rate	When using CALIBRATION_SCHEME=BEST_FIT_LM the user can control the minimum level of model volatility permitted within the calibration.	0.01%
LGMM_MAX_SIGMA	Rate	When using CALIBRATION_SCHEME=BEST_FIT_LM the user can control the maximum level of model volatility permitted within the calibration.	2%
LGMM_BEST_FIT_GRAPH_MESH_SIZE	Integer	When the PricerMeasure LGMM_BEST_FIT_ERR is used, this parameter controls how fine the mesh used in the brute force search is.	30

Pricer Measures

Calibration results

When one selects the pricer measure CALIBRATION_RESULTS the pricer will show details of the intermediate steps in the calibration.


CALIBRATION_RESULTS -1

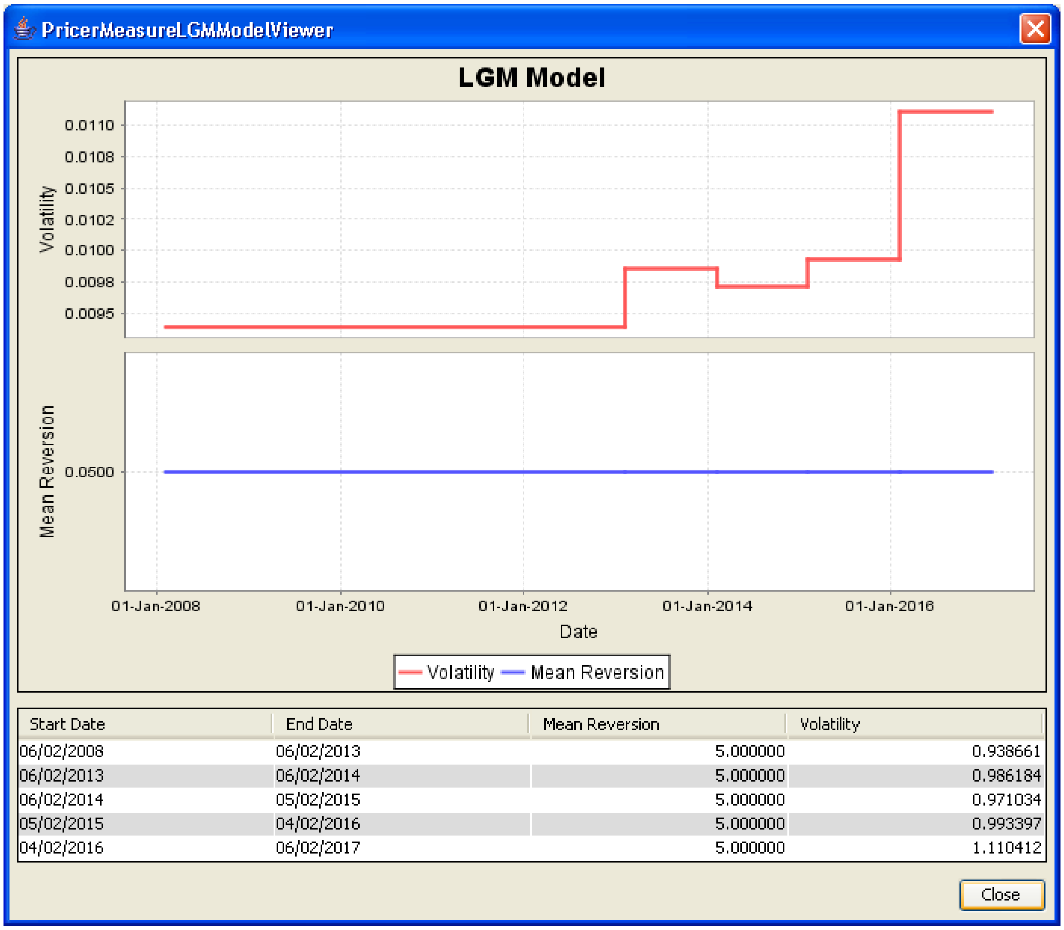
LGMM Calibration Results

Description	Expiry	Start	End	Tenor	TTE	Spot	Basis	Strike	RelStrike	PVBP	MktBlackVol	MktPV	ModelBlackVol	ModelPV	AbsVolErr	RelPVErr
SHORT SWAPTION	06/02/2013	08/02/2013	10/02/2014	1Y	5.0055	4.656237	0.0000	4.000000	65.6237	81.90	16.1600	8,208.53	17.4814	8,585.96	1.32E-02	4.60E-02
CORE SWAPTION	06/02/2013	08/02/2013	08/02/2018	5Y	5.0055	5.039735	0.0000	4.000000	103.9735	369.67	15.2200	46,740.32	15.2200	46,740.32	8.13E-08	5.78E-13
CORE SWAPTION	06/02/2014	10/02/2014	08/02/2018	1438D	6.0055	5.148875	0.0000	4.000000	114.8875	287.77	15.1012	40,108.45	15.1012	40,108.45	6.41E-08	6.27E-13
CORE SWAPTION	05/02/2015	09/02/2015	08/02/2018	1079D	7.0027	5.224361	0.0000	4.000000	122.4361	210.41	14.9933	31,405.34	14.9933	31,405.34	4.33E-08	5.69E-13
CORE SWAPTION	04/02/2016	08/02/2016	08/02/2018	2Y	8.0000	5.333699	0.0000	4.000000	133.3699	136.79	14.8569	22,087.45	14.8569	22,087.45	4.07E-08	9.39E-13
CORE SWAPTION	06/02/2017	08/02/2017	08/02/2018	1Y	9.0082	5.548238	0.0000	4.000000	154.8238	66.43	14.7636	12,114.36	14.7636	12,114.36	6.79E-08	2.91E-12
LONG SWAPTION	06/02/2017	08/02/2017	08/02/2022	5Y	9.0082	5.440664	0.0000	4.000000	144.0664	299.91	14.0303	51,098.17	13.5384	50,486.42	4.92E-03	1.20E-02

Close

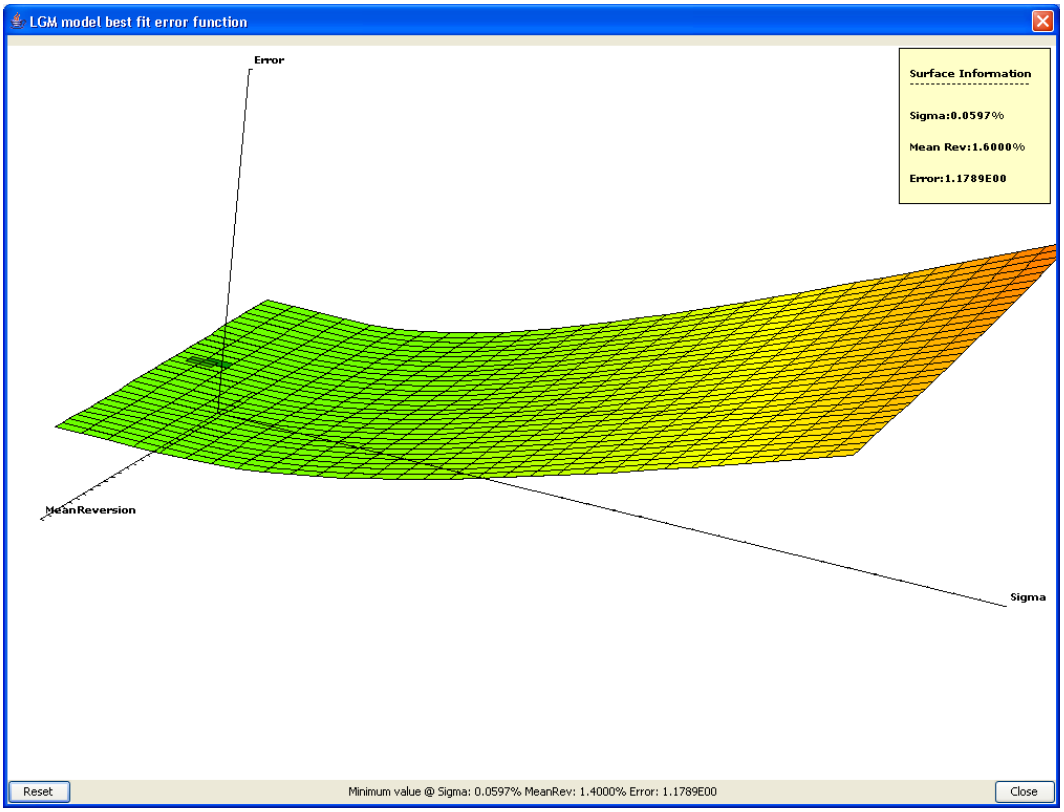
LGM Model

When one selects the pricer measure LGM_MODEL the pricer will show the LGM model with values used when computing the NPV.



