Bootstrap

Christer Rydberg,

Nov 14, 2012

1 Vanilla IRS

A vanilla IRS is priced as,

$$NPV(t) = -NPV(t)_{fix} + NPV(t)_{flt}, \tag{1}$$

where,

$$NPV(t)_{fix} = \sum_{i=1}^{m} N \cdot R_{fix} \tau(T_i^S, T_i^E) P_d(t, T_i^P), \tag{2}$$

and

$$\begin{aligned}
NPV(t)_{flt} &= \sum_{j=1}^{n} N \cdot \left(F_f(t, T_j^S, T_j^E) + S\right) \tau(T_j^S, T_j^E) P_d(t, T_j^P) & (3)\\
&= \sum_{j=1}^{n} N \cdot \left((\frac{P(t,T_1)}{P(t,T_2)} - 1)/\tau(T_1, T_2) + S \right) \tau(T_i^S, T_i^E) P_d(t, T_j^P) & (4)\\
&= \sum_{j=1}^{n} N \tau(T_i^S, T_i^E) \cdot \left(\frac{1}{\tau(T_i^S, T_i^E)} \frac{P(t,T_1)}{P(t,T_2)} P_d(t, T_j^P) - (\frac{1}{\tau(T_i^S, T_i^E)} + S) P_d(t, T_j^P) \right) & (5)
\end{aligned}$$

where:

- $NPV(t)_{fix}$ and $NPV(t)_{flt}$ are the NPV of the fixed and float legs respectively
- N is the Notional
- R_{fix} is the rate for the fixed leg

2 Deposit

The deposits we are using are not real deposit instruments but rather swap fixings giving us the libor rate directly. The extraction is an exercise in conversion to componding rates.

$$\frac{1}{1+\tau(T^S,T^E)\cdot r_{swapFix}} = P(T^S,T^E) = \frac{P(t,T^E)}{P(t,T^S)} \tag{6}$$

Please observe that the DCF is dependent on the set up in murex, ask G!

3 Futures

The futures require some more effort. Some calendar calculations have to be perfomed. The instrument comes as EDXN where X is the start month either H-March, M-June, U-September or Z-December, N is the last digit of the start year. GBP, USD, EUR and CHF start on the 3rd Wednesday of the month specified. GBP has zero days to spot and hence fixes on the Wednesday, while USD, EUR and CHF has 2 days to spot and fixes on the Monday.

The contract of the future is for 3 months (how about weekends?) and we need to extract the day count fraction corresponding to 3 months.

The start of the future is T^S and the end of the contract period is T^S+3M.

$$\frac{1}{1+\tau(T^S,T^S+3M)\cdot(1-(Price+S)/100)} = P(T^S,T^S+3M) \tag{7}$$

Here S denotes the spread in price due to the convexity adjustment. This should already be included in our reports. It should be noted that for legacy reasons (not having a global bootstrapper) we need the futures periods to completely overlap through extending/compressing the periods to cover the period T^S to the start of the period of the next future T^S_{+1}.

$$P(T^S,T^S_{+1}) = P(T^S,T^S+3M)^{\frac{\tau(T^S,T^S_{+1})}{\tau(T^S,T^S+3M)}} \tag{8}$$

or,

$$\left(\frac{1}{1+\tau(T^S,T^S+3M)\cdot(1-(Price+S)/100)}\right)^{\frac{\tau(T^S,T^S_{+1})}{\tau(T^S,T^S+3M)}} = P(t,T^S_{+1})/P(t,T^S) \tag{9}$$

4 FRAs

Two different kind of discounting methods are implimented,

$$NPV = N \cdot P_d(t,T^P)\frac{\left(F_{strike} - F_f(t,T^S,T^E)\right)\tau(T^S,T^E)}{1+F_f(t,T^S,T^E)\tau(T^S,T^E)}. \tag{10}$$

and

$$NPV = N \cdot P_d(t,T^P)\frac{\left(F_{strike} - F_f(t,T^S,T^E)\right)\tau(T^S,T^E)}{(1+F_f(t,T^S,T^E)\tau(T^S,T^E))(1+F_{strike})\tau(T^S,T^E))}. \tag{11}$$

again we have

$$F_f(t,T^S,T^E) = \left(\frac{P(t,T_S)}{P(t,T_E)} - 1\right)/\tau(T_S,T_E) \tag{12}$$

