Adenza

Calypso Interest Rate Derivatives Analytics

Version 17.0

February 2022 – 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.

- **For curve generation information, please refer to the *Calypso Yield Curve Generation Guide*.**
- **Detailed information on inflation analytics can be found in the *Calypso Inflation Derivatives Analytics Guide*.**
- **For information on pricers provided by the Calypso Analytics Library, please refer to the *Calypso Analytics Library Guide* (CALIB).**

Revision Date	Comments
February 2022	First edition for version 17.0.

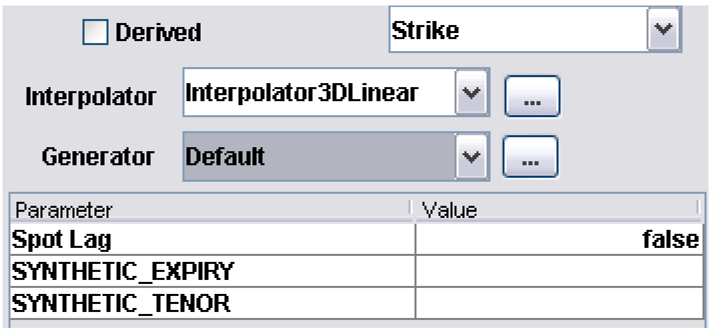
Contents

Section 1. Volatility Surfaces

1.1 Simple Generators

1.1.1 Default Generator

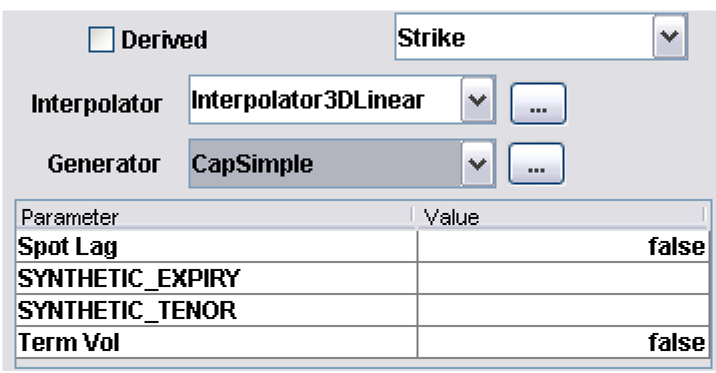
Default Generator – Can be used for caps, swaptions, bond options, etc. So-called simple surface where surface points are defined by the user, not using underlying instruments. If the spot lag parameter is set to true, the generated exercise dates are rolled using the conventions of the definition screen.



Note that SYNTHETIC_EXPIRY and SYNTHETIC_TENOR are not currently used.

1.1.2 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.

1.1.3 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

1.1.4 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

1.1.5 SwaptionSimple Generator

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

1.1.6 CMSBasisAdjSimple Generator

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

1.1.7 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.

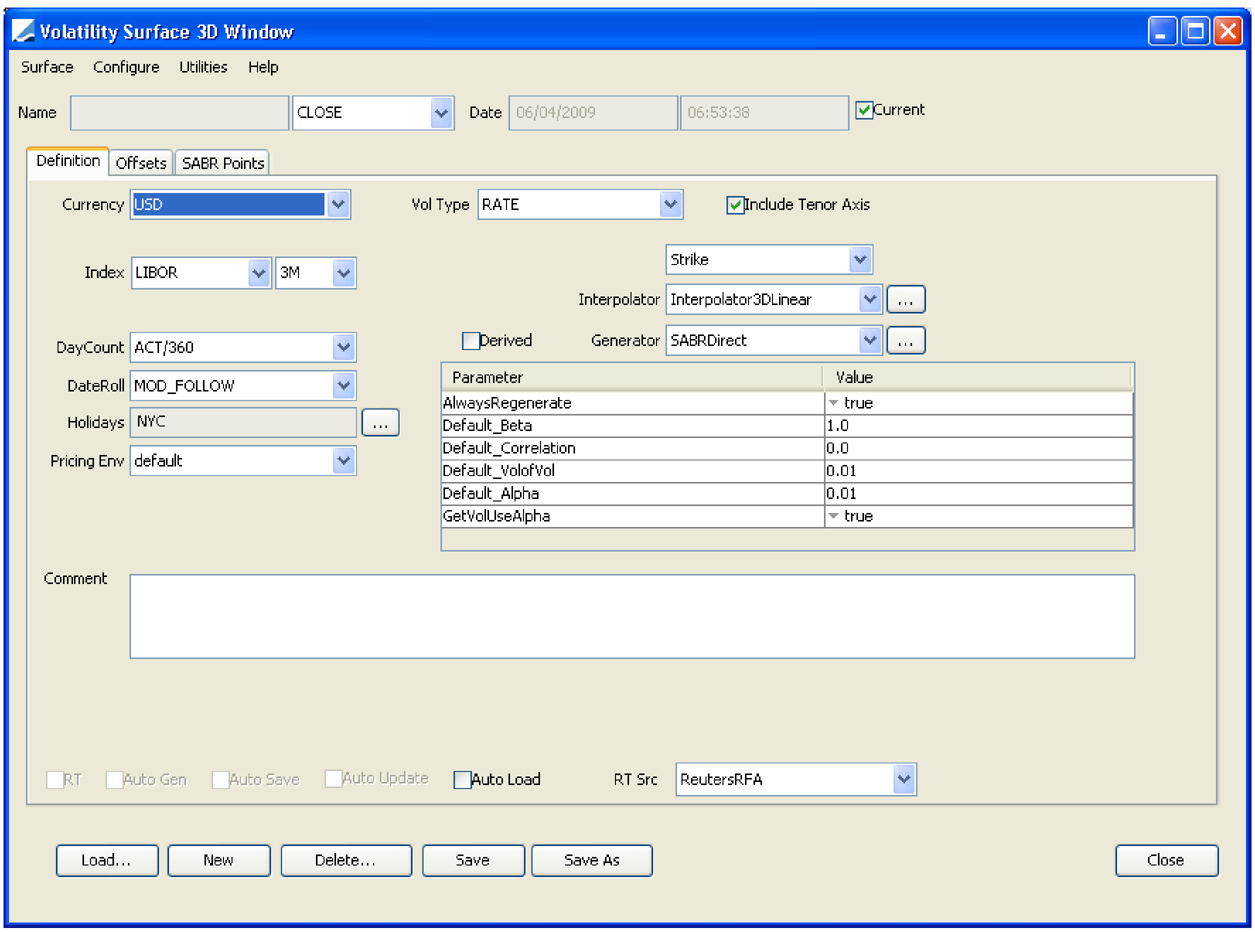
1.1.8 SABRDirect Generator

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;