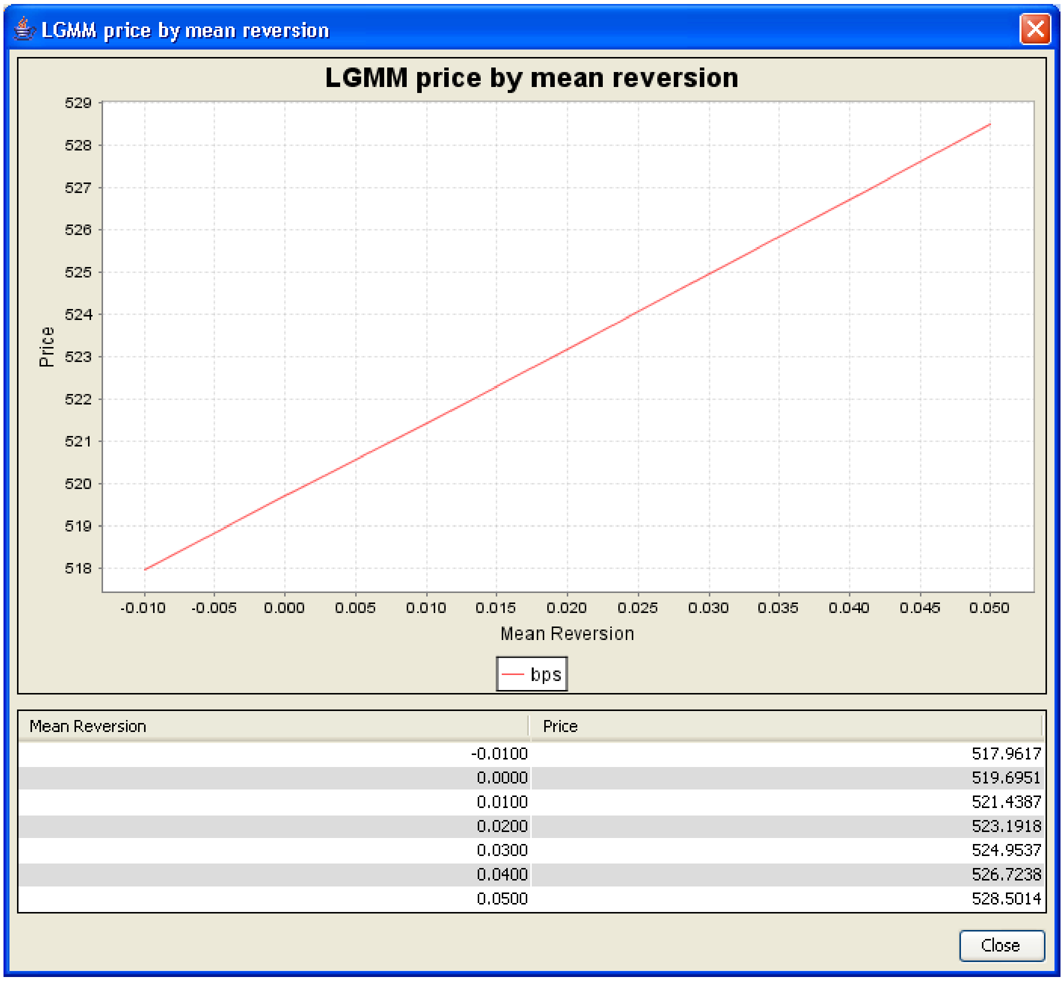
LGMM Bestfit Err

When one selects the pricer measure LGMM_BESTFIT_ERR the pricer will do an additional calculation, and search in a brute force fashion over a range of mean reversion and sigma values, and display the best-fit error function. Note: The calibration scheme BEST_FIT_LM does not use this brute-force method, the brute-force method is simply for the user to get a feel for the error function and double check the BEST_FIT_LM calibration.



LGMM Mean Reversion Scenario

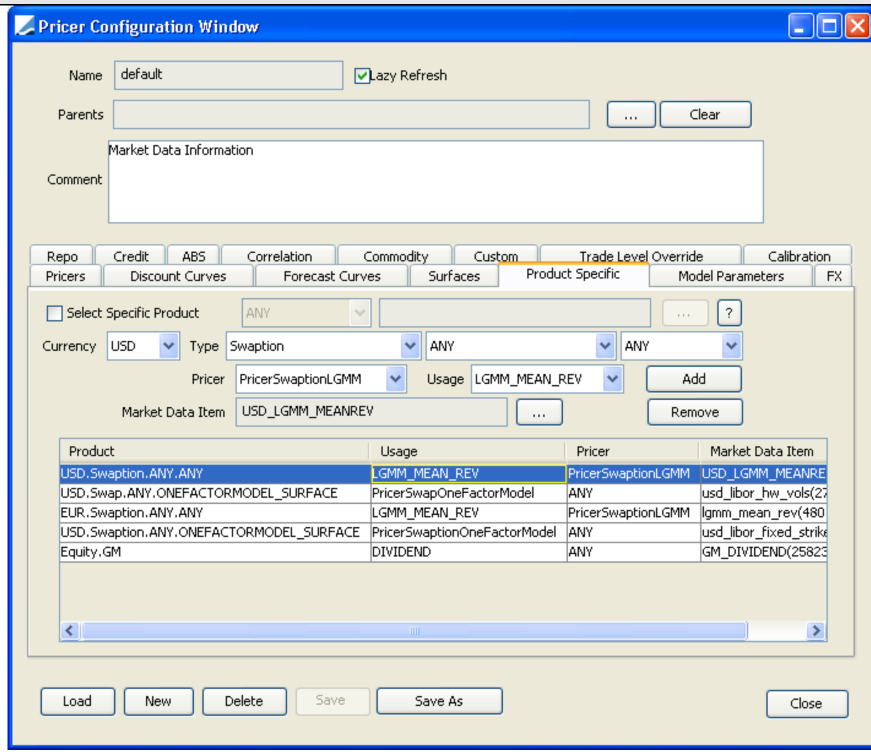
When one selects the pricer measure LGMM_MEANREV_SCEN, the pricer will do additional valuation, specifically, will vary the constant mean reversion parameter and recompute the NPV. The purpose is give the user a sense of how the Bermudan price varies with the mean reversion parameter.

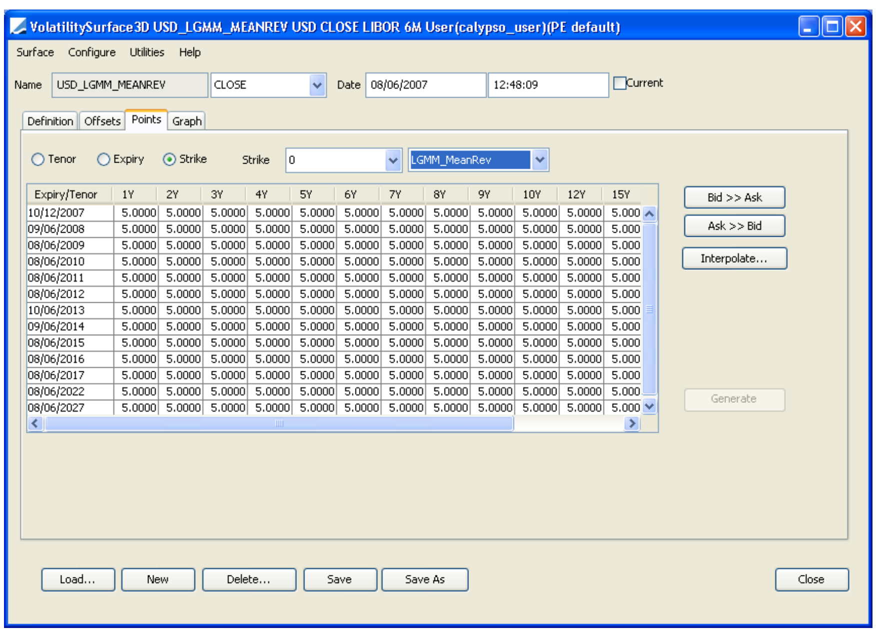


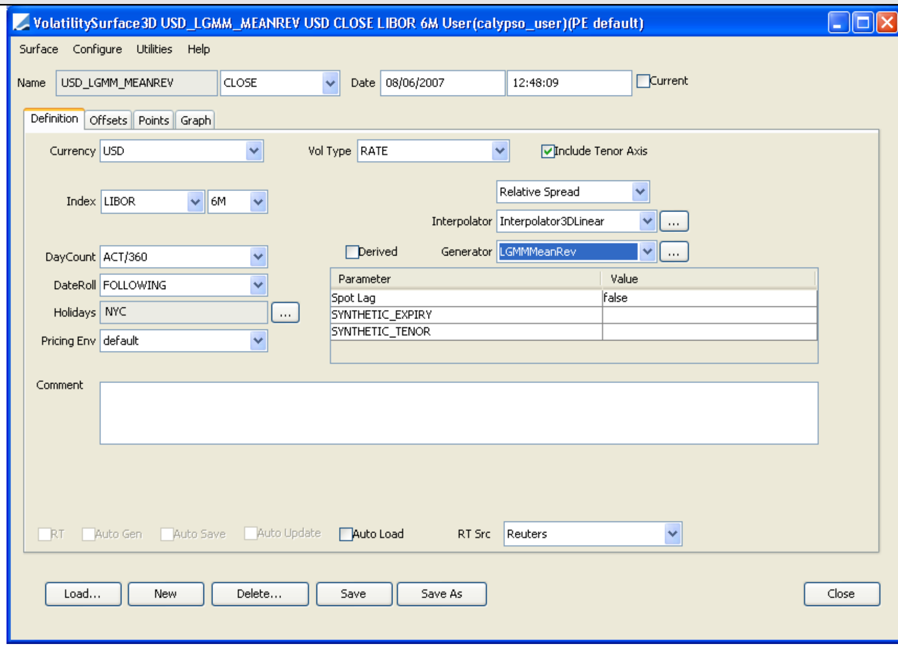
Market Data Configuration

Market Data	Description
Discount Curve	Same as PricerSwaption
Forecast Curve	Same as PricerSwaption
Swaption Volatility	Same as PricerSwaption
LGMM Mean Reversion Matrix	The pricer expects a mean reversion matrix to be configured within the pricer config. Specifically, in the product specific tab one must a volatility surface whose generator is LGMMMeanRev, and assign it the PricerSwaptionLGMM where the Usage=LGMM_MEAN_REV

Market Data	Description





Market Data	Description
	

References

- Hagan, P.S. and Woodward, D.E. (1999), “Markov Interest Rate Models”, Applied Mathematical Finance, 6:233-260
- Calypso Technology, “Linear Gauss Markov Model”.

