Algorithmica
Research AB

Dual bootstrap in Quantlab

< Guide for use of the dual bootstrap functionality >

Last update: 2012-04-12, ver 1.15.

1. Introduction

This document describes how to use the dual curve bootstrap functionality in Quantlab 3.1. By dual curve bootstrap, we refer to the new paradigm in the swap market for curve construction where the forward index rates (projecting rates) and discount rates are separate curves. The “old style” curve construction used the same curve for the forward index rates and for discounting. The market standard discount curve now consists of OIS rates.

Curve construction in the swap market, in the new paradigm, means paying attention to the tenor of the forward index rates (i.e. tenor basis spreads) and carefully select the instruments from which the curves are created and the blending principles applied. Normally it is necessary to also create synthetic instruments (e.g. FRA and/or Deposits) in specific, basis curve consistent, ways to fill up some segments of the forward curve or discount curve. In addition, it is desirable to obtain smooth (forward) curves. Quantlab offers several interpolation models with excellent smoothness properties (e.g. Adams-Deventer Maximum Smoothness-model and Hagan-West forward monotone convex spline). It is also possible to apply Hyman filters to all spline models. Furthermore, when creating the forward index rates, in Quantlab, the interpolation models always operate directly on the tenor rate that is calculated, ensuring a “smooth” behavior of the forward rates. The swap curve in each currency is, in the new paradigm, best described as a swap-“surface” where the extra dimension is the tenor of the forward index rate. Quantlab also provides a swap-surface object from which interpolated rates for non-standard tenors can be obtained.

For bootstrapping the discounting curve (single curve bootstrap), Quantlab includes, in addition to all standard single curve bootstrap features, modeling with a step function between Central bank meeting dates while, simultaneously, using a different interpolation model for the longer end of the curve.

Quantlab 3.1 encapsulates all the complexities involved and provides ways to create discount functions and forward curves in a very succinct way.

Typically the instruments used in dual bootstrap are (all supported):

1. OIS swaps
2. Deposits (normally synthetic)
3. Forward rate agreements (FRA) (real and synthetic)
4. Money market futures
5. IRS (fixed/floating interest rate swap)
6. Basis swaps - as 2 swaps

7. Basis swaps - as 1 swap

There are principally two ways to use the dual bootstrap functionality. One is to use a set of core Quantlab functions and classes. This will give the user maximum flexibility and control of every aspect of the curve building process. The other way is to use a set of wrapper functions that tries to provide “best-practice” curve building for a number of specific situations and therefore facilitates the creation of discount functions and forward functions. This document describes the latter. The code for the wrapper functions are available for viewing in Qlang, making it very easy for the user to create different versions as well as understanding exactly the details involved.

A selection of the provided wrapper functions are explained in this document, as follows:

1. **disc_curve_create (base):** builds a discount function, for example from OIS instruments,

2. **disc_curve_create (extended):** builds a discount function with the possibility to extend the curve if it is shorter than the swap curve.

3. **fwd_curve_create:** builds a forward function from FRA/IRS (and synthetic instruments)

4. **fwd_b2_curve_create:** builds a forward function from FRA/IRS/Basis swap(2swaps) (and synthetic instruments)

5. **fwd_b1_curve_create:** builds a forward function from FRA/IRS/Basis swap(1swap) (and synthetic instruments)

6. **swap_surface_b2_create:** creates a swap curve surface with all tenors from FRA/IRS/Basis swap(2swaps). (A swap_surface_b1_create is available but not discussed)

Quantlab 3.1 also includes additional variants of these wrapper functions - for example instead of a Basis swap it is possible to use an IRS directly for the tenor we are calculating.

Additionally, cross currency curve building is also supported but not included in this document.

2. Creating a discount function

A discount function can be created with a single wrapper function call. There are two main versions, a base version and an extended version. The extended version allows for extending the discounting curve based on a second curve by extrapolating the basis spread between the discounting curve and the extrapolation curve.