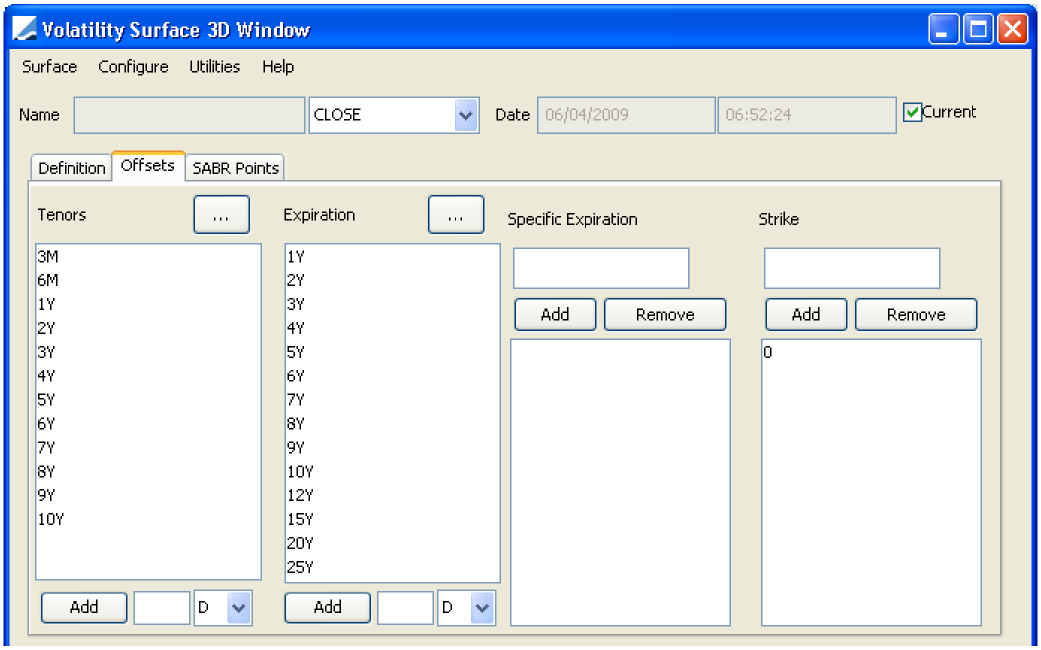
1. Select Include Tenor Axis
2. Select SABRDirect generator
3. Set the generator parameter GetVolUseAlpha = true, (this is important!)



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

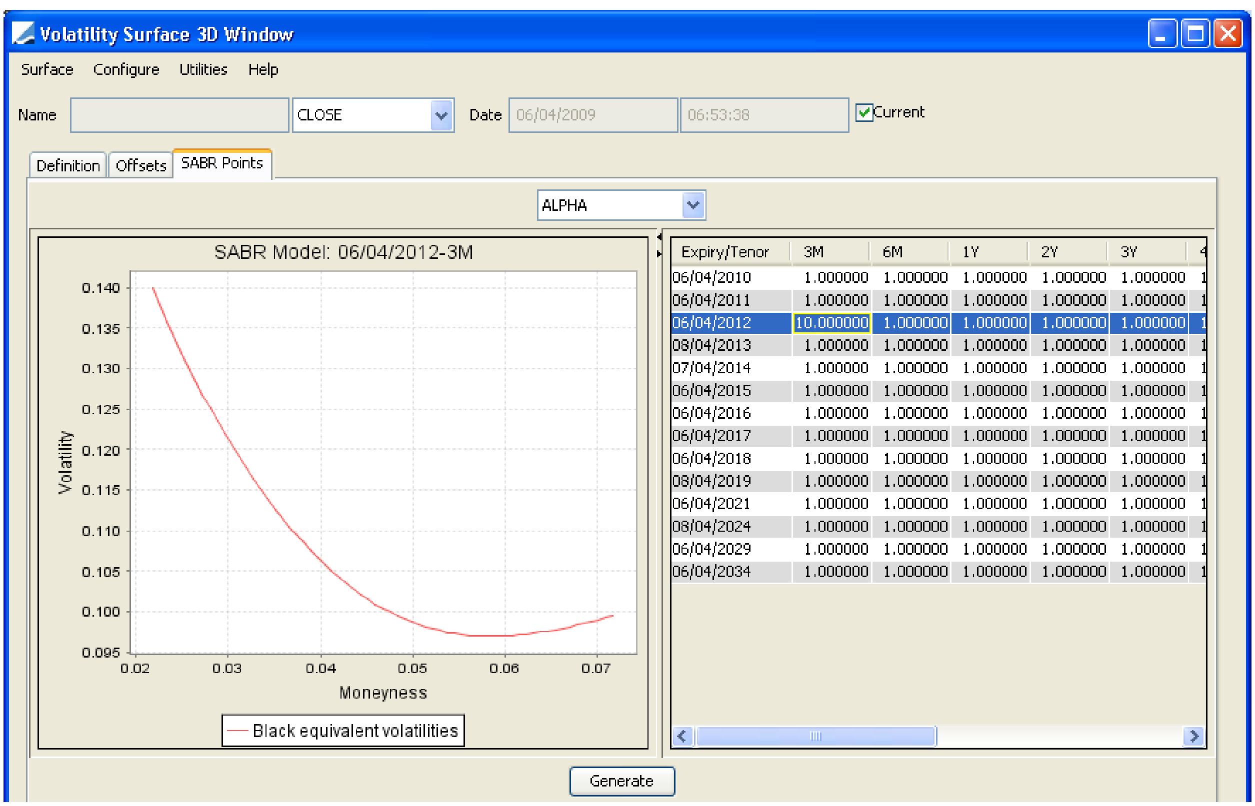
Offsets

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 volatilities.

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.

1.1.9 LGMMeanRev Generator

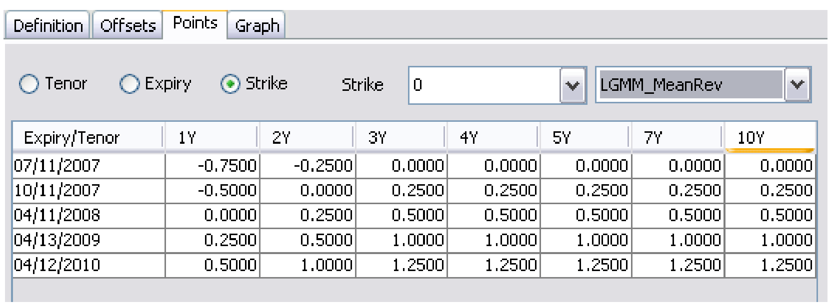
The Linear Gauss Markov Model is really the Hull-White one factor model reset in the Heath-Jarrow-Morton framework for **Bermudan Swaptions**. This alternative characterization greatly helps with calibration and trade valuation.

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

1.2 Derived Generators

1.2.1 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 strikes of underlying caps must be defined as absolute.

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:

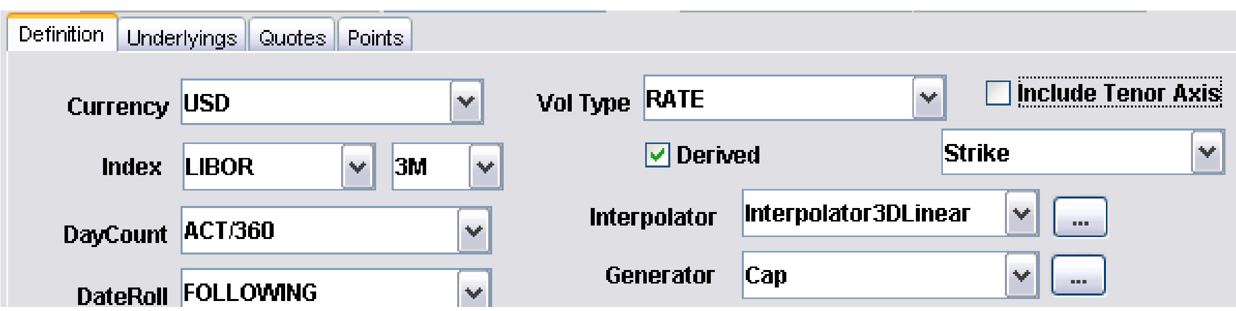
1. A preliminary volatility surface is built for the expiry/tenor/strike points using the surface underlying instruments and their term quotes.
2. Each cap/floor is decomposed into its equivalent caplets/floorlets. Synthetic caps/floors matching the end date of each caplet/floorlet are created and priced 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.