2.16.2 PricerSwaption

The valuation will is based on the Black Model (Black [1976]) already used in PricerSwaption to value European option. The methodology is to identify the natural hedging instruments, namely the core European swaptions; that is, the European option with the same exercise date and underlying swap as those defined at each exercise date on the Bermudan.

The valuation is then obtained by adding the present value of the swaption with the maximum value to a value calculated by multiplying the weighted average of the other option premiums based on the probability that they will be exercised ($N(d_2)_i$) by the probability that the said swaption with the maximum value will be out of money, ($1-N(d_2)_{\max}$); that is,

$$V = V_{\max} + \left(1 - N(d_2)_{\max}\right)\left\{\sum_{i \neq \max} \frac{N(d_2)_i}{\left[\sum_{j \neq \max} N(d_2)_j , 1\right]_+}\left(V_i - [U_i, 0]_+\right)\right\}$$

where:

$[a, b]_+$	Max in two element a, b
$V_{\max}$	Present value of the European option with the maximum value

$\left(1 - N(d_2)_{\max}\right)$	Probability that the option with the maximum value will not be exercised
$\left(V_i - [U_i, 0]_+\right)$	Intrinsic value of European option (U_i : Underlying swap PV (fixed side))
	Weighted average probability based on the probability that each option will be exercised

2.17 American Swaption

An American swaption is an option on a swap that can be exercised any time during the option expiration period.

2.17.1 PricerSwaption

PricerSwaption approximates the American swaption with a Bermudan swaption with a frequent exercise schedule.

2.17.2 PricerSwaptionOneFactorModel

PricerSwaptionOneFactorModel can price and American swaption.

2.18 Barrier Swaption (Trigger Swaption)

A trigger swaption, as defined by Calypso, is a European style option on a swap which only exercises if an external condition is reached (a trigger index rate is above or below a trade-specific trigger strike rate).

Let's introduce the notations first

All dates are calculated using the appropriate day count usage (the corresponding curve's day count).

$N(x)$ represents the cumulative normal distribution function at x (numerical approximation (Hull, p. 252)).

The option period starts at valuation date and ends at maturity date T , with expiration date $\tilde{T}$,

- N is the notional.
- df_t is the discount factor between valuation date and t .
- The trigger index is here a Libor rate of maturity Δ_L ... compared to trigger strike rate K_S
- S_t is the forward trigger index rate between T and $T + \Delta_L$ valuated at date t .
- $\sigma_S(t)$ is the volatility of this forward trigger index rate at date t .
- K_S is the trigger strike rate.

The underlying is a swap

- F_t is the forward swap rate (the forward index rate for the option) valuated at date t .
- $\sigma_F(t)$ is the volatility of this forward swap rate at date t .
- K is the strike rate of the option

- $\rho_{SF}(t)$ is the correlation between the forward trigger rate and the forward index rate at date t.

The pay-off at maturity of the European trigger pay-fixed swaption is:

$$TrigSwtn(T) = Max(F_T - K, 0)\Big|_{\left\{E^{Q_T}(S_T) - K_S \geq 0\right\}}$$

The formula becomes for the European trigger pay-fixed swaption:

$$TrigSwtn(T) = N \cdot df_T \cdot [E^{Q_T}(F_T) \cdot N(d_1) - K \cdot N(d_1 - \rho_{LF}(T) \cdot \sigma_F(T) \cdot \sqrt{T})]$$

where $d_1 = \dfrac{Ln(\dfrac{S_T}{K_S}) - \dfrac{\sigma_S^2(T) \cdot T}{2}}{\sigma_S(T) \cdot \sqrt{T}} + \rho_{LF}(T)\sigma_F(T)\sqrt{T}$

2.19 CMS/InAdvance/InArrears Swap

- PricerSwapHagan (PricerSingleSwapLegHagan) -

The pricers PricerSwapHagan and PricerSingleSwapLegHagan implement the methods described in

Hagan, P. (2003), "Convexity conundrums: pricing CMS swaps, caps and floors", *Wilmott*, (Mar.): 38-44

In this article Hagan developments additional methods to the bond math method for calculating the convexity corrections for Swaps, Caps, and Floors with CMS indexes or in-arrears payments. He develops several approximations, each progressively stronger, the last of which produces near perfect results. His final method, the replication method, uses prices of several payer and receiver vanilla swaptions to replicate the payoff of the caps and floors.

Originally in Calypso, and still in use with PricerSwap, the convexity adjustment was performed using the bond math as developed in Hull. Now we implement other ways of computing this correction so there are several ways to set up the CMS index in Calypso.

The pricers supporting the Hagan convexity methodology are PricerHagan, PricerCapFloorHagan, and PricerSpreadCapFloorGBM2FHagan.

There are many ways available for incorporating this convexity correction. The default behavior is to use the replication method in the CMS case (pricing parameter HAGAN_SWAP_BY_REPLICATION = true) and an approximation in the LIBOR in-arrears case (pricing parameter HAGAN_CASH_BY_REPLICATION = false and HAGAN_CASH_YIELD_CURVE_MODEL = linear) as described in Hagan's paper. The user is given the ability to override these methods by specifying the CMS calculations directly, if there is another preferred method.

A volatility surface may store the CMS adjustments, if desired. In this case a volatility surface generated by the CMSBasisAdj is required to create a layer of points on the points panel titles CMS_BASIS_ADJ. The pricer will look for this layer on the volatility surface if the pricing parameter HAGAN_SWAP_USE_BASIS_ADJ is set to true.

Pricing Parameters	Type	Description	Typical Value
HAGAN_COMPUTE_CORRECTION	Boolean	This field controls whether or not the convexity correction should be computed at all	True
HAGAN_CASH_BY_REPLICATION	Boolean	This field controls whether a cash index such 6M-LIBOR applies convexity correction by replication or analytic approximation	False
HAGAN_SWAP_BY_REPLICATION	Boolean	This field controls whether a swap index such 20Y-CMS-LIBOR applies convexity correction by replication or analytic approximation	True

Pricing Parameters	Type	Description	Typical Value
HAGAN_USE_EXACT_CONVEXITY_FUNC	Boolean	This field is only applicable to the case of valuation by replication for either cash or swap index. True – Use the function defined in Eq. 2.15 of Hagan's article to describe the convexity correction payoff. False – Use the quadratic approximation function defined in Eq. 3.1b of Hagan's article.	True
HAGAN_CASH_YIELD_CURVE_MODEL	List	STANDARD_BOND – Represents the yield curve model "Model 1: Standard model" described in Appendix A.1 of Hagan's article EXACT_BOND – Represents the yield curve model "Model 2: Exact yield model" described in Appendix A.2 of Hagan's article LINEAR– Represents the yield curve model Linear Swap Rate described in, Hunt, P.J. and Kennedy, J.E. (1998) "Financial Derivatives in Theory and Practice", Wiley & Sons, 1st Ed.	LINEAR
HAGAN_SWAP_YIELD_CURVE_MODEL	List	Same list of choices as CASH_YIELD_CURVE_MODEL.	EXACT_BOND
HAGAN_CASH_THRESHOLD	Integer	A correction on a cash index is only made if the forecast end date is significantly different from the payment date. The parameter represents the number of calendar days after which a correction should be made. The point is that on a vanilla swap it is not unusual to have a situation where the forecast end date is different from the payment date by one or two days due to subtle aspects of date generation. In such a case, it may not be worthwhile attempting to apply a convexity correction.	7

2.20 Digital/CMS/InAdvance/InArrears Cap

- PricerCapFloorHagan -

The PricerCapFloorHagan follows the same methodology described above in PricerSwapHagan. Some more details are needed and described here.

Digital caps

Digital caps are priced by the call spread method. In this case two further pricing parameters are needed, namely the spread and the direction of the spread.

Pricing Parameter	Type	Description	Typical Value
STRIKE_SPREAD_EPSILON	Double	This is the size of the spread between the strikes of the call spread. Usually of the order of basis points.	5-10bp

Pricing Parameter	Type	Description	Typical Value
STRIKE_SPREAD_DIRECTION	List	This is the direction of the spread relative to the strike. Denote the strike of the digital as *K*, and the spread as *eps* SUPER – strikes at K and K+eps CENTRAL – strikes at K-0.5*eps, K+0.5*eps SUB – strikes at K-eps and K	CENTRAL

2.21 (Digital) Yield Curve Spread Cap

- PricerSpreadCapFloorGBM2F (Hagan) -

Spread Option Valuation

The methodology used to evaluate the spread option follows closely the observations of Pelsser (2000), which assume a 2 factor geometric brownian motion model (GBM2F).

Index Forwards Calculation

In the case of PricerSpreadCapFloorGBM2FHagan, the pricer compute the index forwards using the methodology7 of Hagan(2003). This is the same method used in PricerSwapHagan.

In the case of PricerSpreadCapFloorGBM2F, the pricer computes the index forwards using the classic convexity correction and timing adjustments described in Hull's textbook. This method is consistent with PricerSwap.

Pricing Parameter	Type	Description	Typical Value
QUAD	List	The quadrature used to evaluate the 1D integral. Hermite – Gauss-Hermite quadrature Legendre – Gauss-Legendre quadrature	LEGENDRE
QUAD_POINTS	Integer	The number points on the quadrature. Legendre supports all values, whilst Hermite has onl 7 and 30 point rules implemented	30
USE_SMILE_VOL	Boolean	This flags allows one to use volatilities other than the ATM volatility for each index when evaluation the spread option. When using the non-ATM volatility, we follow the so-called partial smile model of Berrahoui,M. (2004) "Pricing CMS spread options and digital options with smile", Wilmott, (May):63-69	True

Digital CMS Spread Options

Digital spread options are computed analytically and not by the call spread method.