Base version:

```
disc_func     disc_curve_create(    date                            trade_date,
                                    curve option(nullable)          short_curve,
                                    curve option(nullable)          middle_curve,
                                    curve option(nullable)          long_curve,
                                    curve_segment                   prio1,
                                    curve_segment                   prio2,
                                    logical                         no_overlap_middle,
                                    logical                         merge_middle,
                                    integer                         blend_buf_days,
                                    disc_z_model                    disc_model,
                                    interpolator option(nullable)   disc_ip,
                                    rate_type option(nullable)      disc_rt,
                                    vector(dates) option(nullable)  cb_dates,
                                    number option(nullable)         step_end,
                                    out swap_curve_ext              disc)
```

Arguments and return values explained:

Return value	Comment
disc_func	The disc_func object represents a discount function. This object has many member functions (see the Quantlab function browser).

Arguments	Data type	Comment
trade_date	date	Trade date.
short_curve	curve	The curve for the short segment of the discounting curve. Typically, this is the segment for instruments with time-to-maturity of less than 1 year.
middle_curve	curve	The curve for the middle segment of the discounting curve. This segment of the curve is reserved for forward starting instruments such as FRA or Futures.

long_curve	curve	The curve for the long segment of the discounting curve.
prio1	curve_segment	Indicates which segment has first priority (SHORT, MIDDLE or LONG)
prio2	curve_segment	Indicates which segment has second priority (SHORT, MIDDLE or LONG)
no_overlap_middle	logical	Determines how the blending between the short and middle segments of the curve is done in terms of the last included short instrument. If this flag is true (false) then the blending assures that maturity of the last short instrument is before settlement (maturity) of the first middle instrument. Ex. if false then a 3M Deposit is allowed together with a 2X5 FRA.
merge_middle	logical	The middle segment is merged into the curve.
blend_buf_days	integer	The minimum number of days between “corner” instruments for the curve segments in the curve blending.
disc_model	disc_z_model	DZ_BOOT: standard bootstrap DZ_STEP: bootstrap with a piecewise flat O/N rate. DZ_MAX_SMOOTH: Adams/Deventer maximum smoothness model. DZ_BOOT_STEP: standard bootstrap combined with a piecewise flat O/N rate, up to the step_end date (see below). DZ_TANGGAARD: Maximum smoothness model with residuals. [DZ_TENSION: Tension splines] (Quantlab provides many more models of type “best fit” that are available when using the core functions)
disc_ip	interpolator	Interpolation model object (applicable only for DZ_BOOT and DZ_BOOT_STEP). Available models includes: 1. ip_hagan_west(logical force_pos): Hagan/West monotone convex spline, if force_pos = true, the forward rates are enforced to be positive. 2. ip_hermite_4th_order() : 3. ip_hermite_akima(): 4. ip_hermite_catmull_rom() : 5. ip_hermite_finite_diff (): 6. ip_hermite_fritsch_butland() : 7. ip_hermite_kruger(): 8. ip_hermite_monotone(): 9. ip_hermite_parabolic(): 10. ip_linear(): linear interpolation 11. ip_spline(): cubic spline interpolation

		12. ip_step(): step function interpolation. The interpolator object is created from the above constructors. Each constructor has several more arguments (not shown) related to extrapolation and Hyman filters. It is possible to define extrapolation properties both in the front and end of the curve. It is also possible to apply different types of Hyman filters(eg. monotonicity) to all interpolators except linear and the Hagan/West model. All interpolation models are local except spline. A detailed discussion on these models is beyond the scope of this document.
disc_rt	rate_type	Interpolation rate types (applicable only for DZ_BOOT). Available choices are: RT_CONT: Continuous compounding. RT_ON: Daily compounding. RT_SIMPLE: Simple compounding. RT_EFFECTIVE: Annual effective compounding. RT_SEMI_ANNUAL: Semi-annual compounding. RT_QUARTERLY: Quarterly compounding. RT_MONTHLY: Monthly compounding. RT_BANKDISC: Bank discount rate. RT_LOGDF: Log of the discount function Note: the Hagan/West model is only defined for RT_CONT.
cb_dates	vector(dates)	Central bank meeting dates (not used for a step model).
step_end	date	If a step model is used this date defines the end of the step period.

out Arguments	Data type	Comment
disc	swap_curve_ext	This object represents the created discounting curve object from which several results and information can be retrieved e.g. the blended curve.