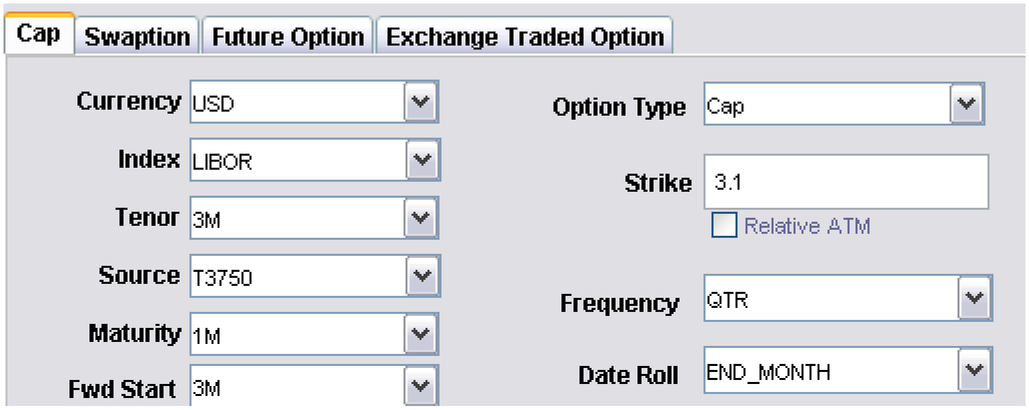
3. Now we have to solve for the volatility of the last caplet of each synthetic cap such that we re-price it exactly. In the example above, the (synthetic) cap 3M volatility is the same as the volatility of the first 3M caplet. Taking the 6M (synthetic) cap target present value, we find the implied volatility 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

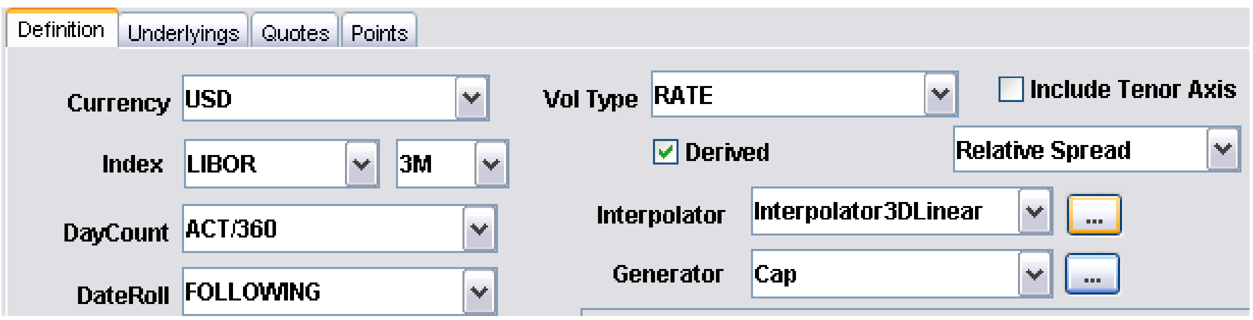


The underlying instruments must be specified using an absolute strike as shown below.

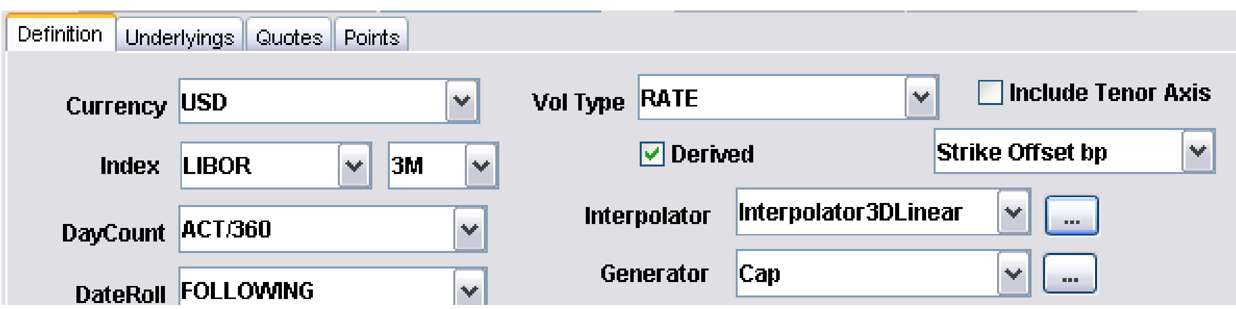


Then, you can enter the quotes and generate the points using the Quotes and Points panels.

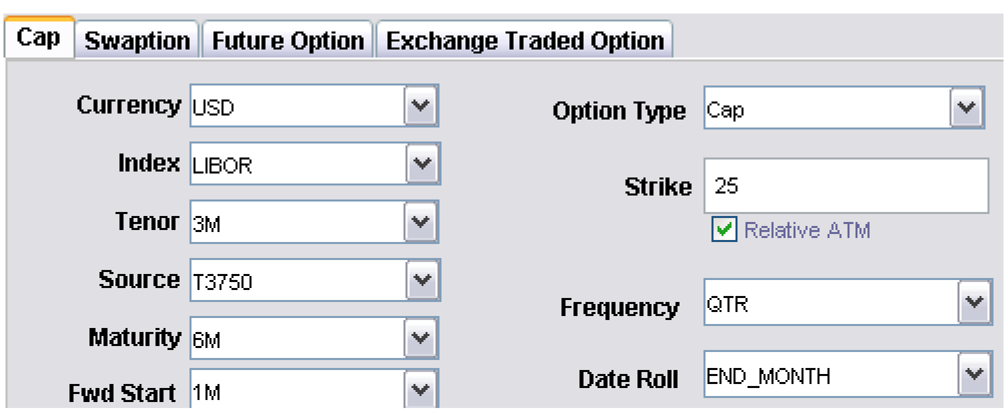
Relative Spread Definition



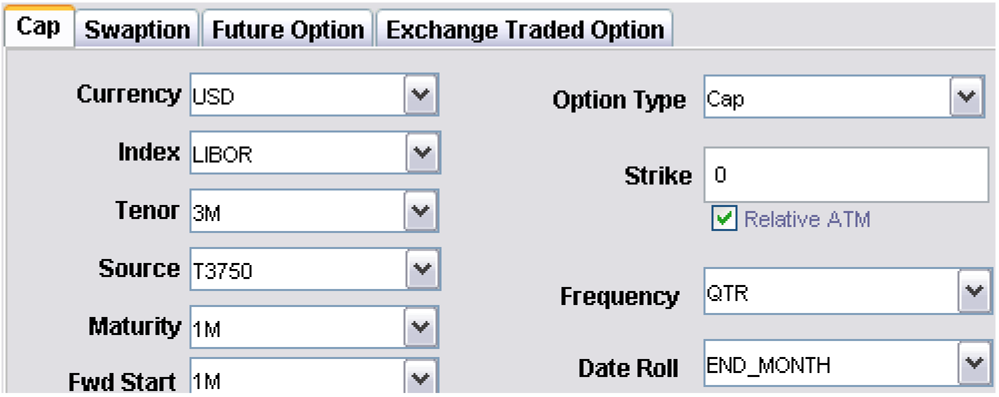
Strike Offset bp Definition



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.