References

Berrahoui,M. (2004) "Pricing CMS spread options and digital options with smile", Wilmott, (May):63-69

Hagan, P. (2003), "Convexity conundrums: pricing CMS swaps, caps and floors", *Wilmott*, (Mar.): 38-44

Pelsser, A. (2000), "Efficient methods for valuing interest rate derivatives", Springer-Verlag

2.22 Fixed Range Accrual Swap (EXSP) – PricerSwapHagan

A range accrual swap is one where the coupon has the following form;

$$\boxed{F.\frac{n}{N}}$$

Where,

F is a fixed rate, e.g. 5%,

n is the number of days on which index lies within a predetermined range

N is the total number of days in an observation period

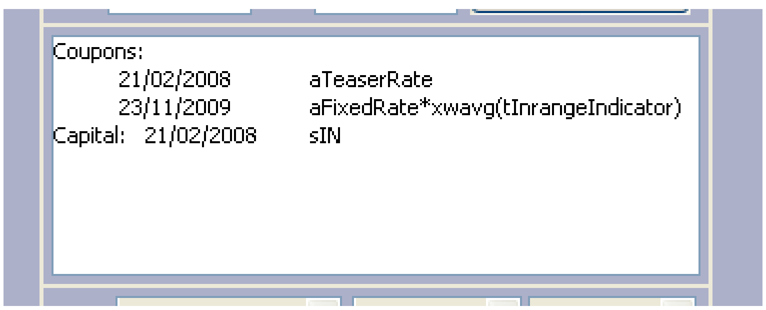
Valuation

The contract can be decomposed in to a sum of digital options on the index. The digitals are a little awkward because the fixing date and the payment date generally do not have the natural lag. To account for this we use the Hagan methodology already developed within PricerCapFloorHagan.

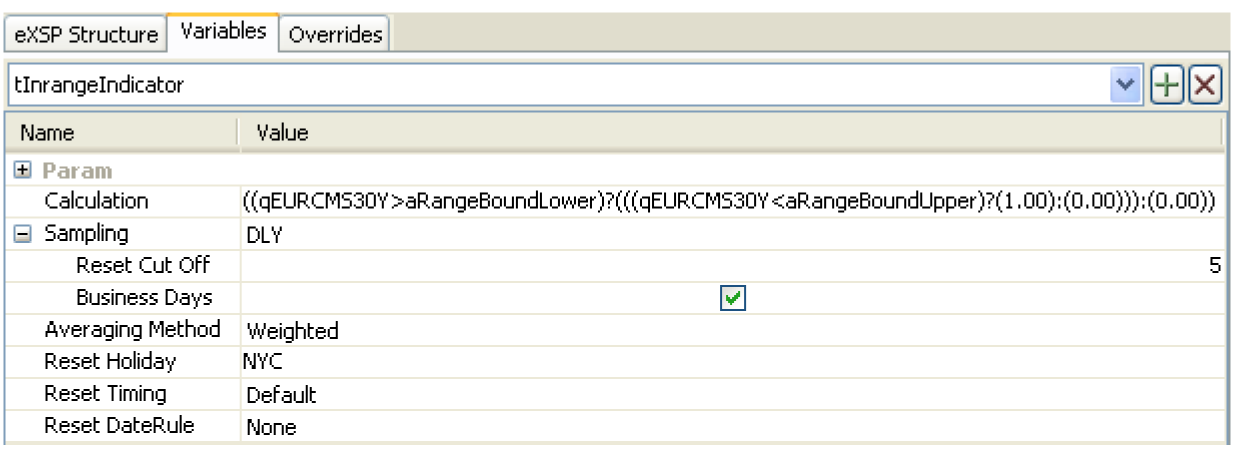
EXSP Terms and Conditions Setup

The coupons are setup as a weighted average of indicator functions. The indicator function is 1 if the index is in the range and 0 otherwise. The weight is generally, 1/N.

