Commodity Simulation Translation

The goal of this work is to translate or convert simulated risk factors into market observables. There are three translations in commodity framework: price translation, volatility translation, and all-in translation.

Market instruments may be daily, for instance, but simulated risk factors may be monthly. Therefore, we need to convert monthly simulated risk factors into daily market instruments. Here we use linear interpolation within the range of risk factors and extrapolation beyond the range.

Let $\{(T, X) \mid (t_1, x_1), (t_2, x_2), \cdots (t_n, x_n)\}$ denotes a series of consecutive risk factor pairs, and $\{(T', X') \mid (t'_1, x'_1), (t'_2, x'_2), \cdots (t'_n, x'_n)\}$ is the corresponding instruments pairs. Given $\{(T, X)\}$ and time $\{T'\}$, the market instrument $\{X'\}$ can be obtained using the following method.

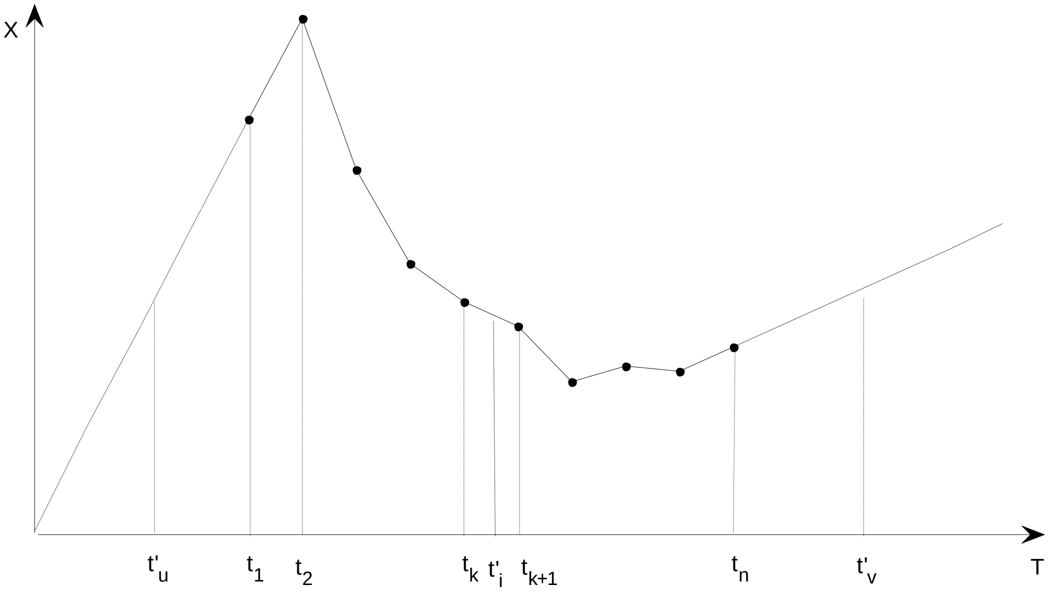

Figure 1. Schematic of linear interpolation

As shown in Figure1, three different cases should be considered:

1. Instrument level closing dates t'_i is within the range of $[t_1, t_n]$.
2. Instrument level closing dates t'_u is ahead of the starting time of risk factor dates t_1.
3. Instrument level closing dates t'_v is behind of the end time of risk factor dates t_n.

In case 1, when the closing date t'_i of instrument level is within $[t_1, t_n]$, linear interpolation is applied (solid lines in the figure). $[t_1, t_n]$ is divided into several sub-range $[t_1, t_2], \cdots [t_{n-1}, t_n]$ based on risk factor pairs. The specific sub-range $[t_k, t_{k+1}]$ in which t'_i locates is found, and the corresponding x'_i is given by:

$$x'_i = x_k + (t'_i - t_k)\frac{x_{k+1} - x_k}{t_{k+1} - t_k} \qquad \text{(interpolation)} \qquad (1)$$

In case 2, when closing date $t'_u < t_1$, extrapolation is performed based upon $\{(t_1, x_1), (t_2, x_2)\}$ (dash-dot line in the figure). The corresponding x'_u is given by:

$$x'_u = x_1 + (t'_u - t_1)\frac{x_2 - x_1}{t_2 - t_1} \qquad \text{(extrapolation)} \qquad (2)$$

In case 3, when closing date $t'_v > t_n$, extrapolation is performed based upon $\{(t_{n-1}, x_{n-1}), (t_n, x_n)\}$ (dash line in the figure). The corresponding x'_v is given by:

$$x'_v = x_{n-1} + (t'_v - t_{n-1})\frac{x_n - x_{n-1}}{t_n - t_{n-1}} \qquad \text{(extrapolation)} \qquad (3)$$

There are 7 risk factor quick deltas (QDs) defined by 1, 10, 25, 50, 75, 90 and 99, whereas the market instruments include 11 QDs: 1, 10, 20, 30, 40, 50, 60, 70, 80, 90 and 99.