NOTE: The Cap generator may yield inaccurate results when used with relative strike underlying. CapATM is recommended when using relative strikes. CapATM is detailed in the document calib_userguide.doc, which is available through the client portal.

1.2.2 CapBpVols Generator

The bootstrapping algorithm is using the BpVol (a.k.a. Normal) model to generate forward volatilities. Volatility surfaces built using this generator can then expose Black or BpVol volatilities to pricers.

- Please refer to the Calypso Analytics Library guide for complete setup details.

1.2.3 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.

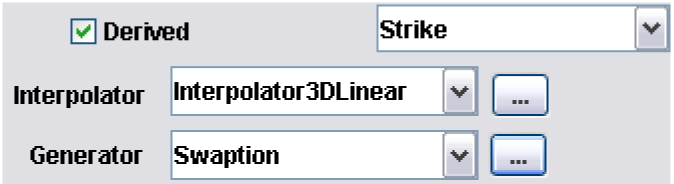
1.2.4 Swaption Generator

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.

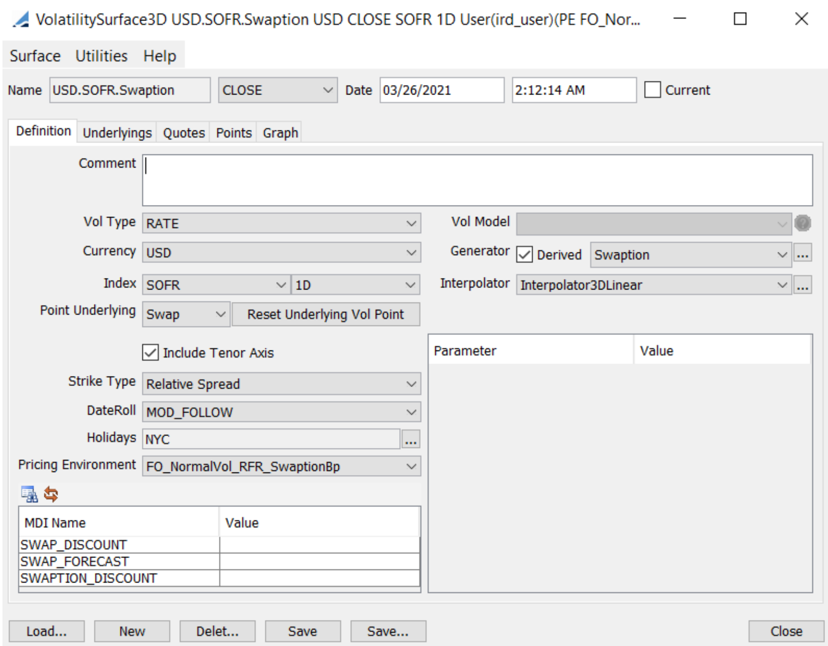
Generation in context of this generator means simply putting the input swaption volatility on the respective vertex point (expiry/tenor/strike) of the volatility surface.

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.

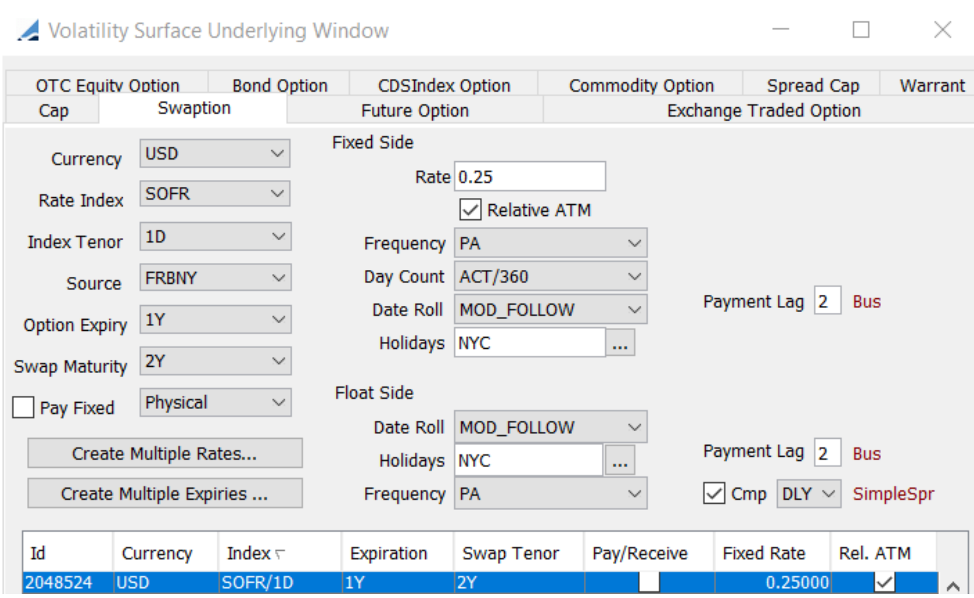
All other quoted strikes are converted to relative strikes by computing a strike offset which is the difference between the actual (absolute) strike and the ATM strike for the given expiry/tenor vertex point.



An RFR swaption volatility surface is supported by this generator. An RFR rate index needs to be assigned in the surface.



A typical RFR Swaption underlying is defined as below. Index Tenor has to be 1D, Cmp checkbox has to be ticked with daily compounding. Compound method SimpleSpr is the only method that's supported.



1.2.5 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.

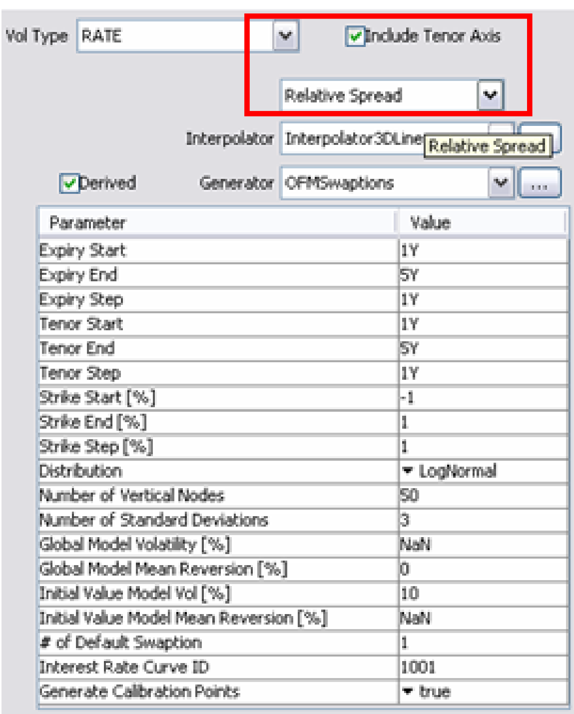
Second, the user has to specify one or more surface underlyings (currently swaptions) on tab 2 and the corresponding market volatilities on tab 3.

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.



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
Global Model Volatility [%]	NaN
Global Model Mean Reversion [%]	0
Initial Value Model Vol [%]	10
Initial Value Model Mean Reversion [%]	NaN
# of Default Swaption	1
Interest Rate Curve ID	1001
Generate Calibration Points	true

1.2.6 CMSBasisAdjSimple Generator

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

1.2.7 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.

An RFR swaption volatility surface is also supported by this generator, similar as generator Swaption.

1.2.8 SwaptionSABR Generator

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

1.2.9 SwaptionSABRDerived Generator

Please refer to the Calypso Analytics Library Guide (CALIB) for complete setup details.