Extended version:

disc_func	**disc_curve_create**(	date	trade_date,
		curve option(nullable)	short_curve,
		curve option(nullable)	middle_curve,
		curve option(nullable)	long_curve,
		curve	extrap_curve,

```
string                              class_name,
curve_segment                       prio1,
curve_segment                       prio2,
logical                             no_overlap_middle,
logical                             merge_middle,
integer                             blend_buf_days,
disc_z_model                        disc_model,
interpolator option(nullable)       disc_ip,
rate_type option(nullable)          disc_rt,
disc_z_model                        extrap_model,
interpolator option(nullable)       extrap_ip,
rate_type option(nullable)          extrap_rt,
vector(dates) option(nullable)      cb_dates,
number option(nullable)             step_end,
out swap_curve_ext                  disc)
```

Most arguments are the same as for the base version above and, hence, are not explained here. The additional arguments are:

Arguments	Data type	Comment
extrap_curve	extrap_curve	Base curve used to extend the discounting curve. The synthetic instruments appended to the discounting curve are Deposits or FRA contracts and their rates are calculated by extrapolating the basis spread.
class_name	string	Class name for the instruments to be added (Deposit or FRA).
extrap_model	disc_z_model	DZ_BOOT: standard bootstrap DZ_MAX_SMOOTH: Adams/Deventer maximum smoothness model. DZ_TANGGAARD: Maximum smoothness model with residuals. [DZ_TENSION: Tension splines] (Quantlab provides many more models of type “best fit” that are available when using the core functions)
extrap _ip	interpolator	Interpolation model object (applicable only for DZ_BOOT). Available models includes: 1. ip_hagan_west(logical force_pos): Hagan/West monotone convex spline, if force_pos = true, the forward rates are enforced to be positive. 2. ip_hermite_4th_order() : 3. ip_hermite_akima():

		4. ip_hermite_catmull_rom() : 5. ip_hermite_finite_diff (): 6. ip_hermite_fritsch_butland() : 7. ip_hermite_kruger(): 8. ip_hermite_monotone(): 9. ip_hermite_parabolic(): 10. ip_linear(): linear interpolation 11. ip_spline(): cubic spline interpolation 12. ip_step(): step function interpolation. For details see previous function.
extrap _rt	rate_type	Interpolation rate types (applicable only for DZ_BOOT). Available choices are: RT_CONT: Continuous compounding. RT_ON: Daily compounding. RT_SIMPLE: Simple compounding. RT_EFFECTIVE: Annual effective compounding. RT_SEMI_ANNUAL: Semi-annual compounding. RT_QUARTERLY: Quarterly compounding. RT_MONTHLY: Monthly compounding. RT_BANKDISC: Bank discount rate. RT_LOGDF: Log of the discount function Note: the Hagan/West model is only defined for RT_CONT.

3. Creating a forward function

3.1 FRA/IRS

A forward function based on a short curve, FRA curve and an IRS curve can be created with a single wrapper function call.

```
fwd_func      fwd_curve_create(   date                               trade_date,
                                  disc_func                          df_disc,
                                  disc_func                          df_synt_base,
                                  curve option(nullable)             short_curve,
                                  curve option(nullable)             fra_curve,
                                  curve  option(nullable)            swap_curve,
                                  curve_segment                      prio1,
                                  curve_segment                      prio2,
                                  string                             depo_class_name,
                                  string                             fra_class_name,
                                  synt_short_style                   synt,
                                  interpolator option(nullable)      fwd_ip,
                                  logical                            merge_fra,
                                  integer                            blend_buf_days,
                                  vector(number) option(nullable) edfut_cvx_adj,
                                  out swap_curve_ext                 fwd_crv,
                                  number option(nullable)            swap_first_fix)
```

Arguments and return values explained:

Return value	Comment
fwd_func	The fwd_func object represents a forward rate function (see the Quantlab function browser for available member functions).