The first one,

$$NPV/N = P_d(t,T^P)\frac{\left(F_{strike} - \left(\frac{P(t,T^S)}{P(t,T^E)} - 1\right)/\tau_f(T^S,T^E)\right)\tau(T^S,T^E)}{1+\left(\frac{P(t,T^S)}{P(t,T^E)} - 1\right)/\tau_f(T^S,T^E)\tau(T^S,T^E)} \tag{13}$$

$$NPV/N = P_d(t,T^P)\frac{\left(F_{strike} - \left(\frac{P(t,T^S)}{P(t,T^E)} - 1\right)/\tau_f(T^S,T^E)\right)\tau(T^S,T^E)}{1+\left(\frac{P(t,T^S)}{P(t,T^E)} - 1\right)/\tau_f(T^S,T^E)\tau(T^S,T^E)} \tag{14}$$

and continue, I leve this as a weekend home exercise for Dibyendu!!!

5 OIS

$$NPV(t)_{OIS} = \sum_{j=1}^{n} N \cdot \left(F_{OIS}(t,T_j^S,T_j^E) + S\right)\tau(T_j^S,T_j^E)P_{OIS}(t,T_j^P) \tag{15}$$

Should follow the standard vanilla.

5.1 Object representation of price

5.1.1 factor

5.1.2 exp

Inherits: factor
Members: scalar, location Methods: interpolate (return two exp)

5.1.3 const

Inherits: factor
Members: scalar

5.1.4 term

List of <factor>
Methods: interpolate, simplify, linearize, evaluate

Interpolate, go through the list and for every exp on a non pillar replace it with two new exp elements

$$\begin{aligned} exp(s \cdot z(l)) \mapsto exp(s \cdot (k_{left} \cdot z(l_{left}) + k_{right} \cdot z(l_{right}))) = && (16) \\ exp(s \cdot k_{right} \cdot z(l_{left})) \cdot exp(s \cdot k_{right} \cdot z(l_{right})) && (17) \end{aligned}$$

where l_{left} and l_{right} is the location of the pillar to the left and right of the replaced pillar respectivly, the k's are the scalars relating to the interpolation operator.

Linearize, input: the guess of the zero curve, output: a part of a row of the matrix

$$\begin{aligned} term &= \left[\prod s_j\right] \cdot \left[\prod exp(s_i \cdot z(l_i))\right] = && (18) \\ &= \left[\prod s_j\right] \cdot \left[\prod exp(s_i \cdot (z_{guess}(l_i) + \delta(l_i)))\right] && (19) \\ &= \left[\prod s_j\right] \cdot \left[\prod exp(s_i \cdot (z_{guess}(l_i)))\right] \cdot \left[\prod exp(s_i \cdot \delta(l_i))\right] && (20) \\ &\mapsto \left[\prod s_j\right] \cdot \left[\prod exp(s_i \cdot (z_{guess}(l_i)))\right] \cdot \left[(1 + \sum s_i \cdot \delta(l_i))\right] && (21) \end{aligned}$$

Here $[\prod s_j] \cdot [\prod exp(s_i \cdot (z_{guess}(l_i)))] \cdot [(1 + \sum s_i \cdot \delta(l_i))]$ is a part of the matrix row.