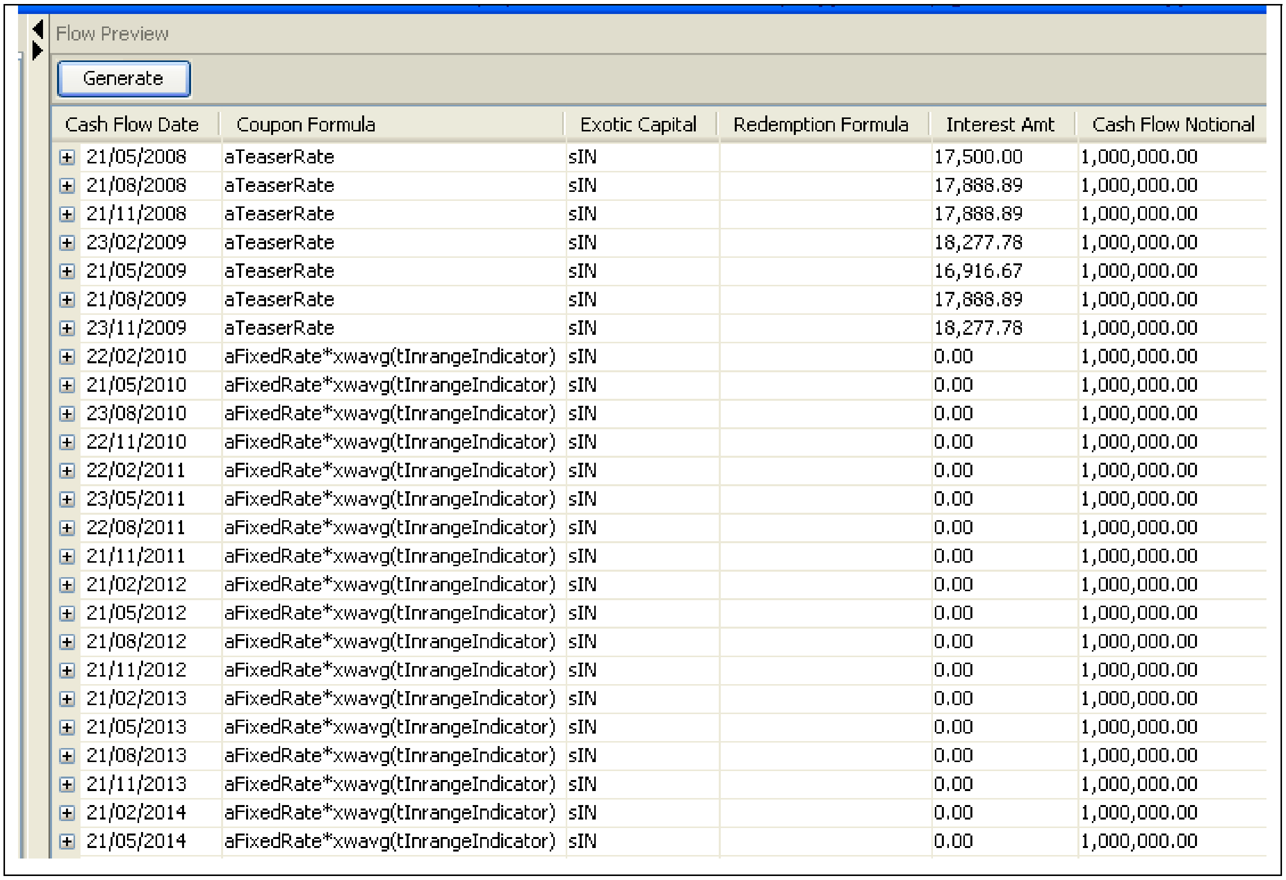
Coupon Formula



In Range Indicator



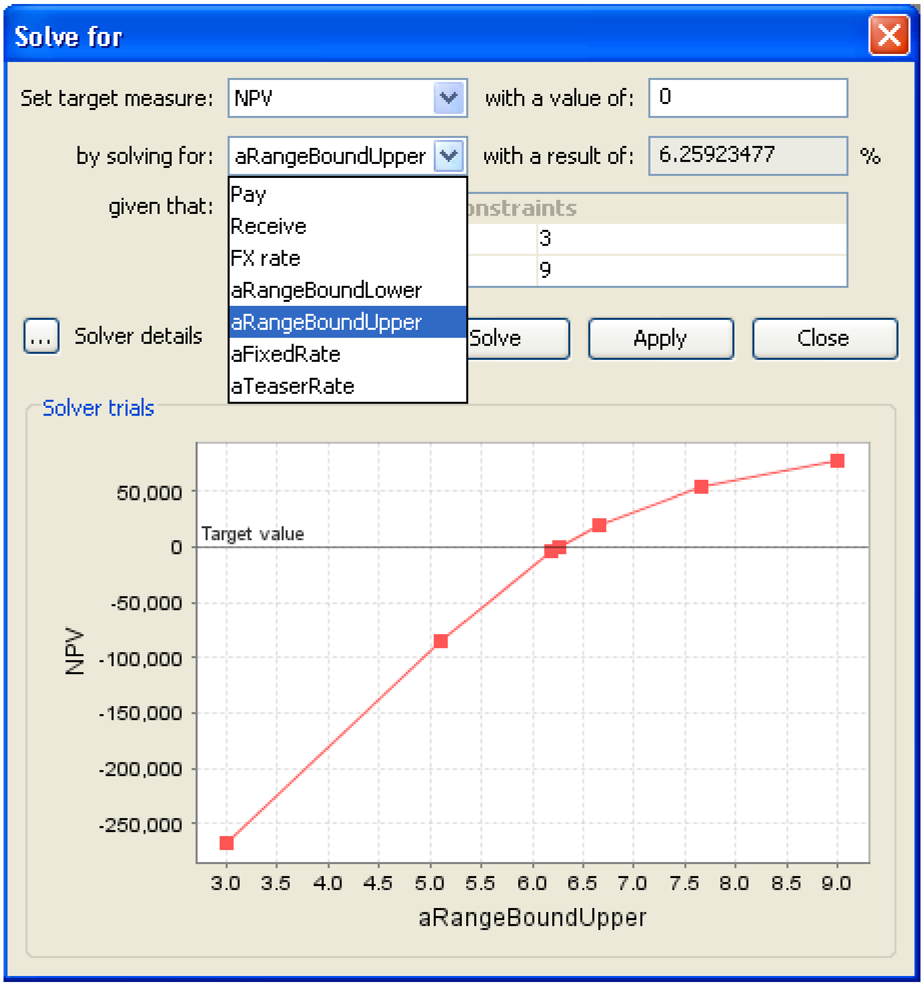
Teaser Rate



Cash Flow Date	Coupon Formula	Exotic Capital	Redemption Formula	Interest Amt	Cash Flow Notional
21/05/2008	aTeaserRate	sIN		17,500.00	1,000,000.00
21/08/2008	aTeaserRate	sIN		17,888.89	1,000,000.00
21/11/2008	aTeaserRate	sIN		17,888.89	1,000,000.00
23/02/2009	aTeaserRate	sIN		18,277.78	1,000,000.00
21/05/2009	aTeaserRate	sIN		16,916.67	1,000,000.00
21/08/2009	aTeaserRate	sIN		17,888.89	1,000,000.00
23/11/2009	aTeaserRate	sIN		18,277.78	1,000,000.00
22/02/2010	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/05/2010	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
23/08/2010	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
22/11/2010	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
22/02/2011	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
23/05/2011	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
22/08/2011	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/11/2011	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/02/2012	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/05/2012	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/08/2012	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/11/2012	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/02/2013	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/05/2013	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/08/2013	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/11/2013	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/02/2014	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00
21/05/2014	aFixedRate*xwavg(tInrangeIndicator)	sIN		0.00	1,000,000.00

2.22.1 Solver

In the formulae above we tend to use array variables, for teaser rate, fixed rate and the range lower and upper bounds. The benefit is that one can now solve for such array variables, for example



Pricing parameters and pricer measures are consistent with existing PricerSwapHagan and PricerCapFloorHagan pricers. In particular, STRIKE_SPREAD_EPSILON and STRIKE_SPREAD_DIRECTION are now available on PricerSwapHagan in the case of fixed range accrual swap.

References

Hagan, P. (2003), "Convexity conundrums: pricing CMS swaps, caps and floors", *Wilmott*, (Mar.): 38-44

2.23 Caption/Floortion – PricerCapFloortionLGMM

A caption is an option on cap, similarly floortion is an option on a floor.

PricerCapFloortionLGMM uses the LGM (Linear Gauss Markov) model to value European options on Caps/Floor.

Trade Setup

To setup a Caption\Floortion in Calypso go to Main Entry > Trade > Generic option, and select the CapFloor product type.

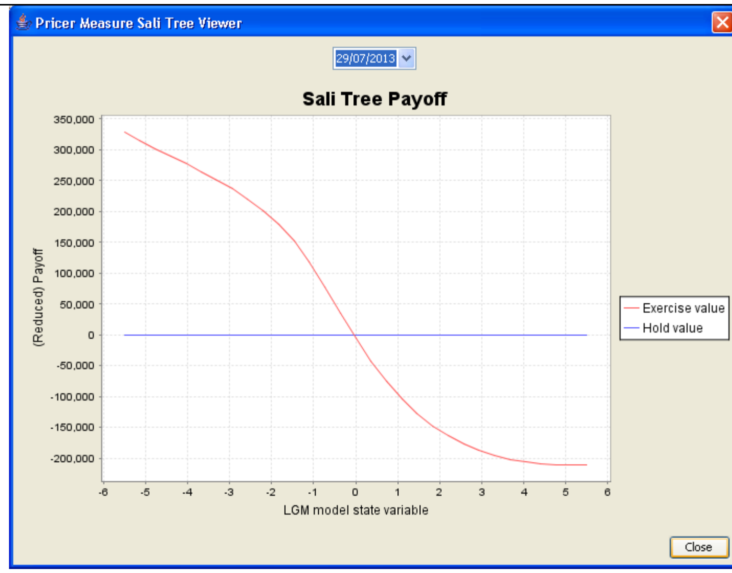
The strike of the option is entered either as an absolute amount or as a relative amount in the fee. Only European exercise is supported with PricerCapFloortionLGMM.

Pricing Parameters

Pricing Parameters	Type	Description	Typical Value
On-the-fly Parameters			
MODEL_VOL	Rate	The LGM model volatility parameter.	1%
MEAN_REV	Rate	The LGM model mean reversion parameter.	5%
IR_RATE	Rate	Transient to describe a constant interest rate zero curve, ACT/365 on continuously compounded basis.	4%
VOLATILITY	Rate	The Back volatility, allows one to enter a constant volatility to calibrate the LGM against, instead of the caplet volatility surface.	
Model Parameters			
LGMM_LATTICE_NODES	Integer	Number of nodes needed on the Sali-Tree.	15
LGMM_LATTICE_CUTOFF	Double	Number for standard deviations at which to clip the Sali-Tree.	5.5

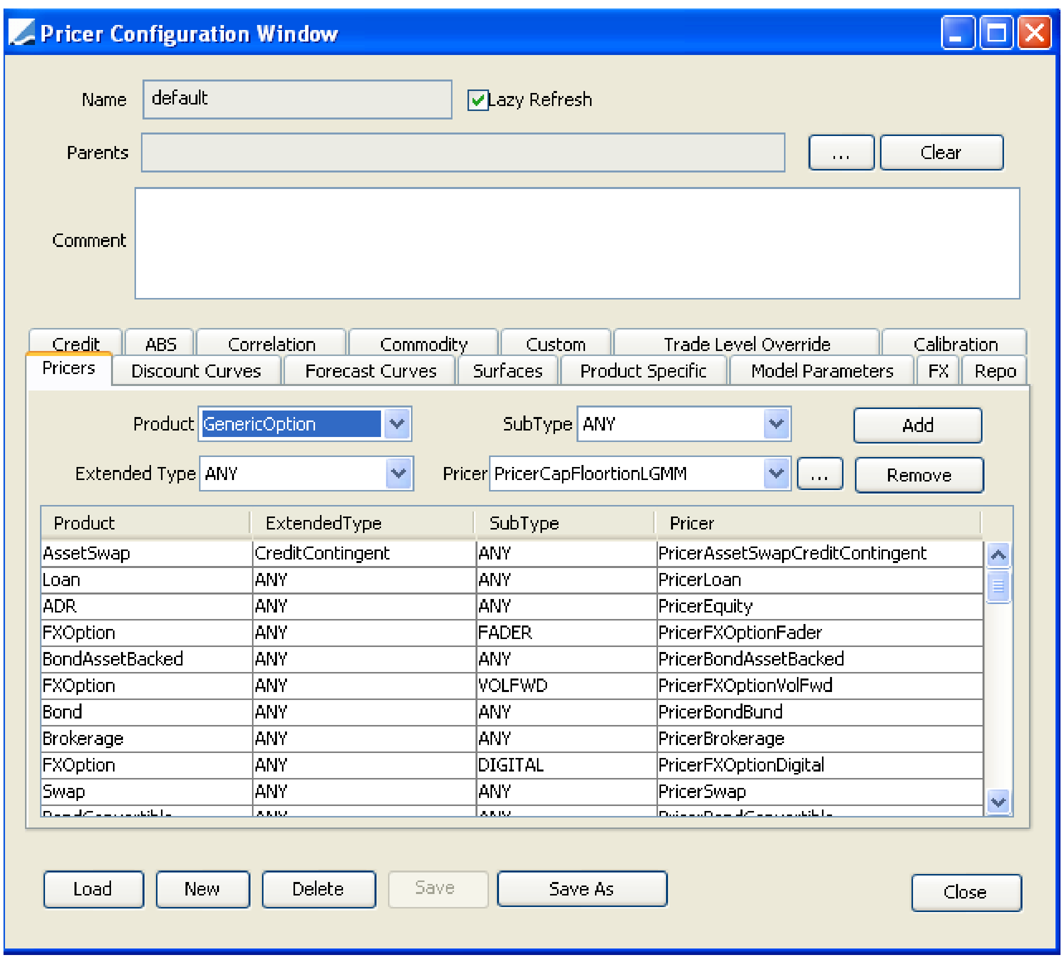
Pricer Measures

Pricer Measure	Description
ATM_STRIKE_AMOUNT	The strike for which the option is ATM
NPV_STRIKE	The NPV of the strike
NPV_UNDERLYING	The NPV of the underlying cap/floor
TIME_VALUE	Time value of the option
NPV_INTRINSIC	The intrinsic value of the option
SALI-TREE PAYOFFS	Shows the time slice of the Sali-Tree on the exercise date



Market Data Configuration

Discount curve, forecast curve and caplet volatility surface are configured in the same way as CapFloor pricing, except if one may need to configure for the product GenericOption instead of CapFloor product in the pricer config mappings.



Section 3. Structured Products

Product	Calypso Product	Description	Pricer
Structured Product	StructuredProduct	Generic composition of a new product as a function of other calypso products.	PricerStructuredProduct

3.1 PricerStructuredProduct

This pricer invokes the pricer(s) of the underlying products which build the structured product.