Since volatility is a surface, two-dimensional linear interpolation is required. We first do linear interpolation in quick delta (QD) direction converting 7QD to 11QD and then perform another linear interpolation in time direction. As seen in Figure 2, the cross-points of the grids are the risk factors $(t_i, QD_j x_{i,j})$. The star dots denote the market instrument points whose volatility values are expected. The range of 7QDs is [1,99], which is the same as 11 QDs. This denotes no extrapolation but interpolation required in QD directions. However, extrapolation may be required in time direction. Similar to Section 2.1, three cases are considered. Assuming risk factors are $\{(T,QD,X)\,|\,(t_{1,1},QD_{1,1}x_1),\cdots(t_{n,7},QD_{n,7}x_{n,7})\}$, and the instrument volatilities $\{(X)\}$ of $\{(T',QD',X')\,|\,(t'_{1,1},QD'_{1,1}\,x_1),\cdots(t'_{n,11},QD'_{n,11}\,x_{n,11})\}$ will be calculated based upon $\{(T,QD,X)\}$ and $\{(T',QD')\}$. Again, three cases should be considered:

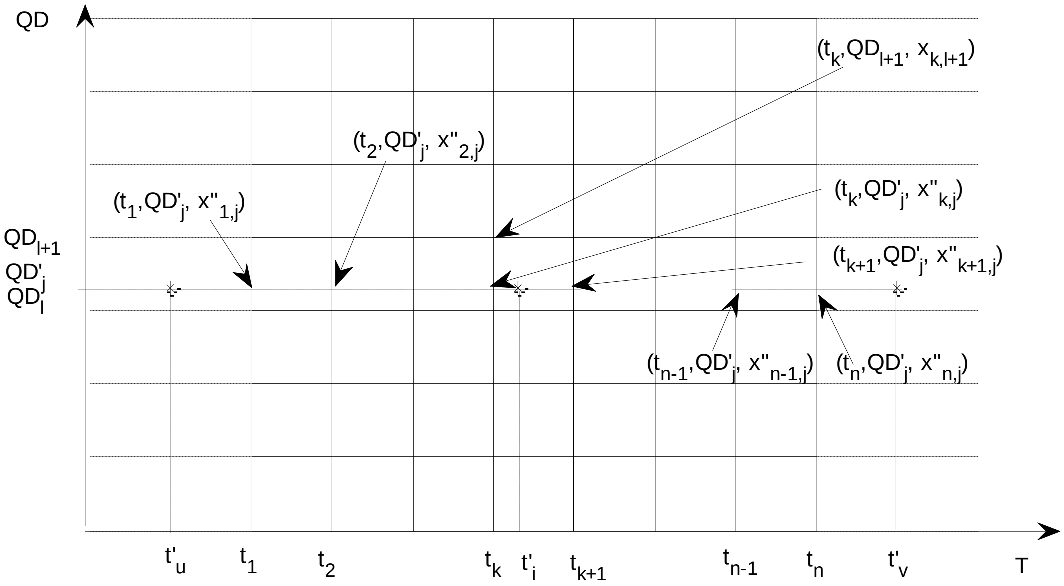
QD
QD_{l+1}
QD'_j
QD_l
$(t_k, QD_{l+1}, x_{k,l+1})$
$(t_2, QD'_j, x''_{2,j})$
$(t_1, QD'_j, x''_{1,j})$
$(t_k, QD'_j, x''_{k,j})$
$(t_{k+1}, QD'_j, x''_{k+1,j})$
$(t_{n-1}, QD'_j, x''_{n-1,j})$
$(t_n, QD'_j, x''_{n,j})$
t'_u
t_1
t_2
t_k
t'_i
t_{k+1}
t_{n-1}
t_n
t'_v
T

Figure 2. Schematic of 2D linear interpolation

Case 1, point $(t'_i, QD'_j, x'_{i,j})$ is in the range of $[t_1, t_n]$ and $[QD_1, QD_7]$, linear interpolation is applied to get the $x''_{i,j}$ of $(t_k, QD'_j, x''_{k,j})$ using points $(t_k, QD_l, x_{k,l})$ and $(t_k, QD_{l+1}, x_{k,l+1})$, and $x''_{i,j}$ of $(t_{k+1}, QD'_j, x''_{k+1,j})$ using points $(t_{k+1}, QD_l, x_{k,l})$ and $(t_{k+1}, QD_{l+1}, x_{k+1,l+1})$. The linear interpolation is again applied to get $x'_{i,j}$ of $(t'_i, QD'_j, x'_{i,j})$ using points $(t_k, QD'_j, x''_{k,j})$ and $(t_{k+1}, QD'_j, x''_{k+1,j})$.

Case 2, if t'_u in $(t'_u, QD'_j, x''_{u,j})$ is less than t_1, interpolation is applied to get $x''_{1,j}$ of $(t_1, QD'_j, x''_{1,j})$ and $x''_{2,j}$ of $(t_2, QD'_j, x''_{2,j})$, extrapolation is performed to get $x'_{u,j}$ of $(t'_u, QD'_j, x'_{u,j})$.

Case 3, if t'_v in $(t'_v, QD'_j, x'_{v,j})$ is greater than t_n. Similar to as case 2, interpolation is performed to get $(t_{n-1}, QD'_j, x''_{n-1,j})$ and $(t_n, QD'_j, x''_{n,j})$, followed by the extrapolation on them to get $(t'_v, QD'_j, x'_{v,j})$.

In order to capture cross Gamma for pipelines that have spreads over bases, we need construct all-in scenarios. In our system, we simulate relative base and absolute spread.

Let x denote base, y denote spread and z denote all-in. The base RF simulation result is given by

$$A = \frac{x_t - x_0}{x_0} \tag{3}$$

The spread RF simulation result is given by

$$B = y_t - y_0 \tag{4}$$

The all-in scenario in absolute return form is

$$z_t - z_0 = (y_t - y_0) + (x_t - x_0) = B + Ax_0 \qquad (5)$$

where A is simulated base, B is simulated spread, x_0 is the closing rate for base.

References:

https://finpricing.com/lib/IrBasisCurve.html