Since the terms add up to the NPV, if we take fix terms as negative, we have,

$$NPV(t) = -NPV(t)_{fix} + NPV(t)_{flt} = \sum term_i, \tag{22}$$

Further,

$$\begin{aligned}
\sum_k term_k &= \sum_k \left[\prod s_{j,k}\right] \cdot \left[\prod exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))\right] \cdot & (23)\\
&\cdot \left[(1 + \sum_i s_{i,k} \cdot \delta(l_{i,k}))\right] & (24)\\
&= 0, & (25)
\end{aligned}$$

Which gives,

$$\begin{aligned}
&\sum_k \left[\prod s_{j,k}\right] \cdot \left[\prod exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))\right] \cdot \sum_i s_{i,k} \cdot \delta(l_{i,k})) = & (26)\\
&= -\sum_k \left[\prod s_{j,k}\right] \cdot \left[\prod exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))\right] & (27)\\
& & (28)
\end{aligned}$$

Turning this into the matrix and vector give us,

$$\begin{aligned}
v_{instrument} = -\sum_k \left[\prod s_{j,k}\right] \cdot \left[\prod exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))\right] & \quad (29)\\
& \quad (30)
\end{aligned}$$

And for the j'th column of the matrix,

$$\begin{aligned}
M_{instrument,j} = \sum_k \left[\prod s_{j,k}\right] \cdot \left[\prod exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))\right] \cdot \sum_i s_{i,k} \cdot \boldsymbol{\delta}(l_{i,k}, j)) & \quad (31)\\
& \quad (32)
\end{aligned}$$

where $\boldsymbol{\delta}(i, j)$ is the Kronecker delta, 0 if $i \neq j$ and 1 if $i = j$.
Translating this into pseudocode, with no performance conciderations.
Set matrix row corresponding to the instrument to zero
for(k in all terms)
for(j in all curve locations)

calculate $tmp = [\prod_i s_{i,k}] \cdot [\prod_i exp(s_{i,k} \cdot (z_{guess}(l_{i,k})))]$
for(l in all factors in $term_k$ with factor with a location of j, if any)
$M_{instrument,j} + = tmp \cdot s_l$
end for
end for
end for

5.1.5 price

List of <term>
Methods: interpolate, simplify, linearize

Linearize, input: the guess of the zero curve, output: a full row of the matrix

$$price = \sum term_k \tag{33}$$
$$\mapsto \sum_k \left\{ \left[\prod s_j\right] \cdot \left[\prod exp(s_i \cdot (z_{guess}(l_i)))\right] \cdot \left[(1 + \sum s_i \cdot \delta(l_i))\right] \right\} \tag{34}$$

5.1.6 curveConstituents

List of <price>
Methods: interpolate, simplify, linearize

Linearize, input: the guess of the zero curve, output: a matrix and vector describing the linear problem

The full matrix row and constant (in vector) is calculated in a couple of loops. The number of columns in the matrix is equal to the number of unique pillars points (could be over several curves) and the number of rows is the number of instruments included in the bootstraping (could be different (more) than the number of columns. The vector of constants has the same length as the number of rows of the matrix.

First, set all elements in the matrix, M and vector to zero,v. Loop through all instruments, each instrument corresponds to a row in the matrix. Loop through all cash flows (term object) (both fixed and floating). For each cashflow (term object) $tmp = [\prod s_j] \cdot [\prod exp(s_i \cdot (z_{guess}(l_i)))]$ is first calculated. Then each matrix element in column corresponding to l_i and in the row is corresponding to the trade updated according to $+ = tmp \cdot s_i$. An equivalent update will be made to the vector as $+ = tmp$.

5.2 Solving the linear problem

The resulting matrix and vector is then to be solved using some standard methodology. The solution is then for the update δ to the initial guess. Our subsequent guess is then $z_{guess} + \delta$. We then repeat this until the δ is small enough.

In the case of square matrix a simple inversion is ok. If the number of rows is higher than the number of columns we have an over determined system and some standard technique such as QR decomposition.

6 Examples

The fixed cashflow $N \cdot R_{fix}\tau(T_i^S, T_i^E)P_d(t, T_i^P)$ turns into one term object as; constant object $s = N \cdot R_{fix}\tau(T_i^S, T_i^E)$ and one exp object with constant $-T$ and location P_i on the discounting curve.

The floating cashflow turns into two separate terms. The first term consists of one constant object and three exp objects, the second term consist of one constant object and one exp object.