Arguments	Data type	Comment
trade_date	date	Trade date.
df_disc	disc_func	The discount function e.g. OIS.
df_synt_base	disc_func	Discount function used as base discount function when creating the synthetic instruments.
short_curve	curve	The curve for the short segment of the forward curve.
fra_curve	curve	The curve for the FRA/Futures segment of the forward curve.

		All FRAs must have the same tenor and the same tenor as the IRS floating leg.
swap_curve	curve	The curve for the IRS segment of the forward curve. All swaps must have the same tenor of the floating leg and this tenor must match all the FRA/Futures contract tenors.
prio1	curve_segment	Indicates which segment has first priority (SHORT, FRA or IRS)
prio2	curve_segment	Indicates which segment has second priority (SHORT, FRA or IRS)
depo_class_name	string	Class name for relevant Deposits (for synthetic instruments).
fra_class_name	string	Class name for relevant FRA contracts (for synthetic instruments).
synt	synt_short_style	Possible choices are SY_FRA, SY_DEPO and SY_NONE. The two built-in ways to create synthetic instruments for the short end of the curve up to the first FRA/ Futures contract. (As explained above, if the user wish to use a different principle it is possible to access the core functions directly.) 1. SY_FRA: In this case, both synthetic FRAs and a synthetic Deposit are created. The FRAs are all of the relevant tenor and starting 1w, 2w and every month until the first FRA starts. One Deposit equal to the tenor is added to the curve. 2. SY_DEPO: Only synthetic Deposits are created. Tenors are O/N, T/N, 1W, 2W and every month until the first FRA starts. This setting should be used with caution for forward curve creation. 3. SY_NONE: No synthetic instruments are created. For details on the creation of synthetic short instruments, see below.
fwd_ip	interpolator	Interpolation model object for the tenor rates. See chapter 2 for details.
merge_fra	logical	The FRA contracts are merged into the curve.
blend_buf_days	integer	The minimum number of days between curve segments in the curve blending.
vector(number)	edfut_cvx_adj	A vector of spreads that adjusts the FRA/Futures segment of

		the curve. This parameter can, for example, be used to provide a convexity adjustment for Futures.
number	swap_first_fix	The first floating rate fixing for the IRS. If null, no fixing is used.

out Arguments	Data type	Comment
fwd_crv	swap_curve_ext	This object represents the created forward curve object from which several results and information can be retrieved e.g. the blended curve.

3.2 Creating synthetic short instruments (before the first FRA)

The synthetic instruments, if included, are created from a base discount function and its basis spread vs. the FRA contracts.

Step 1. Create a basis spread function using all the FRAs and the provided base discount function. The base discount function could an OIS discount function, a discount function calculated from a different tenor forward function or any other discount function.

Step 2. Calculate the implied rates for the synthetic instruments from the base discount function and add the interpolated basis spread from step 1.

The created instruments now have a basis consistent with the FRA contracts and they will provide added information for the short end of the curve. Their curvature will depend on the base discount function and the interpolated basis spread.

3.3 Forward index rate interpolation

The interpolation models used when creating a forward function operates directly on the tenor rate we wish to calculate. This ensures that we get forward rates without the oscillating behavior that sometimes can be seen in the standard bootstrapping models for a discount function (when interpolating different types of spot rates). See below for some typical forward rate examples.

3.4 The forward function <fwd_func>

The forward function contains all the forward rates for any date for a certain tenor. The member function fwd() which takes an Act365 period (from trade date) as argument returns the rate. Of course, if the period corresponds to a point used in the original curve we get this point exactly otherwise, it is interpolated with the model used in the bootstrap-function.