Section 4. Analytics

4.1 Black-Scholes Model

The standard Black-Scholes model is used for European style (single exercise) single asset options on equities, indices and futures.

The general form of the Black-Scholes model is used to value European style vanilla options. European options do not give the holder to exercise before maturity and therefore have an analytic solution. For calls and puts, we derive the following equations (see below):

$$C = e^{-r_p t_p}(FN(d_1) - KN(d_2))$$

$$P = e^{-r_p t_p}(KN(-d_2) - FN(-d_1)),$$

where $F = \frac{Se^{-qt_e}}{e^{-r_g t_e}} = Se^{(r_g - q)t_e}$, $d_1 = \frac{\ln\left[\frac{F}{K}\right]}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$ and $d_2 = d_1 - \sigma\sqrt{t_e}$.

This particular variation on the Black-Scholes model accounts for the timing of cashflows encountered in real-world transactions. The model does this by using different time periods for each type of cashflow. The variables are described as follows:

- S is the spot price of the underlying security.
- K is the strike price
- r_p is the continuously compounded risk-free rate with base act/365
- q is the continuously compounded dividend yield with base act/365
- r_g is the continuously compounded growth rate with base act/365

  Typically, periodically compounded rates which can be changed to continuous rates using the following equation:

  $$e^r = \left(1 + \frac{r}{n}\right)^{\frac{1}{n}}$$

  with n being the periodicity.
- σ is the volatility of returns of the underlying security
- t_e is the time period from the valuation date to the option's expiration date, i.e. the time for which the option is traded
- t_p is the time to payment (i.e. from the valuation date to the settlement date (usually two days after expiration)
- N(x) is the cumulative standard normal distribution function

Also, it is important to note that the value from the Option Pricing model may need adjustments to obtain the desired value (e.g. FX, Libor, and Swaptions).

All pricer measures (npv and Greeks) as defined below can be expressed also as of spot date by dividing the pricer measures as of value date by the discount factor between value date and spot date.

Implementation of the European Option Pricing Model

The general equation can be utilized to price various types of instruments. The parameter values are shown in the table below for each instrument. The correct pricing equation can be found by setting the parameters as shown.

Underlying Security	r_p equals	q equals	r_g equals
Stocks w/o dividends	risk-free rate	0	risk-free rate
Stocks w/ dividends	risk-free rate	dividend	risk-free rate
FX	quoting currency's interest rate	base currency's interest rate	quoting currency's interest rate
Futures	risk-free rate	0	0
Libor	risk-free rate	0	0
Swap	risk-free rate	0	0

Black Scholes Model 4-1.

4.1.1 Call Options

NPV

$$C = e^{-r_p t_p}(FN(d_1) - KN(d_2))$$

where

$$F = Se^{(r_g - q)t_e}$$

$$d_1 = \frac{\ln\left[\frac{F}{K}\right]}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$d_2 = d_1 - \sigma\sqrt{t_e} \qquad (1)$$

$$d_2^2 = d_1^2 - 2\sigma d_1\sqrt{t_e} + \sigma^2 t_e$$

$$d_2^2 = d_1^2 - 2\sigma\frac{\ln\left[\frac{F}{K}\right] + \sigma^2 t_e}{\sigma\sqrt{t_e}}$$

$$d_2^2 = d_1^2 - 2\ln(\frac{F}{K}) \qquad (2)$$

$$n(d_2)=\frac{e^{\frac{-d_2^2}{2}}}{\sqrt{2\pi}}$$

$$=\frac{e^{\frac{-d_1^2}{2}+\ln(\frac{F}{K})}}{\sqrt{2\pi}}$$

$$=\frac{Fe^{\frac{-d_1^2}{2}}}{K\sqrt{2\pi}}$$

$$=n(d_1)\frac{F}{K} \qquad (3)$$

Delta

$$\Delta_c=\frac{\partial C}{\partial S}=e^{-r_pt_p}\left(e^{(r_g-q)t_e}(Sn(d_1)\frac{\partial d_1}{\partial S}+N(d_1))-Kn(d_2)\frac{\partial d_2}{\partial S}\right)$$

Substituting equation (3) gives :

$$\Delta_c=e^{-r_pt_p+(r_g-q)t_e}N(d_1)+e^{-r_pt_p+(r_g-q)t_e}Sn(d_1)\frac{\partial d_1}{\partial S}-e^{-r_pt_p}Kn(d_1)\frac{\partial d_1}{\partial S}\frac{F}{K}$$

$$\Delta_c=e^{-r_pt_p+(r_g-q)t_e}N(d_1)+e^{-r_pt_p+(r_g-q)t_e}Sn(d_1)\frac{\partial d_1}{\partial S}-Se^{-r_pt_p+(r_g-q)t_e}n(d_1)\frac{\partial d_1}{\partial S}$$

$$\frac{\partial d_1}{\partial S}=\frac{\partial}{\partial S}\frac{(\ln Se^{-qt_e}-\ln Ke^{-rt_e})}{\sigma\sqrt{t_e}}=\frac{1}{S\sigma\sqrt{t_e}}$$

$$\Delta_c=e^{-r_pt_p+(r_g-q)t_e}N(d_1)$$

Delta Premium

$$\Delta_{c,premium}=e^{-r_pt_p+(r_g-q)t_e}N(d_1)*100-e^{-r_pt_p}(FN(d_1)-KN(d_2))$$

Delta Forward

$$\Delta_{c,forward}=\frac{\partial C}{\partial F}=e^{-r_pt_p}\left((Fn(d_1)\frac{\partial d_1}{\partial F}+N(d_1))-Kn(d_2)\frac{\partial d_2}{\partial F}\right)$$

Substituting equation (3) gives :

$$\Delta_c=e^{-r_pt_p}N(d_1)+e^{-r_pt_p}Fn(d_1)\frac{\partial d_1}{\partial F}-e^{-r_pt_p}Kn(d_1)\frac{\partial d_1}{\partial F}\frac{F}{K}$$

$$\Delta_c=e^{-r_pt_p}N(d_1)+e^{-r_pt_p}Fn(d_1)\frac{\partial d_1}{\partial F}-Fe^{-r_pt_p}n(d_1)\frac{\partial d_1}{\partial F}$$

$$\Delta_c=e^{-r_pt_p}N(d_1)$$

Gamma

$$\Gamma_c = \frac{\partial^2 C}{\partial S^2} = \frac{\partial}{\partial S}\Delta_c$$

$$\Gamma_c = e^{-r_p t_p + (r_g - q)t_e} n(d_1)\frac{\partial d_1}{\partial S}$$

$$\frac{\partial d_1}{\partial S} = \frac{1}{S\sigma\sqrt{t_e}}$$

Combining the above equations gives :

$$\Gamma_c = \frac{1}{S\sigma\sqrt{t_e}}\left(e^{-r_p t_p + (r_g - q)t_e} n(d_1)\right)$$

Vega

$$\nu_c = \frac{\partial C}{\partial \sigma} = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial \sigma} - e^{-r_p t_p} Kn(d_2)\frac{\partial d_2}{\partial \sigma}$$

$$\nu_c = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial \sigma} - Ke^{-r_p t_p} n(d_1)\frac{\partial d_2}{\partial \sigma}\frac{S}{K}e^{r_g t_e - q t_e}$$

$$\nu_c = Se^{-r_p t_p + (r_g - q)t_e} n(d_1)(\frac{\partial d_1}{\partial \sigma} - \frac{\partial d_2}{\partial \sigma})$$

where

$$\frac{\partial d_1}{\partial \sigma} = -\frac{\ln\left[\frac{F}{K}\right]}{\sigma^2\sqrt{t_e}} + \frac{\sqrt{t_e}}{2}$$

and

$$\frac{\partial d_2}{\partial \sigma} = -\frac{\ln\left[\frac{F}{K}\right]}{\sigma^2\sqrt{t_e}} - \frac{\sqrt{t_e}}{2}$$

$$\nu_c = e^{-r_p t_p + (r_g - q)t_e} S\sqrt{t_e}\, n(d_1)$$

rho$_{r_g}$

$$\rho_{c,g} = \frac{\partial C}{\partial r_g} = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial r_g} + St_e e^{-r_p t_p + (r_g - q)t_e} N(d_1) - Ke^{-r_p t_p} n(d_2)\frac{\partial d_2}{\partial r_g}$$

$$d_1 = \frac{\ln Se^{-qt_e} - \ln Ke^{-r_g t_e}}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$\frac{\partial d_1}{\partial r_g} = \frac{\partial d_2}{\partial r_g} = \frac{t_e}{\sigma\sqrt{t_e}}$$

Substituting equation (3) gives :

$$\rho_{c,g} = \frac{t_e e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)}{\sigma\sqrt{t_e}} + St_e e^{-r_p t_p + (r_g - q)t_e} N(d_1) - Ke^{-r_p t_p} n(d_2)\frac{\partial d_2}{\partial r}$$

$$\rho_{c,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(d_1)}{\sigma\sqrt{t_1}} + t_e N(d_1)) - Ke^{-r_p t_p} n(d_2)\frac{\partial d_2}{\partial r}$$

$$\rho_{c,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(d_1)}{\sigma\sqrt{t_1}} + t_e N(d_1)) - e^{-r_p t_p} Kn(d_1)\frac{\partial d_1}{\partial r}\frac{S}{K}e^{(r_g - q)t_e})$$

$$\rho_{c,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(d_1)}{\sigma\sqrt{t_e}} + t_e N(d_1)) - Se^{-r_p t_p + (r_g - q)t_e} n(d_1)\frac{\partial d_1}{\partial r}$$

$$\rho_{c,g} = e^{-r_p t_p + (r_g - q)t_e} St_e N(d_1)$$

rho$_{r_p}$

$$\rho_{c,p} = \frac{\partial C}{\partial r_p} = e^{-r_p t_p} Kt_p N(d_2) - t_p e^{-r_p t_p + (r_g - q)t_e} SN(d_1)$$

rho$_q$

$$\rho_{c,q} = \frac{\partial C}{\partial q} = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial q} - e^{-r_p t_p + (r_g - q)t_e} St_e N(d_1) - Ke^{-r_p t_p} n(d_2)\frac{\partial d_2}{\partial q}$$

$$d_1 = \frac{\ln Se^{-qt_e} - \ln Ke^{-rt_e}}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$\frac{\partial d_1}{\partial q} = \frac{\partial d_2}{\partial q} = \frac{-t_e}{\sigma\sqrt{t_e}}$$