1.2.10 FutureOption Generator

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

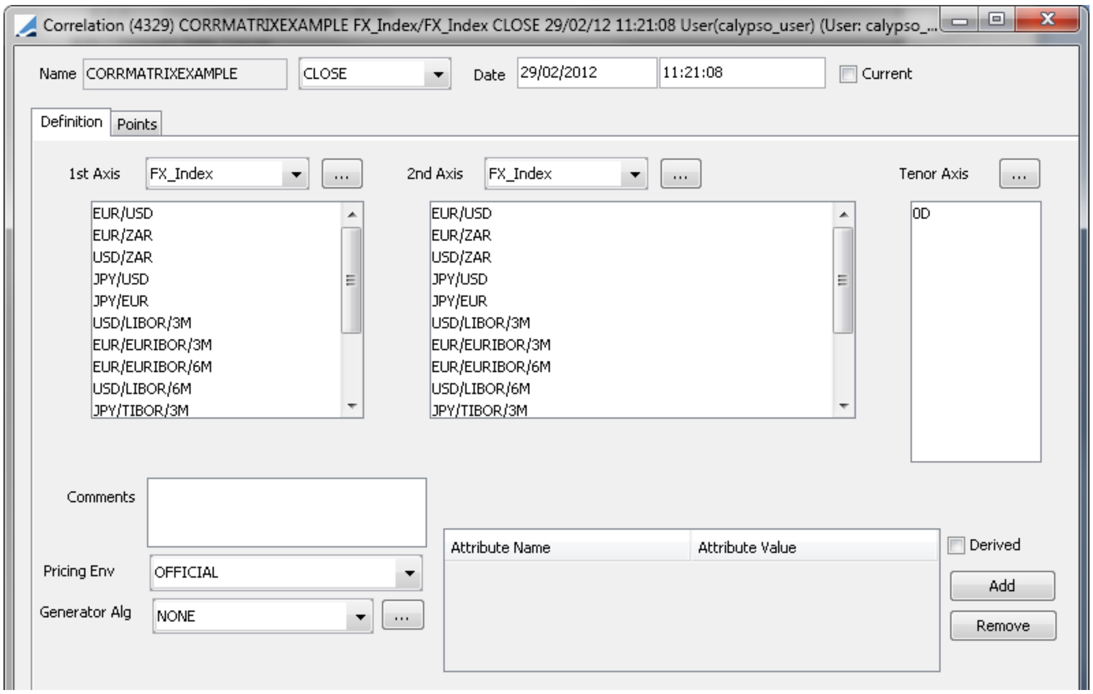
This only applies if the quote type of the future is rate. If the quote type of the future is price, use instead the MMFUTUTE volatility type and the FutureOption generator.

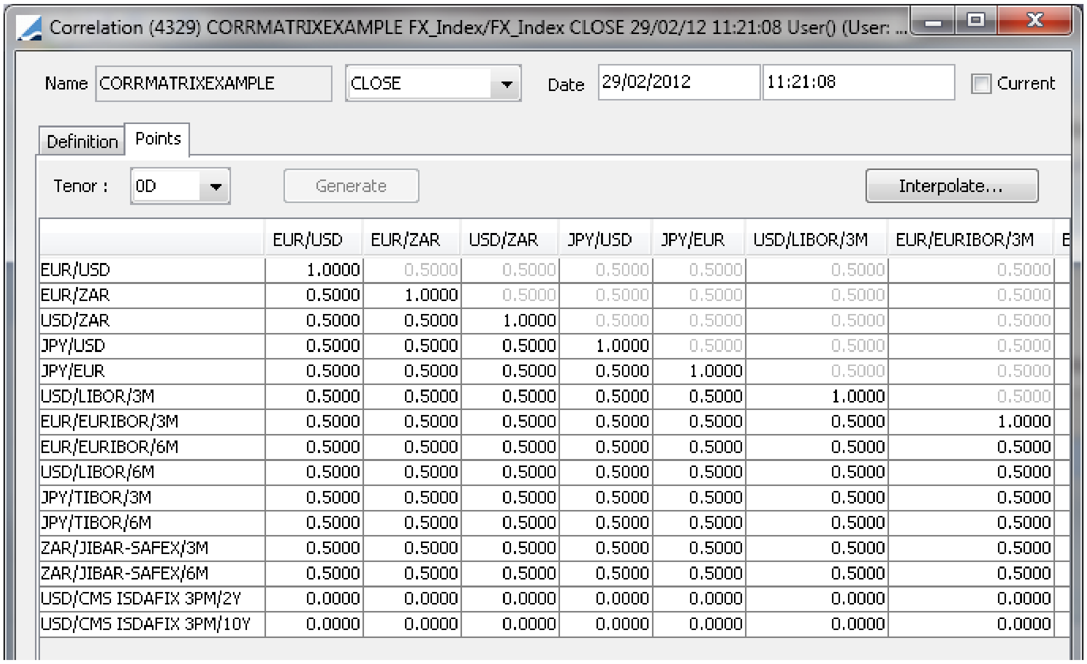
1.3 Correlation Matrices

Correlation Matrices are created for trades, such as Spread Cap/Floor where there are two or more factors that have a quantifiable relationship or Equity trades.

The matrices are simple and populated by the user to fine tune pricing of an asset. Users should create a correlation matrix for each foreseeable product pair available on the axis dropdown.

The screenshot below shows an FX_INDEX v FX_INDEX pair where we are able to compare the correlation of currency pairs. For every currency pair selected in 1st Axis an extra row is added to the points matrix, and for every currency pair selected on 2nd Axis an extra column is generated.





	EUR/USD	EUR/ZAR	USD/ZAR	JPY/USD	JPY/EUR	USD/LIBOR/3M	EUR/EURIBOR/3M
EUR/USD	1.0000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
EUR/ZAR	0.5000	1.0000	0.5000	0.5000	0.5000	0.5000	0.5000
USD/ZAR	0.5000	0.5000	1.0000	0.5000	0.5000	0.5000	0.5000
JPY/USD	0.5000	0.5000	0.5000	1.0000	0.5000	0.5000	0.5000
JPY/EUR	0.5000	0.5000	0.5000	0.5000	1.0000	0.5000	0.5000
USD/LIBOR/3M	0.5000	0.5000	0.5000	0.5000	0.5000	1.0000	0.5000
EUR/EURIBOR/3M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	1.0000
EUR/EURIBOR/6M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
USD/LIBOR/6M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
JPY/TIBOR/3M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
JPY/TIBOR/6M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
ZAR/JIBAR-SAFEX/3M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
ZAR/JIBAR-SAFEX/6M	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000	0.5000
USD/CMS ISDAFIX 3PM/2Y	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
USD/CMS ISDAFIX 3PM/10Y	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000

1.4 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.