For example, assume the forward function is calculated for a 6M tenor and we wish to calculate the forward rate in one month i.e. we would like to know the 1x7 forward rate. Assume today is 9-Sep-2011 and the 1M date is 13-Oct-2011.

```
fwd_func f_fwd = fwd_curve_create(…);
fwd_rate_1x7 = f_fwd.fwd((#2011-10-13 - #2011-09-09)/365);
```

3.5 FRA/IRS/BasisSwap(2Swap)

This version uses FRA/Futures, IRS and Basis swaps (2 Swaps) while the short end of the curve is created synthetically.

```
fwd_func     fwd_b2_curve_create( date                              trade_date,
                                  disc_func                         df_disc,
                                  disc_func                         df_synt_base,
                                  curve option(nullable)            short_t1_curve,
                                  curve option(nullable)            fra_t1_curve,
                                  curve  option(nullable)           swap_t2_curve,
                                  curve  option(nullable)           basis_t1t2_curve,
                                  curve_segment                     prio1,
                                  curve_segment                     prio2,
                                  string                            depo_class_name,
                                  string                            fra_class_name,
                                  synt_short_style                  synt,
                                  interpolator option(nullable)     fwd_ip,
                                  logical                           merge_fra,
                                  integer                           blend_buf_days,
                                  vector(number) option(nullable) edfut_cvx_adj,
                                  logical                           x_pol_basis,
                                  out swap_curve_b2_ext             fwd_crv,
                                  number option(nullable)           swap_first_fix_t1)
```

Since most arguments are the same as for the previous wrapper, only the unique arguments will be explained. Note, t1 and t2 stands for the different tenors involved.

Arguments	Data type	Comment
basis_t1t2_curve	curve	The curve for the Basis swap of the forward curve. All Basis swaps must have one tenor of the floating leg that matches the IRS curve. The tenor of the other floating leg must match the FRA contracts. The spreads must be a 2 swaps quotation.
x_pol_basis	logical	If true the basis curve will be extrapolated, with a constant basis spread, up to the length of the swap curve.

out Arguments	Data type	Comment
fwd_crv	swap_curve_b2_ext	This object represents the created forward curve object from which several results and information can be retrieved e.g. the blended curve.

Note that the FRA contracts correspond to the tenor we wish to calculate. The IRS combined with the Basis swap will convert the IRS to the FRA tenor. For example, the FRA curve could be a [EUR 3M FRA]-curve, the IRS curve could be a [EUR IRS 6M EURIBOR]-curve and the Basis swap could be a [EUR 3M/6M EURIBOR]-curve. In this case t1 = 3M and t2 = 6M. The [EUR IRS 6M EURIBOR]-curve will be converted to a [EUR IRS 3M EURIBOR]-curve and we proceed with calculations for the 3M tenor. We could also, with the same principle, calculate the 6M tenor. A difference in daycount conventions between the Basis swap and the IRS is handled by converting the basis spread.

3.6 FRA/IRS/BasisSwap(1Swap)

This version uses FRA/Futures, IRS and Basis swaps (1 Swap) while the short end of the curve is created synthetically.

```
fwd_func      fwd_b1_curve_create( date                        trade_date,
                                   disc_func                   df_disc,
                                   disc_func                   df_synt_base,
                                   curve option(nullable)      short_t2_curve,
                                   curve option(nullable)      fra_t2_curve,
                                   curve  option(nullable)     swap_t2_curve,
                                   curve option(nullable)      short_t1_curve,
curve option(nullable)            fra_t1_curve,
curve  option(nullable)           basis_t1t2_curve,
curve_segment                     prio1,
curve_segment                     prio2,
string                            depo_class_name,
string                            fra_t1_class_name,
synt_short_style                  synt,
interpolator option(nullable)     fwd_ip,
logical                           merge_fra,
integer                           blend_buf_days,
vector(number) option(nullable) edfut_cvx_adj_t2,
vector(number)  option(nullable) edfut_cvx_adj,
out swap_curve_b1_ext             fwd_crv,
number option(nullable)           first_fix_t1
number option(nullable)           first_fix_t2)
```

This function is similar to the previous case with the difference that we have a complete FRA/IRS curve for the tenor in the basis swap we take as input. There is one more wrapper available (not shown here) that instead of the tenor = t2 curve takes a tenor = t2 forward function as input.

out Arguments	**Data type**	**Comment**
fwd_crv	swap_curve_b1_ext	This object represents the created forward curve object from which several results and information can be retrieved e.g. the blended curve.