Substituting equation (3) gives :

$$\rho_{c,q} = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial q} - Ke^{-r_p t_p} n(d_1)\frac{S}{K}e^{(r_g - q)t_e}\frac{\partial d_1}{\partial q} - St_e e^{-r_p t_p + (r_g - q)t_e} N(d_1)$$

$$\rho_{c,q} = -St_e e^{-r_p t_p + (r_g - q)t_e} N(d_1)$$

Theta

Defining $t_1 = T_1 - \tau$, $t_2 = T_2 - \tau$, and $t_3 = T_3 - \tau$:

$$\Theta_c = -\frac{\partial C}{\partial \tau} = e^{-r_p t_p + (r_g - q)t_e}(q + r_p - r_g)SN(d_1) - e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial \tau} + Ke^{-r_p t_p} n(d_2)\frac{\partial d_2}{\partial \tau} - e^{-r_p t_p} Kr_p N(d_2)$$

Substituting equation (3) gives:

$$\Theta_c = e^{-r_p t_p + (r_g - q)t_e}(q + r_p - r_g)SN(d_1) - e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\frac{\partial d_1}{\partial \tau} + Ke^{-r_p t_p + (r_g - q)t_e} n(d_1)\frac{S}{K}\frac{\partial d_2}{\partial \tau} - e^{-r_p t_p} Kr_p N(d_2)$$

$$\Theta_c = e^{-r_p t_p + (r_g - q)t_e}(q + r_p - r_g)SN(d_1) + e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\left(\frac{\partial d_2}{\partial \tau} - \frac{\partial d_1}{\partial \tau}\right) - e^{-r_p t_p} Kr_p N(d_2)$$

Given:

$$\frac{\partial d_2}{\partial \tau} - \frac{\partial d_1}{\partial \tau} = \frac{\partial}{\partial \tau}\sigma\sqrt{t_e} = -\frac{\sigma}{2\sqrt{t_e}}$$

$$\Theta_c = e^{-r_p t_p + (r_g - q)t_e}(q + r_p - r_g)SN(d_1) - \frac{e^{-r_p t_p + (r_g - q)t_e} S\sigma n(d_1)}{2\sqrt{t_e}} - e^{-r_p t_p} Kr_p N(d_2)$$

4.1.2 Put Options

NPV

$$P = Ke^{-r_p t_p} N(-d_2) - Se^{-r_p t_p + (r_g - q)t_e} N(-d_1)$$

$$d_1 = \frac{\ln\left[\dfrac{Se^{-qt_e}}{Ke^{-r_g t_e}}\right]}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$d_2 = d_1 - \sigma\sqrt{t_e} \qquad (1)$$

$$d_2^2 = d_1^2 - 2\sigma d_1\sqrt{t_e} + \sigma^2 t_e$$

$$d_2^2 = d_1^2 - 2\sigma\frac{\ln\left[\dfrac{Se^{-qt_e}}{Ke^{-r_g t_e}}\right] + \sigma^2 t_e}{\sigma\sqrt{t_e}}$$

$$d_2^2 = d_1^2 - 2\ln\left(\frac{S}{K}e^{r_g t_e - qt_e}\right) \qquad (2)$$

$$\begin{aligned} n(d_2) &= \frac{e^{\frac{-d_2^2}{2}}}{\sqrt{2\pi}} \\ &= \frac{e^{\frac{-d_1^2}{2} + \ln\left(\frac{S}{K}e^{r_g t_e - qt_e}\right)}}{\sqrt{2\pi}} \\ &= \frac{Se^{\frac{-d_1^2}{2}}}{K\sqrt{2\pi}} e^{r_g t_e - qt_e} \\ &= n(d_1)\frac{S}{K}e^{r_g t_e - qt_e} \qquad (3) \end{aligned}$$

Delta

$$\Delta_p = \frac{\partial P}{\partial S} = e^{-r_p t_p}\left(-e^{(r_g-q)t_e}(Sn(-d_1)\frac{\partial - d_1}{\partial S} + N(-d_1)) + Kn(-d_2)\frac{\partial - d_2}{\partial S}\right)$$

Substituting equation (3) in the equation above gives :

$$\Delta_p = -e^{-r_p t_p + (r_g-q)t_e}N(-d_1) - e^{-r_p t_p + (r_g-q)t_e}Sn(-d_1)\frac{\partial - d_1}{\partial S} + e^{-r_p t_p}Kn(-d_2)\frac{\partial - d_2}{\partial S}\frac{F}{K}$$

$$\Delta_p = -e^{-r_p t_p + (r_g-q)t_e}N(-d_1) - e^{-r_p t_p + (r_g-q)t_e}Sn(-d_1)\frac{\partial - d_1}{\partial S} - Se^{-r_p t_p + (r_g-q)t_e}n(-d_1)\frac{\partial - d_1}{\partial S}$$

$$\frac{\partial - d_1}{\partial S} = \frac{\partial - d_2}{\partial S} = \frac{\partial}{\partial S}\frac{(\ln Se^{-qt_e} - \ln Ke^{-rt_e})}{\sigma\sqrt{t_e}} = -\frac{1}{S\sigma\sqrt{t_e}}$$

$$\Delta_p = -e^{-r_p t_p + (r_g-q)t_e}N(-d_1)$$

Delta Premium

$$\Delta_{p,premium} = -e^{-r_p t_p + (r_g-q)t_e}N(-d_1)*100 + Ke^{-r_p t_p}N(-d_2) - Se^{-r_p t_p + (r_g-q)t_e}N(-d_1)$$

Delta Forward

$$\Delta_{p,forward} = \frac{\partial P}{\partial F} = e^{-r_p t_p}\left((-Fn(-d_1)\frac{\partial - d_1}{\partial F} - N(-d_1)) + Kn(-d_2)\frac{\partial - d_2}{\partial F}\right)$$

Substituting equation (3) gives :

$$\Delta_{p,forward} = -e^{-r_p t_p}N(-d_1) - e^{-r_p t_p}Fn(-d_1)\frac{\partial - d_1}{\partial F} + e^{-r_p t_p}Kn(-d_1)\frac{\partial - d_1}{\partial F}\frac{F}{K}$$

$$\Delta_{p,forward} = -e^{-r_p t_p}N(-d_1) - e^{-r_p t_p}Fn(-d_1)\frac{\partial - d_1}{\partial F} + Fe^{-r_p t_p}n(-d_1)\frac{\partial - d_1}{\partial F}$$

$$\Delta_{p,forward} = -e^{-r_p t_p}N(-d_1)$$

Gamma

$$\Gamma_p = \frac{\partial^2 P}{\partial S^2} = \frac{\partial}{\partial S} - e^{-r_p t_p + (r_g-q)t_e}N(-d_1)$$

$$\frac{\partial - d_1}{\partial S} = \frac{\partial}{\partial S}\frac{-(\ln Se^{-qt_e} - \ln Ke^{-rt_e})}{\sigma\sqrt{t_e}} = -\frac{1}{S\sigma\sqrt{t_e}}$$

Combining the above equations gives :

$$\Gamma_p = \frac{e^{-r_p t_p + (r_g-q)t_e}n(d_1)}{S\sigma\sqrt{t_e}}$$

Vega

$$\nu_p = \frac{\partial P}{\partial \sigma} = -e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\frac{\partial - d_1}{\partial \sigma} + e^{-r_p t_p} Kn(-d_2)\frac{\partial - d_2}{\partial \sigma}$$

$$\nu_p = -e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\frac{\partial - d_1}{\partial \sigma} + Ke^{-r_p t_p} n(-d_1)\frac{\partial - d_1}{\partial \sigma}\frac{S}{K}e^{(r_g - q)t_e}$$

$$\nu_p = -e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\left(\frac{\partial - d_1}{\partial \sigma} - \frac{\partial - d_2}{\partial \sigma}\right)$$

The difference in the partial derivatives with respect to σ can be derived from equation (1) and is found to be - $\sqrt{t_1}$.
Therefore, ν can be written as follows:

$$\nu_p = e^{-r_p t_p + (r_g - q)t_e} Sn(d_1)\sqrt{t_e}$$

rho$_{r_p}$

$$\rho_{p,p} = \frac{\partial P}{\partial q} = -t_p e^{-r_p t_p} KN(-d_2) + t_p e^{-r_p t_p + (r_g - q)t_e} SN(-d_1)$$

rho$_{r_g}$

$$\rho_{p,g} = \frac{\partial P}{\partial r_g} = -e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\frac{\partial - d_1}{\partial r_g} - St_e e^{-r_p t_p + (r_g - q)t_e} N(-d_1) + Ke^{-r_p t_p} n(-d_2)\frac{\partial - d_2}{\partial r_g}$$

$$d_1 = \frac{\ln Se^{-qt_e} - \ln Ke^{-r_g t_e}}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$\frac{\partial - d_1}{\partial r} = \frac{\partial - d_2}{\partial r} = -\frac{t_e}{\sigma\sqrt{t_e}}$$

$$\rho_{p,g} = \frac{t_e e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)}{\sigma\sqrt{t_e}} - St_e e^{-r_p t_p + (r_g - q)t_e} N(-d_1) + Ke^{-r_p t_p} n(-d_2)\frac{\partial - d_2}{\partial r_g}$$

$$\rho_{p,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(-d_1)}{\sigma\sqrt{t_e}} - t_e N(d_1)) + Ke^{-r_p t_p} n(-d_2)\frac{\partial - d_2}{\partial r_g}$$