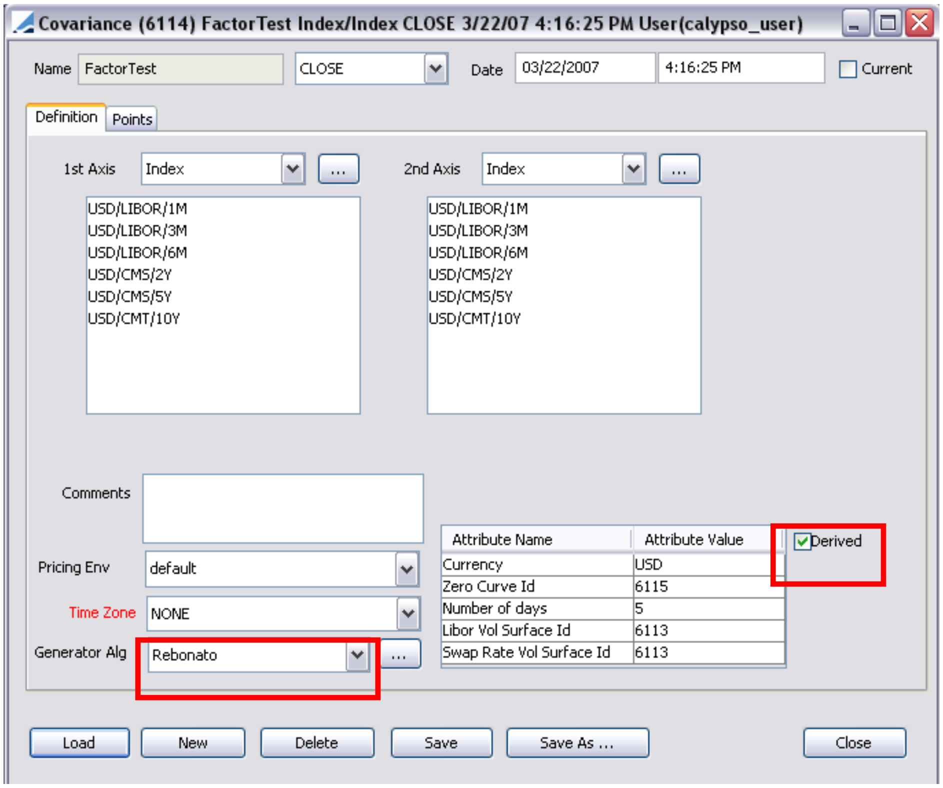
Calypso’s implementation of the model uses Monte Carlo simulation to repeatedly walk down possible paths and then average out the total collection of values to determine an expected value of a given trade.

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.

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.


Covariance (6114) FactorTest Index/Index CLOSE 3/22/07 4:16:25 PM User(calypso_user)
Name FactorTest
CLOSE
Date 03/22/2007
4:16:25 PM
Current
Definition
Points
1st Axis
Index
2nd Axis
Index
USD/LIBOR/1M
USD/LIBOR/3M
USD/LIBOR/6M
USD/CMS/2Y
USD/CMS/5Y
USD/CMT/10Y
USD/LIBOR/1M
USD/LIBOR/3M
USD/LIBOR/6M
USD/CMS/2Y
USD/CMS/5Y
USD/CMT/10Y
Comments
Pricing Env
default
Time Zone
NONE
Generator Alg
Rebonato
Attribute Name
Attribute Value
Currency
USD
Zero Curve Id
6115
Number of days
5
Libor Vol Surface Id
6113
Swap Rate Vol Surface Id
6113
Derived
Load
New
Delete
Save
Save As ...
Close

Section 2. IRD Trades

2.1 Supported Pricers

Types of Trades	Calypso Product	Pricer
Cancelable Cross Currency Swap	CancellableXCCySwap	No native pricing supported.
Cancelable Swap - Single and multiple cancellation dates.	CancellableSwap	PricerCancellableSwap PricerCancSwapOneFactorModel
Cancelable Swap – Vanilla Swap using the Cancelable feature. Single and multiple cancellation dates.	Swap	PricerSwapLGMM1F
Credit Contingent Swap – Vanilla Swap using the CreditContingent feature. Single name or Basket contingency.	Swap	No native pricing supported.
Cross Currency Swap	SwapCrossCurrency	PricerSwap
Cross Currency Swap	XCCySwap	PricerXCCySwap
Vanilla CapFloor Basis CapFloor	CapFloor	PricerCapShiftedLognormal PricerCapFloorMultiFactorModel
Digital CapFloor - Cash or nothing. Single exercise date which decides if holder gets predefined payoff (cash) or not (European style).	CapFloor	PricerCapShiftedLognormal PricerCapFloorHagan PricerCapFloorMultiFactorModel
Capped, Floored, Collared Swap - Each leg can be individually capped, floored or collared.	CappedSwap	No native pricing supported.
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	No native pricing supported.

Types of Trades	Calypso Product	Pricer
Extendible Swap - Single and multiple extension dates.	ExtendibleSwap	No native pricing supported.
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	No native pricing supported.
Vanilla Swap Basis Swap Yield Curve Spread Swap	Swap	PricerSwap PricerSwapMultiFactorModel
Vanilla Swaption (European style)	Swaption	PricerSwaptionLGMM1F PricerSwaptionShiftedLognormal PricerSwaptionBpVol PricerSwaptionSABR PricerSwaptionCEV
American / Bermudan Swaption	Swaption	PricerSwaptionLGMM1F
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.

2.2 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:

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

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 $\bar{T}_i$

– $L(t,T_i,T_{i+1})$ 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 $\widetilde{T}_i$, 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:

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

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

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

2.8 Vanilla Cap Floor

2.8.1 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
- δ_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 - For each caplet, the volatility used is the one for the caplet expiry date and strike

The value of the caplet (floorlet) is then:

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

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

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 $F_{\tilde{T}_i}$).

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

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 $\tilde{T}_i$.

2.8.2 Standard on a Basis Index

A caplet(floorlet) on an index that does have a liquid interest rate market but not a liquid volatility market (such as TIBOR) can be valued by utilizing the assumption that the basis spread between the index and its counterpart with a liquid volatility market (e.g. JPY LIBOR) remains constant at the forward.

Setting the parameter NO_FWD_BASIS_SPREAD_ADJ to false and assigning the basis curve (e.g. TIBOR vs. JPY LIBOR) as the forecast curve, the value of a caplet becomes

$$caplet_i = df_{\tilde{T}_i} \cdot N_i \cdot \delta_i \cdot [F^{base}{}_{\tilde{T}_i} \cdot N(d_1) - K^{adj}{}_i \cdot N(d_1 - \sigma_F(\tilde{T}_i) \cdot \sqrt{\tilde{T}_i})]$$

where $d_1 = \dfrac{\ln(\dfrac{F^{base}{}_{\tilde{T}_i}}{K^{adj}{}_i}) + \dfrac{\sigma_F^2(\tilde{T}_i)\cdot \tilde{T}_i}{2}}{\sigma_F(\tilde{T}_i)\cdot\sqrt{\tilde{T}_i}}$

$F^{base}{}_{\tilde{T}_i}$ is the forward rate computed off the *base* curve (e.g. JPY LIBOR)

$F_{\tilde{T}_i}$ is the forward rate computed off the *basis* curve (e.g. TIBOR)

$K^{adj}{}_i = K_i - \left(F_i - F^{base}{}_i\right)$ is the adjusted, by the forward basis spread, strike

and the implied volatility is picked off the volatility surface associated with the base curve using the adjusted strike. The value of the floorlet follows in a similar fashion.

2.8.3 Amortizing

- See Amortizing structures for a vanilla Swap under Amortizing.

2.8.4 Compounding

- See Compounding structures for a vanilla Swap under Compounding.

2.8.5 Averaging

- See Averaging structures for a vanilla Swap under Averaging.

2.8.6 Convexity correction

- See Convexity correction for a vanilla Swap under Convexity correction.

2.8.7 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.

2.8.8 In Arrears

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

2.8.9 Collar

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

2.8.10 Straddle

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

2.8.11 Negative Forward Rates

PricerCapFloorHagan handle negative forward rates in the following manner.