The first term object has the following constant $N\tau(T_i^S, T_i^E)/\tau(T_i^S, T_i^E)$ with the following exp objects

	const	location
1^{st}	$-T^S$	forward curve at T^S
2^{nd}	T^E	forward curve at T^E
3^{rd}	$-T^P$	discount curve at T^P

The second term object has the following constant $N\tau(T_i^S, T_i^E)(1/\tau(T_i^S, T_i^E) + S)$ with the following exp object

	const	location
1^{st}	$-T^P$	discount curve at T^P

7 A two year Swap

Let's do a simple example for a 2 year swap with two cashflows at one and at two years.

$$
\begin{aligned}
NPV &= NPV_{fix} + NPV_{flt} && (35)\\
&= \sum_{i=1}^{m} N \cdot R_{fix}\tau(T_i^S, T_i^E)P_d(t, T_i^P) + && (36)\\
&+ \sum_{j=1}^{n} N\tau(T_i^S, T_i^E) \cdot \left(\frac{1}{\tau(T_i^S, T_i^E)} \frac{P(t, T_1)}{P(t, T_2)} P_d(t, T_j^P) - (\frac{1}{\tau(T_i^S, T_i^E)} + S)P_d(t, T_j^P) \right) && (37)\\
& && (38)
\end{aligned}
$$

$$
\begin{aligned}
\frac{NPV}{N} &= R_{fix}(1 * P_d(1Y) + P_d(2Y)) + && (39)\\
&+ \left(\frac{P(0Y)}{P(1Y)} P_d(1Y) - (1 + S)P_d(1Y) \right) + && (40)\\
&+ \left(\frac{P(1Y)}{P(2Y)} P_d(2Y) - (1 + S)P_d(2Y) \right), && (41)
\end{aligned}
$$

Skipping the spread

$$
\begin{aligned}
\frac{NPV}{N} &= R_{fix}P_d(1Y) + R_{fix}P_d(2Y)) + && (42)\\
&+\frac{P(0Y)}{P(1Y)} P_d(1Y) + && (43)\\
P_d(1Y) + & && (44)\\
&+\frac{P(1Y)}{P(2Y)} P_d(2Y) - P_d(2Y), && (45)
\end{aligned}
$$

$$
\begin{aligned}
\frac{NPV}{N} &= R_{fix} \cdot \exp(-z_d(1Y)) + && (46)\\
&+R_{fix} \cdot \exp(-2z_d(2Y)) + && (47)\\
&+ \exp(z(1Y)) \cdot \exp(z_d(1Y)) + && (48)\\
&- \exp(-z_d(1Y)) + && (49)\\
&+ \exp(-z(1Y)) \cdot \exp(2z(2Y)) \cdot \exp(-2z_d(2Y)) + && (50)\\
&- \exp(-2z_d(2Y)), && (51)
\end{aligned}
$$

$$
\begin{aligned}
\frac{NPV}{N} &= R_{fix} \cdot \exp(-g_d(1Y)) \exp(-\delta_d(1Y)) + && (52)\\
&+R_{fix} \cdot \exp(-2g_d(2Y)) \exp(-2\delta_d(2Y)) + && (53)\\
&+ \exp(g(1Y)) \exp(\delta(1Y)) \cdot \exp(g_d(1Y)) \exp(\delta_d(1Y)) + && (54)\\
&- \exp(-g_d(1Y)) \exp(-\delta_d(1Y)) + && (55)\\
&+ \exp(-g(1Y)) \exp(-\delta(1Y)) \cdot \exp(2g(2Y)) \exp(2\delta(2Y)) \cdot && (56)\\
&\cdot \exp(-2g_d(2Y)) \exp(-2\delta_d(2Y)) + && (57)\\
&- \exp(-2g_d(2Y)) \exp(-2\delta_d(2Y)), && (58)
\end{aligned}
$$

