

The decentralisation trap

The FRTB standardised approach

The penny is starting to drop, says Steve Davis, head of design at Vector Risk. The standardised approach may seem simple enough, but front-office tools lack key refinements and will require continuous attention to ensure complete capture of sensitivities

Lacking the high-performance computing requirements and profit-and-loss (P&L) controversy of the FRTB internal model approach (IMA), it has been assumed that the standardised approach (SA) will be simple to implement. After all, even small banks have to run it. Common wisdom has been that delta, vega and curvature sensitivities will flow from front-office systems into an off-the-shelf calculator that will apply the regulatory risk weights and correlations. However, the penny is finally starting to drop.

FRTB sensitivity generation requires a great deal of supporting logic to ensure trades are allocated correctly. Credit spread, equity and commodity risk factor sensitivities must be labelled with Basel-defined credit quality, industry sector, market capitalisation, economy or commodity group. A large number of definitions have to be maintained in each system that generates sensitivities. Some systems will not support the large parallel shifts for the curvature sensitivities, and some front-office pricing is still carried out in spreadsheets. Drill-down tools are needed to investigate results. A snapshot of rates must be kept in case reruns are required. Most importantly, there should be a mechanism to guarantee that all of the risk has been captured.

A decentralised solution is likely to be mixed in its support for these crucial functions, so does a viable alternative exist?

Many risk engines already have sophisticated stress-testing capabilities that bring a regular taxonomy to describing sensitivities as bumps on curves across all markets. This includes basis point shifts on rates, percentage shifts on prices and volatilities, and up/down parallel shifts for curvature. Risk management staff have ready access to results, drill-down and reruns. The pricing is very fast, so shortcuts like adjoint algorithmic adjustment are not required. The sensitivity outputs, along with default and residual risk raw data, become the inputs to the relatively simple final step that applies the Basel risk weights and correlations to produce the capital number.

Traditional versus risk factor-driven sensitivities

Compare these two approaches to defining sensitivities: in the 'traditional front-office' model, a sensitivity is explicitly defined for each risk factor, with a description that helps align it to the appropriate FRTB bucket. If a risk factor is missing from the list of sensitivity definitions, no error is recorded and the risk is not captured. Auditing for completeness becomes a manual process and requires constant attention.



Steve Davis

In the 'risk-factor driven' model, sensitivity definitions contain wild cards so one definition can match all risk factors of a given type, with a secondary match on FRTB bucket. This secondary match is only possible if the FRTB bucket is also recorded as part of the risk factor definition in the market data store. Now users only need one sensitivity definition per FRTB bucket. New risk factors are automatically included as the bank trades them, and it can be guaranteed that every risk factor on every trade will generate a sensitivity. If your market data has a risk factor with an unassigned FRTB mapping, a sensitivity will still be calculated, and routed to the FRTB 'other' bucket for that risk type where it will attract the highest capital.

The 'risk-factor driven' model is far more elegant and auditable than the traditional front-office approach because all the FRTB logic is centralised and minimised. The bank's regulator can have confidence that the bank is capturing all of its sensitivities and not underestimating capital. The bank itself has a mechanism to maintain the quality of its FRTB mappings simply by checking which risk factors end up being allocated to each FRTB bucket.

Multi-tenancy risk engines are becoming available on the cloud with standardised application programming interfaces for loading trades and market data. With no installation on the client side, they can be plugged in to fill a specific requirement, such as standardised model sensitivity generation, or further utilised to satisfy the full FRTB requirement with little or no disruption to the monolithic front-office systems supplying the data.

A final consideration is that a decentralised approach is a dead end, leaving no natural pathway to the more capital-efficient internal model approach. It is paradoxical that small banks trading in a single location – often in liquid markets with vanilla instruments and low volumes – can avoid the worst of the performance, P&L attribution and non-modellable risk factor issues faced by large banks. Throw in the low-cost, shared infrastructure of a cloud software-as-a-service and growing support from market rate vendors for shared historical data sets, and suddenly the internal model doesn't look so daunting after all.

Contact

Steve Davis • Head of Design, Cloud Software
E steven.davis@vectorrisk.com
www.vectorrisk.com