When interest rates become very low or even negative, the Black model cannot work. It does not permit negative forward rates. Should the pricer encounter a forward rate which is below 1 basis point, the model is altered to a (slightly) shifted lognormal model where the shift is the difference between 1 basis point and the forward rate, which is usually a small quantity.

2.9 Yield Curve Spread Cap Floor

- 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 - For each caplet, the volatility used is the one for the caplet expiry date and strike.
- $S_{2,t}$ is the second forward index valuated at date t.
- b is the factor for that second forward index.
- $\sigma_2(t)$ is the volatility - For each caplet, the volatility used is the one for the caplet expiry date and strike.
- $\rho_{12}(t)$ 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 $F_T = E^{Q_T}(\dfrac{S_{1,T}}{b \cdot S_{2,T} + \overline{K}})$

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:

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

2.10 Basis Cap Floor

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

2.11 Digital Cap Floor

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 - For each caplet, the volatility used is the one for the caplet expiry date and strike.
- F_t is the forward index rate valuated at date t.
- $\sigma_F(t)$ is the volatility - For each caplet, the volatility used is the one for the caplet expiry date and strike,
- K_i is the strike rate of the caplet.

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

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

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

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

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

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).

2.12 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.

2.12.1 PricerSwaptionBpVol

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.

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	EXACT

Pricing Parameters	Type	Description	Typical Value
		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"	
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

The core pricer measures are defined with the theoretical sensitivities now coming from the Bachelier formula.

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	
VEGA	The change in the option value predicted by the option vega: $\Delta V = Vega \bullet \Delta\sigma$ with $\Delta\sigma$ of 1 basis point (0.01%). The option Vega is the theoretical (Bachelier) vega.	

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.

2.12.2 PricerSwaptionCEV

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%.

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_HIGHORDER

Pricing Parameters	Type	Description	Typical Value
		HAGAN_WOODWARD_LOW ORDER, 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"	

Pricer measures:

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.

2.13 Bermudan / American Swaption

Please refer to the Calypso Analytics Library (CALIB) User Guide.

2.14 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
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

Pricing Parameters	Type	Description	Typical Value
		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.	
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

2.15 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
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

2.16 (Digital) Spread Cap/Floor

2.16.1 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 computes the index forwards using the methodology 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.

Correlation

Set up of correlation matrices is shown in Correlation Matrices Section 1.3.

Digital CMS Spread Options

PricerSpreadCapFloorGBM2F is used to price cap/floor spread options, when these options are digital Calypso provides four methods for valuation.

Digital spread options are calculated using replication methods detailed by Hagan (2005). The three replication methods, Sub, super and central replication, are available in Calypso through the DIGITAL_VALUATION_METHOD pricing parameter.

The fourth replication method assumes lognormal distribution and is calculated using Black volatility at the strike of the option and is selected using the string THEORETICAL in DIGITAL_VALUATION_METHOD.

More details on the Calypso implementation of these replication methods can be found in the Calypso whitepaper – 'Valuation of digital options in the presence of a smile', Calypso (2010).

Pricing Parameters

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_VALUATION_METHOD	String	CENTRAL_REPLICATE: $NPV = \frac{1}{\varepsilon}\left[C_0\left(K-0.5\varepsilon\right)-C_0\left(K+0.5\varepsilon\right)\right]$ SUB_REPLICATE: $NPV = \frac{1}{\varepsilon}\left[C_0\left(K-0.5\varepsilon\right)-C_0\left(K\right)\right]$ SUPER_REPLICATE: $NPV = \frac{1}{\varepsilon}\left[C_0\left(K\right)-C_0\left(K+0.5\varepsilon\right)\right]$ THEORETICAL	CENTRAL_REPLICATE

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

Hagan, P. S. (2005). "Accrual swaps and range notes." *working paper*.

Calypso technology, (2010). “Valuation of digital options in the presence of a smile”.

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

2.17 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

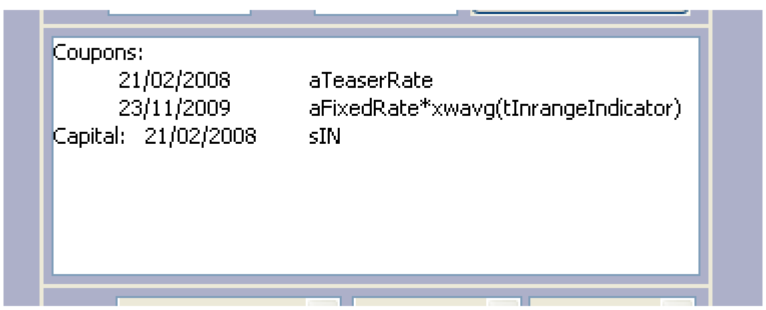
Valuation

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.