The t1 curves represent the tenor we wish to calculate. The t2 curves will create a t2 forward function according to section 3.1. The t2 forward function combined with the Basis swap will create instruments with the same tenor as the t1 curves. For example, the t2-FRA curve could be a [EUR 6M FRA]-curve, the t2-IRS curve could be a [EUR IRS 6M EURIBOR]-curve, the t1-FRA-curve could be [EUR 3M FRA]-curve and the Basis swap could be a [EUR 3M/6M EURIBOR]-curve. In this case, we calculate the 3M tenor. We could also, with the same principle, calculate the 6M tenor.

4. Creating a swap surface

The single currency swap curve can be represented as a surface where the extra dimension is the tenor of the forward index rate. The discount function is assumed to be the lowest tenor and will anchor the rate interpolation below the first forward tenor. Extrapolation is currently not done beyond the longest tenor.

There are several wrapper functions for creating the swap surface depending on what instruments are used and other assumptions. Here, the function which uses the FRA/IRS/Basis swap(2 swaps) is shown and will represent the principles.

```
swap_curve_surface    swap_surface_b2_create(    date                              trade_date,
                                                 disc_func                         df_disc,
                                                 curve option(nullable)   short_main_crv,
                                                 curve option(nullable)            fra_main_crv,
                                                 curve option(nullable)            swap_main_crv,
                                                 vector(curve))                    short_t_crv,
                                                 vector(curve)                     fra_t_crv,
                                                 vector(curve)                     basis_t_crv,
                                                 curve_segment                     prio1,
                                                 curve_segment                     prio2,
                                                 string                   depo_class_name,
                                                 string                   fra_main_class_name,
                                                 vector(string)           fra_t_class_name,
                                                 synt_short_style                  synt,
                                                 interpolator option(nullable)     bs_ip,
                                                 logical                           merge_fra,
                                                 number                   blend_buf_days,
                                                 vector(number) option(nullable) edfut_cvx_adj,
                                                 logical                           x_pol_basis,
                                                 number option(nullable) swap_first_fix )
```

The arguments that have been explained previously will not be discussed.

Return value	Comment
swap_curve_surface	The swap_curve_surface object is an “interpolation”-object and contains the discount function and all tenor forward functions.

Arguments	Data type	Comment
short_main_curve	curve	The main tenor curve for the short segment of the forward curve.
fra_main_curve	curve	The main tenor curve for the FRA/Futures segment of the forward curve. All FRAs must have the same tenor and the same tenor as the main IRS floating leg.

swap_main_curve	curve	The main tenor curve for the IRS segment of the forward curve. All swaps must have the same tenor of the floating leg and this tenor must match all the FRA/Futures contract main tenors.
short_t_curve	curve	The non-main tenor short curves.
fra_t_curve	vector(curve)	The non-main tenor FRA/Futures curve.
basis _t_curve	vector(curve)	The Basis swap curves. All Basis swaps must have one tenor of the floating leg that matches the main tenor IRS curve. The tenor of the other floating leg must match the corresponding non-main tenor FRA/Futures curve. The spreads must be a 2 swaps quotation.

5. Curves

All curves needs to be setup to include only one type of instrument. The blending of different types of instruments will be done at runtime and blending of instruments with different tenors will typically generate an error (for forward curves).

A complete set of curves for a currency thus comprises of:

- Discounting curve e.g. OIS.
- IRS curve for the main floating leg tenor.
- FRA's or Futures for the main tenor.
- Basis swap curves (or IRS curves) for other tenors.
- FRA's or Futures for other tenors.