Substituting equation (3) gives :

$$\rho_{p,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(-d_1)}{\sigma\sqrt{t_1}} - t_e N(-d_1)) + e^{-r_p t_p} Kn(-d_1)\frac{\partial - d_1}{\partial r_g}\frac{S}{K} e^{(r_g - q)t_e})$$

$$\rho_{p,g} = e^{-r_p t_p + (r_g - q)t_e} S(\frac{t_e n(-d_1)}{\sigma\sqrt{t_e}} - t_e N(-d_1)) + Se^{-r_p t_p + (r_g - q)t_e} n(-d_1)\frac{\partial d_1}{\partial r_g}$$

$$\rho_{p,g} = -e^{-r_p t_p + (r_g - q)t_e} St_e N(-d_1)$$

rho$_q$

The sensitivity of the option price to dividends

$$\rho_{2,p} = \frac{\partial P}{\partial q} = -e^{-r_p t_p + (r_g - q)t_e} (Sn(-d_1)\frac{\partial - d_1}{\partial q} - St_e N(-d_1)) + Ke^{-r_p t_p} n(-d_2)\frac{\partial - d_2}{\partial q}$$

$$d_1 = \frac{\ln Se^{-qt_e} - \ln Ke^{-rt_e}}{\sigma\sqrt{t_e}} + \frac{\sigma\sqrt{t_e}}{2}$$

$$\frac{\partial d_1}{\partial q} = \frac{\partial d_2}{\partial q} = \frac{-t_e}{\sigma\sqrt{t_e}}$$

Substituting equation (3) gives :

$$\rho_{2,p} = -e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\frac{\partial - d_1}{\partial q} + Ke^{-r_p t_p} n(-d_1)\frac{S}{K} e^{(r_g - q)t_e}\frac{\partial - d_1}{\partial q} + St_e e^{-r_p t_p + (r_g - q)t_e} N(-d_1)$$

$$\rho_{2,c} = St_e e^{-r_p t_p + (r_g - q)t_e} N(-d_1)$$

Theta

$$\Theta_p = -\frac{\partial P}{\partial \tau} = e^{-r_p t_p + (r_g - q)t_e}((-q - r_p + r_g)SN(-d_1) + Sn(-d_1)\frac{\partial - d_1}{\partial \tau}) - e^{-r_p t_p} Kn(-d_2)\frac{\partial - d_2}{\partial \tau} + e^{-r_p t_p} Kr_p N(-d_2)$$

Substituting equation (3) gives :

$$\Theta_p = e^{-r_p t_p + (r_g - q)t_e}((-q - r_p + r_g)SN(-d_1) + Sn(-d_1)\frac{\partial - d_1}{\partial \tau}) - e^{-r_p t_p} Kn(-d_1)\frac{S}{K}e^{(r_g - q)t_e}\frac{\partial - d_2}{\partial \tau} + e^{-r_p t_p} Kr_p N(-d_2)$$

$$\Theta_p = e^{-r_p t_p + (r_g - q)t_e}(-q - r_p + r_g)SN(-d_1) + e^{-r_p t_p + (r_g - q)t_e} Sn(-d_1)\left(\frac{\partial - d_1}{\partial \tau} - \frac{\partial - d_2}{\partial \tau}\right) + e^{-r_p t_p} Kr_p N(-d_2)$$

Given :

$$\frac{\partial - d_1}{\partial \tau} - \frac{\partial - d_2}{\partial \tau} = \frac{\partial}{\partial \tau} - \sigma\sqrt{t_e} = -\frac{\sigma}{2\sqrt{t_e}}$$

$$\Theta_p = e^{-r_p t_p + (r_g - q)t_e}((-q - r_p + r_g)SN(-d_1) - \frac{e^{-r_p t_p + (r_g - q)t_e} S\sigma n(-d_1)}{2\sqrt{t_e}}) + e^{-r_p t_p} Kr_p N(-d_2)$$

4.2 Black Model

This is an adapted version of Black-Scholes used to price European style caps, floors and swaptions.

4.3 One Factor Interest Rate Model

Usually interest rates are modeled as jointly correlated random variables. The simplest approach is to consider the evolution of the interest rate term structure to be highly collinear and thus introduce one factor of randomness. This single factor of uncertainty is attributed to a fictitious short term interest rate from which the whole term structure of interest rates is subsequently derived. Calypso offers this class of models described below as “Vasicek-type One Factor Interest Rate models”.

Calypso’s implementation models the following stochastic differential equation:

$$df(r) = [\theta(t) - a(t)f(r)]dt + \sigma(t)dz$$

Variable	Description
$f(r)$	a function of the short-rate; $f(r) = r$ for normal process and $f(r) = \ln r$ for log-normal process. In Calypso, it is set in the parameter “IS_NORMAL_PROCESS”. True for normal. False for log-normal.
$\theta(t)$	mean reversion level of the rate; it is calibrated automatically in Calypso using the discount curve to ensure that the model is consistent with market interest rates.
$a(t)$	mean reversion; this can be time-dependent or constant. In the latter case, the pricing parameter CONSTANT_MEAN_REVERSION is used.
$\sigma(t)$	volatility of the rate; this can be time-dependent or constant. In the latter case, the pricing parameter CONSTANT_MEAN_REVERSION is used.
$df(r)$	the change of the function of the instantaneous short-term interest rate over a small interval.
dt	a small change in time.

dz	a Wiener process (the source of uncertainty).

Solving the following system of equations:

$p_u + p_m + p_d = 1$ with p_u, p_m, p_d as the 3 transition probabilities from node $N_{i,t}$ to node $N_{ijk,t+\Delta t}$

$$p_u x_u + p_m x_m + p_d x_d = E[x]$$

$$p_u {x_u}^2 + p_m {x_m}^2 + p_d {x_d}^2 = V[x] + E[x]^2$$

implies constructing a trinomial lattice. Calypso's unique implementation draws to optimize performance versus accuracy. The following points are worth to be highlighted:

- Separation of lattice geometry construction from calibration to initial term structure (computation of drifts); allows most of the code to be used for both normal, lognormal as well as mixed distributions;
- Non-equidistant lattice: Lattice time points (horizontal axis) are determined by relevant cashflow dates of trade underlying product(s); no caching of lattices; no big demand on RAM; for each trade, a "custom" lattice is generated for given interest rate, volatility (and mean reversion) term structures;
- In first time interval, lattice grows in an n-nomial transition to fan out to as many states (vertical dimension) as dictated by the number of (vertical) nodes (input parameter)
- Transition probabilities are computed such that the local mean and variance are preserved; at the outer edges, if the projected node indices fall beyond the lattice boundary (as defined by the number of standard deviations which is an input parameter) we switch from a trinomial to a binomial transition. In rare cases, we have to revert to a monomial transition;

Calypso's One-Factor Model actually comprises four different one-factor models. The type of model the user wants to use is controlled through the pricing parameters associated with the OneFactorModel pricers.

Model	*f(r)*	**Distribution**	**Volatility**	**Mean reversion**
Hull-White	*r*	Normal	Constant or Term structure	Constant or Term structure
Black-Karasinski	ln*(r)*	lognormal	Constant or Term structure	Constant or Term structure
Normal Ho-Lee	*r*	Normal	Constant or Term structure	0
Lognormal Hoo-Lee	ln*(r)*	lognormal	Constant or Term structure	0

4.4 Multi Factor Interest Rate Model

In contrast to the One Factor Models which assume all the interest rate movements are a function of the (unobservable) short rate, the Multi Factor Model assumes the future of the interest rate movements depend on several observable forward rates which are allowed to move with different random behavior. This lends the model to be calibrated more accurately to the market and is why this model is also known by the name Libor Market Model.

Calypso's implementation of the model uses Monte Carlo simulation to repeatedly walk down possible paths and then average out the total collection of values to determine an expected value of a given trade.

4.5 Linear Gauss Markov Model

The Linear Gauss Markov Model is really the Hull-White one factor model reset in the Heath-Jarrow-Morton framework for Bermudan swaptions. This alternative characterization greatly helps with calibration and trade valuation.

4.6 CEV model

The underlying asset is modeled as following a so-called constant elasticity of variance (CEV) process (as opposed to lognormal in Black's model), specifically the underlying S is modeled with the following S.D.E.

$$dS = \alpha S^{\beta} dW$$

There is a separate whitepaper available, "C.E.V. and displaced diffusion models", that describes the Calypso implementation in detail.

4.7 Stochastic Alpha Beta Rho (SABR) Model

The SABR model allows for stochastic volatility as part of the model. The reason for is to accommodate for the paradox of the smile/skew observed in the market data volatilities. The Black model assumes that the volatility is constant across strikes but this is different than what is actually observed in the market place. The SABR model predicts a smile across strikes that fits observed market data very well. Because of this, the SABR model can also be used for describing the smile in a parametric form.

Therefore in Calypso we have two uses of SABR. One implements the SABR pricing model where the inputs are the calibrated parameters of the model: alpha, beta, rho, nu. In the other case, these parameters define a nice smile in the surface given current market volatility quotes and therefore create a useful volatility surface generator which can deliver accurate volatilities by using this parametric form.

Index