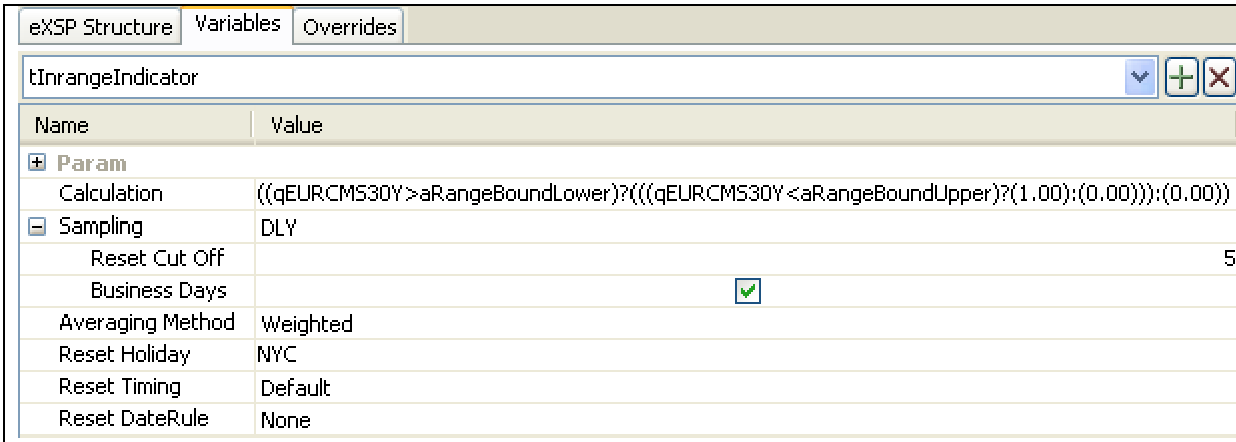
EXSP Terms and Conditions Setup

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

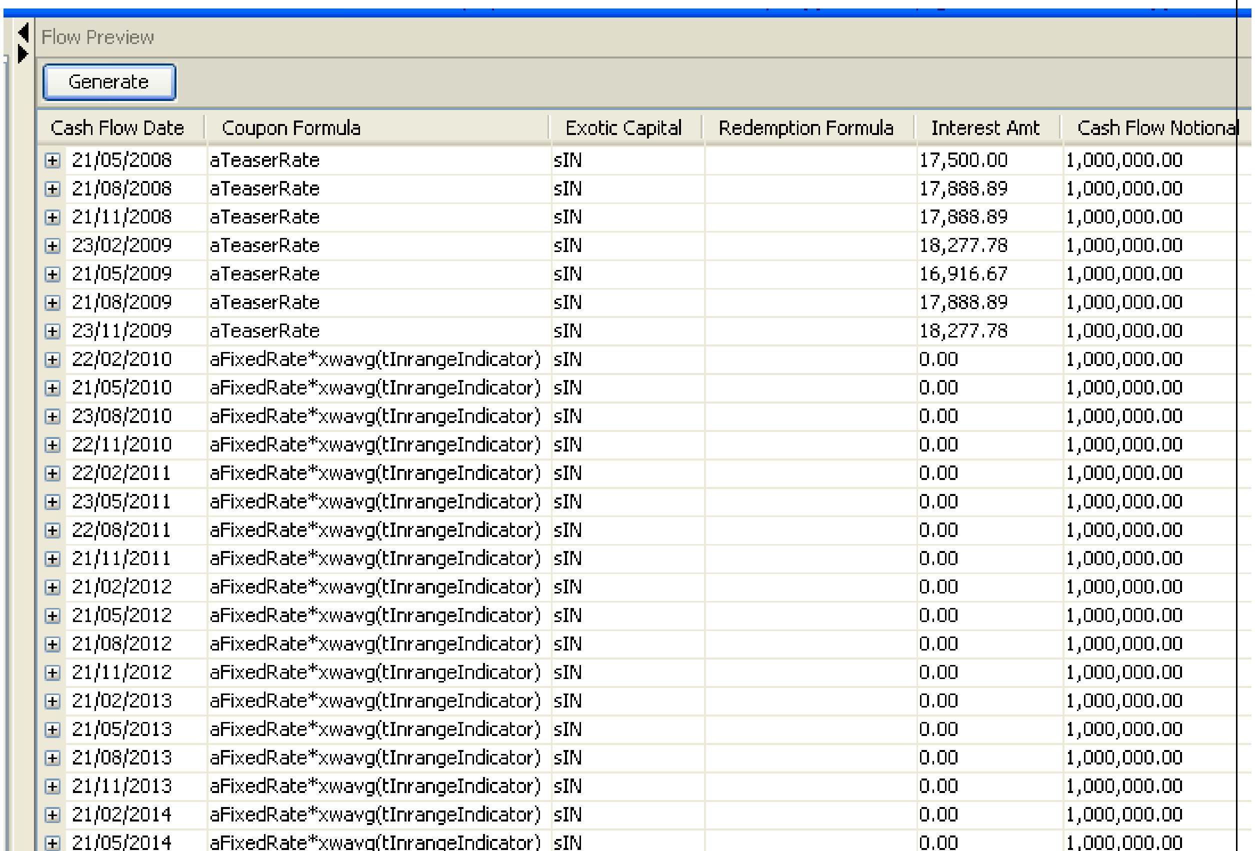


In Range Indicator



Name	Value
Param	
Calculation	((qEURCMS30Y>aRangeBoundLower)?(((qEURCMS30Y<aRangeBoundUpper)?(1.00):(0.00))):(0.00))
Sampling	DLY
Reset Cut Off	5
Business Days	✓
Averaging Method	Weighted
Reset Holiday	NYC
Reset Timing	Default
Reset DateRule	None

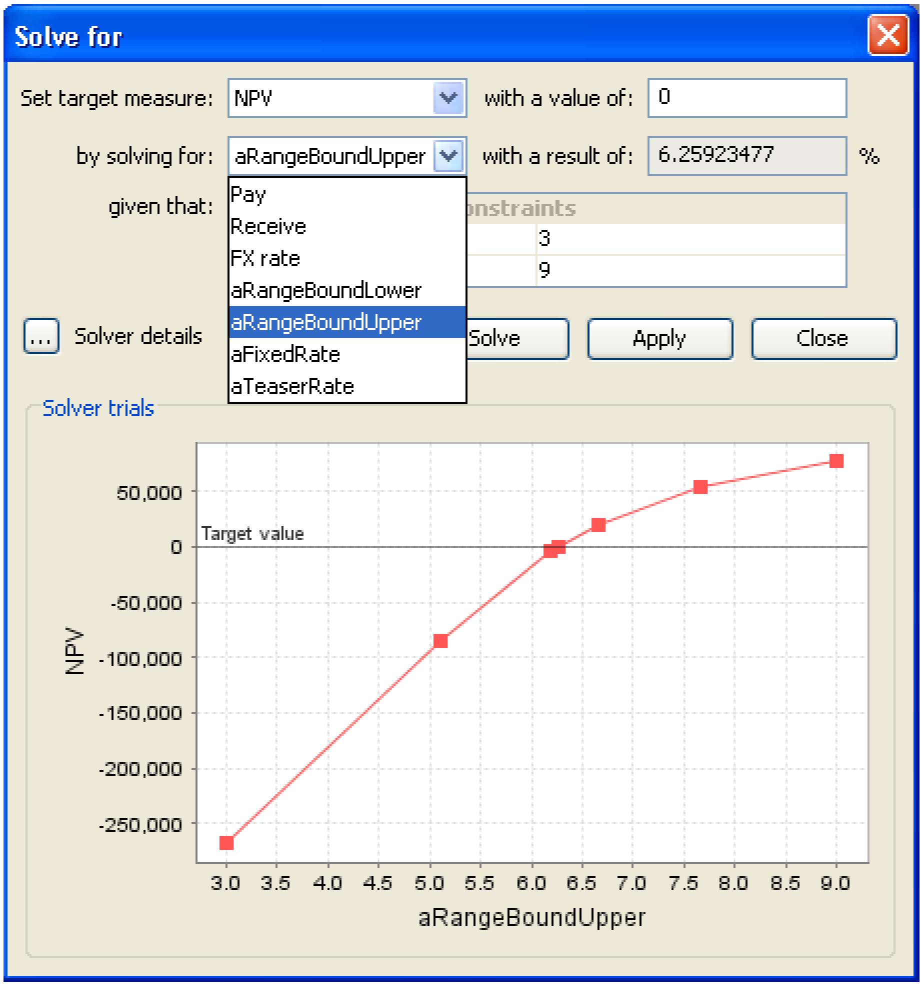
Teaser Rate



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

2.17.1 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

Section 3. Analytics

3.1 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 3-1.

3.1.1 Call Options

NPV

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

where

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

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

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

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

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

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

$$\begin{aligned} n(d_2) &= \frac{e^{\frac{-d_2^2}{2}}}{\sqrt{2\pi}} \\ &= \frac{e^{\frac{-d_1^2}{2} + \ln(\frac{F}{K})}}{\sqrt{2\pi}} \\ &= \frac{Fe^{\frac{-d_1^2}{2}}}{K\sqrt{2\pi}} \\ &= n(d_1)\frac{F}{K} \qquad (3) \end{aligned}$$

Delta

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

Substituting equation (3) gives :

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

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

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

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

Delta Premium

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

Delta Forward

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

Substituting equation (3) gives :

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

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

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

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$_{r_g}$

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

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

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

Substituti ng 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$_q$

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

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

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

Substituti ng 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 $t_3 = T_3 - \tau$:

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

Substituti ng 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 :

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

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

3.1.2 Put Options

NPV

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

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

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

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

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

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

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

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 derivative s 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$_{r_g}$

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

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

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

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

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

Substituti ng 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

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

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

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

Substituti ng 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 :

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

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

3.2 Black Model

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

3.3 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.

Variable	Description
$\sigma(t)$	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

Model	f(r)	Distribution	Volatility	Mean reversion
Normal Ho-Lee	r	Normal	Constant or Term structure	0
Lognormal Hoo-Lee	$\ln(r)$	lognormal	Constant or Term structure	0

3.4 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.

3.5 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.

3.6 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.

3.7 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.