As an example, for EUR this would typically mean:

- Eonia is used for discounting.

6M tenor (main):

- IRS: Annual fixed vs. 6M EURIBOR; 1yr – 60yrs.
- FRA: 6M EURIBOR; 0x6, 6x12, 12x18, 18x24 and 1x7, 2x8, …

3M tenor:

- Basis swaps (quoted as either 1 swap or 2 swaps): 3M/6M; 1yr – 50 yrs.
- Money market futures: 3M EURIBOR (convexity adjusted)
- FRA: 3M EURIBOR; 0x3, 3x6, 6x9,… and 1x4, 2x5, …

1M tenor:

- Money market monthly IRS: 2M-12M.

- Basis swaps (quoted as either 1 swap or 2 swaps): 1M/6M; 1yr – 50 yrs.

- FRA: 1M EURIBOR; 0x1

12M tenor:

- Basis swaps (quoted as either 1 swap or 2 swaps): 6M/12M; 1yr – 50 yrs.

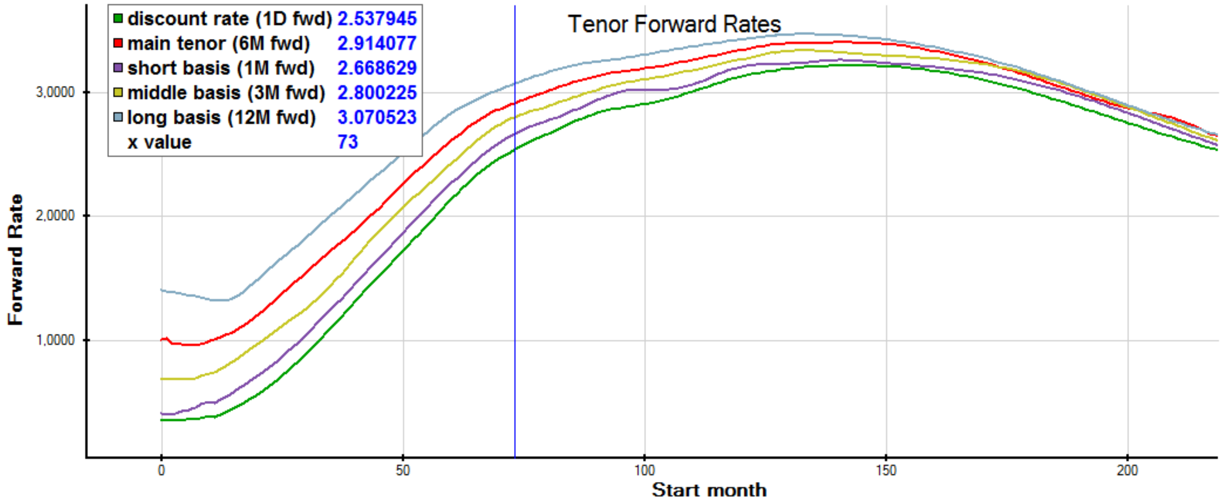
- FRA: 12M EURIBOR; 12x24

As an example of typical a result, let us look at the EUR curves for 14-Mar-2012 (using ICAP prices provided by Reuter).

In the graph below the OIS, 1M, 3M, 6M and 12M tenors are shown. The curves are created from:
OIS: EONIA swaps
1M: Money Market IRS and 1M/6M Basis swaps,
3M: 3M Futures and 3M/6M Basis swaps,
6M: 6M FRA and 6M IRS,
12M: 12M FRA and 6M/12M Basis swaps

In addition, for each forward tenor, synthetic Deposits and FRAs are created to fill up the short end before the first instrument. The rates for the synthetic instruments are calculated from an interpolated basis vs. the OIS curve.

In the graph below, only "exact fit"-models are used. The interpolation model for the OIS curve is the Hagan/West forward monotone convex spline. The forward rate interpolation model is a hermite cubic spline.



Created in Quantlab