Turning financial regulations into practical algorithms

Nowadays, regulations are becoming more and more complex in the financial industry. **That is why technology comes to play an important role within the compliance of these regulations**, creating practical solutions to facilitate the process of calculation and reporting.

This is the case with the new regulation of Fundamental review of the trading books, which forces financial institutions to make drastic changes in their operations and the minimum capital requirements and therefore changes in resources to deal with these changes.

The Fundamental Review of the Trading Book (FRTB) comes after Basel II, which contains all the recommendations on banking laws and regulations issued by the Basel Committee on Banking Supervision (BCBS). After Basel II failed to avoid the subprime crisis, the Basel Committee decided to add and change rules, and it resulted in this new Basel Committee framework for the next generation market risk regulatory rules.

Thus, the FRTB main goal is to highlight the shortcomings of the current Basel 2.5 market risk capital framework. In that respect, the FRTB aims to review risks valorisation and the capital requirement rules.

My work at **Fintops** as a fintech engineer is to make these formulas easy to process through our Prisme solution for calculation automation for this law.

Here, we will focus on the minimal capital requirement value, which will change once the FRTB will be effective in 2022.

To compute the value of minimal capital requirement, one needs to compute the total risk charge which consists in summing up sensitivity-based risk charge with the default risk charge and the residual add-on.

Due to the FRTB, the standardized approach will be modified. There are 3 mains risk changes. The first one is the calculation procedure of the default risk charges. The second one is the residual risk add-on and the last one is the sensitivity based risk charge.

Here, we will focus on the new sensitivity based risk charge, and more precisely, on the curvature risk charge, because it's the most complex part of the implementation.

According to the Basel Committee on Banking Supervision, the curvature risk capital requirements are computed by calculating the maximum loss of two scenarios of shocks – an upward shock and a downward shock. Banks revalue their non-linear instruments based on those shocks, and calculate the incremental value change beyond what would be estimated using sensitivities. This incremental amount is the additional capital requirement for curvature risk.

The curvature risk charge is the last component of the sensitivity based approach for calculating market risk capital using standardised approach. The curvature risk capital requirements are computed by calculating the maximum loss of two different scenarios of shocks — an upward shock and a downward shock.

$$\textit{Curvature risk} = \sqrt{\max\left(0, \sum_b K_b^2 + \sum_b \sum_{c \neq b} \gamma_{bc} S_b S_c \psi(S_b, S_c)\right)}$$

Formula of the Curvature risk charge, from the Basel Committee on Banking Supervision Consultative Document "Revisions to the minimum capital requirements for market risk"

Kb is the curvature risk exposure;

γ_{bc} is the corresponding prescribed correlations;

Sb and Sc are the sum of all risk factors in bucket b and c;

ψ(Sb,Sc) is equal to 0 if Sb and Sc have both negative sign, and equal to 1 in all other cases;

At first sight, this formula may seem complicated to implement as an algorithm, but with the appropriate tools, it can be used easily. A way to estimate the result of this formula is to calculate the sum of all curvatures risk charges for all risks individually, which is the same process for calculating the net curvature risk charge for one risk, and then to sum all of those results.

In fact, the key of the formula for the individual curvature risk charge is to be able to apply an upshock and a downshock directly to the portfolio, and then compare between those two figures in order to consider only the smallest. The practical formula we obtain is the following one :

$$CurvatureRiskCharge_k = -\min(UpShock, DownShock)$$

$$UpShock = \sum_i \left[V_i\left(x_k^{RW(Up)}\right) - V_i(x_k) - RW_k^{(Base)}.s_{ik}\right]$$

$$DownShock = \sum_i \left[V_i\left(x_k^{RW(Down)}\right) - V_i(x_k) + RW_k^{(Base)}.s_{ik}\right]$$

Other formulation of the Curvature risk charge formula

i represents an instrument for which CVR is calculated for the risk factor *k*;

x_k is the current level of risk factor;

The first Vi is the value of the instrument *i* when the risk factor x_k is applied upward or a downward shock and the second Vi represents the value of instrument i depending on the current level of risk factor k;

RW is the risk weight of the instrument *i* for the risk factor *k;*

s_{ik} is the sum of delta sensitivities.

Those shocks can be estimated and applied with several methods. Usually financial institutions have some position keeping software that holds their portfolio and allows users to value the different instruments it contains given a market data context. FRTB regulation stipulates that upward and downward shocks should be perturbations of risk factor underlying market data by +30 BP and -30 BP (Basis Point, which is one hundredth of a percent).

```python
import ValueTrade  #Using function Valuetrade from the client

def ApplyMarketShift(Trade,ShiftValue):
    riskWeight = ValueTrade(Trade,MarketDataContext)*ShiftValue
    shiftedMarketDataContext = Shift(MarketDataContext,ShitValue)
    result = ValueTrade(Trade,shiftedMarketDataContext) - riskWeight
    return(result)

def CurvatureRiskCharge(Portfolio):
    CurvatureValue
    for trade in portfolio:  #ShiftValue given by the FRTB
        netCurvature = -min(ApplyMarketShift(trade,30),ApplyMarketShift(trade,-30))
        curvatureValue += netCurvature
    return(curvatureValue)
```

Example of an implementation of the previous formula in Python

Thus, thanks to an accurate interpretation of the regulation formula finality and financial purpose, it is quite simply possible to implement a complex formula such as the curvature risk charge one.

About the Author

Walid Maaroufi - Fintech engineer

Walid Maaroufi is a fintech engineer who will graduate in 2020 in the Ecole des Mines de Saint-Etienne. He joined the R&D part of [Fintops] in 2020 with the will to develop innovative products.