linerize,

$$
\begin{aligned}
\frac{NPV}{N} &= R_{fix} \cdot \exp(-g_d(1Y))(1 - \delta_d(1Y)) + && (59)\\
&+R_{fix} \cdot \exp(-2g_d(2Y))(1 - 2\delta_d(2Y)) + && (60)\\
&+ \exp(g(1Y)) \cdot \exp(g_d(1Y))(1 - \delta(1Y) - \delta_d(1Y)) + && (61)\\
&- \exp(-g_d(1Y))(1 - \delta_d(1Y)) + && (62)\\
&+ \exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (63)\\
&\cdot(1 - \delta(1Y) + 2\delta(2Y) - 2\delta_d(2Y)) + && (64)\\
&- \exp(-2g_d(2Y))(1 - 2\delta_d(2Y)), && (65)
\end{aligned}
$$

Put on matrix form

$$
\begin{aligned}
& \frac{NPV}{N} + && (66)\\
& -R_{fix} \cdot \exp(-g_d(1Y)) + && (67)\\
& -R_{fix} \cdot \exp(-2g_d(2Y)) + && (68)\\
& -\exp(g(1Y)) \cdot \exp(g_d(1Y)) + && (69)\\
& +\exp(-g_d(1Y)) + && (70)\\
& -\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) + && (71)\\
& +\exp(-2g_d(2Y)) && (72)\\
= \quad & R_{fix} \cdot \exp(-g_d(1Y))(-\delta_d(1Y)) + && (73)\\
& +R_{fix} \cdot 2 \cdot \exp(-2g_d(2Y))(-2\delta_d(2Y)) + && (74)\\
& +\exp(g(1Y)) \cdot \exp(g_d(1Y))(-\delta(1Y)) + && (75)\\
& +\exp(g(1Y)) \cdot \exp(g_d(1Y))(-\delta_d(1Y)) + && (76)\\
& -\exp(-g_d(1Y))(-\delta_d(1Y)) + && (77)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (78)\\
& \cdot(-\delta(1Y)) + && (79)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (80)\\
& \cdot(2\delta(2Y)) + && (81)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (82)\\
& \cdot(-2\delta_d(2Y)) + && (83)\\
& -\exp(-2g_d(2Y))(-2\delta_d(2Y)), && (84)
\end{aligned}
$$

$$
\begin{aligned}
& \frac{NPV}{N} + && (85)\\
& -R_{fix} \cdot \exp(-g_d(1Y)) + && (86)\\
& -R_{fix} \cdot \exp(-2g_d(2Y)) + && (87)\\
& -\exp(g(1Y)) \cdot \exp(g_d(1Y)) + && (88)\\
& +\exp(-g_d(1Y)) + && (89)\\
& -\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) + && (90)\\
& +\exp(-2g_d(2Y)) && (91)\\
= \;& R_{fix} \cdot \exp(-g_d(1Y))(-\delta_d(1Y)) + && (92)\\
& +\exp(g(1Y)) \cdot \exp(g_d(1Y))(-\delta_d(1Y)) + && (93)\\
& -\exp(-g_d(1Y))(-\delta_d(1Y)) + && (94)\\
& +R_{fix} \cdot 2 \cdot \exp(-2g_d(2Y))(-2\delta_d(2Y)) + && (95)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (96)\\
& \cdot(-2\delta_d(2Y)) + && (97)\\
& -\exp(-2g_d(2Y))(-2\delta_d(2Y)) + && (98)\\
& +\exp(g(1Y)) \cdot \exp(g_d(1Y))(-\delta(1Y)) + && (99)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (100)\\
& \cdot(-\delta(1Y)) + && (101)\\
& +\exp(-g(1Y)) \cdot \exp(2g(2Y)) \cdot \exp(-2g_d(2Y)) \cdot && (102)\\
& \cdot(2\delta(2Y)) && (103)
\end{aligned}
$$

8 Overall algorithm

Input: instruments over a number of curves, initial guess

Output: bootstrapped curves

Process; get the price object of all instruments into a vector (the curveConstituents object). Interpolate every element in the vector. Loop over call the get linear method of the curveConstituents object with the initial guess. Recive the matrix and vector, solve linear problem, update the guess. End loop

THE END!