Neural network calibration of the DDSVLMM interest rates model, and application to weights calculation

Pierre-Edouard Arrouy,
Paul Bonnefoy,
Alexandre Boumezoued,
Yousra Cherkaoui Tangi,
Grzegorz Darkiewicz,
Sophian Mehalla,

Market-consistent economic scenarios are at the core of the measurement of the Technical Provisions under Solvency II. EIOPA guidelines and the French regulator ACPR, among others, have highlighted the importance of linking the calibration process with the risk profile of liabilities. Such requirement also applies to other regulations.

Weights reflecting the sensitivity of the insurer's liabilities are therefore to be assigned to each market information within the calibration process. This paper presents how neural networks can help deal with the intensive recalibrations required to achieve this objective.

The European Insurance and Occupational Pensions Authority (EIOPA) guidelines state that:

"Insurance and reinsurance undertakings should be able to demonstrate that the choice of financial instruments used in the calibration process is relevant given the characteristics of [their] obligations."[1]

To model interest rates in a risk-neutral environment, a calibration procedure is used to match the observable financial instruments in the market (the reference instruments). It is based on minimising discrepancies between the market prices and the prices implied by the model (so-called model prices), for a selected set of reference instruments. Such calibration process is performed for different risk-neutral models, from the simplest ones (e.g., one- or two-factor short rate models) to the most complex ones, such as the LIBOR Market Model with Stochastic Volatility and Displaced Diffusion (DDSVLMM). The latter is available, among other models, in the Economic Scenarios Generator Milliman CHESS™.[2] This model is considered further in this paper, although the technique is generic and can be applied to other models.

The most commonly used reference instruments are swaptions. The swaption market is rather deep and liquid, and covers instruments with very different characteristics, represented by three main parameters: maturity (time at which the option on swap can be exercised), tenor (period during which interest rate swap cash flows are considered) and strike (which specifies the moneyness of the option). When the strike is equal to the current swap forward rate, the option is said to be at-the-money (ATM); otherwise, we refer to away-from-the-money (AFM).

One of the main drivers of increasing sophistication of risk-neutral interest models is to improve their ability to replicate the so-called volatility cube, i.e., swaptions for different maturities, tenors and strikes. Although the DDSVLMM is considered to be very good at replicating the entire volatility cube, in reality this replication is still not perfect, especially due to a finite number of model parameters as well as pricing approximations involved in the calibration process; hence prices of some instruments are replicated better than the others. Of course, this statement applies even more to simpler models, like one-factor or two-factor short rate models.

[1] EIOPA Guideline 57 on the valuation of technical provisions, calibration process.

[2] More information is available at https://www.milliman.com/en/products/milliman-chess.

For this reason, a refined calibration to swaption prices is required in order to take into account the relative importance of swaptions for valuation. Each swaption is assigned a specific weight w_j in order to reflect the related sensitivity of the insurer's balance sheet. This type of weighting aims to improve the replication of the subsets of market data that are of higher importance with respect to the company's risk profile, as opposed to other subsets.

In general, if we denote the market price of j–th swaption by ${PS_j}^{Market}$ and the price for the same swaption implied by the interest rate model with model parameters Θ by ${PS_j}^{Model}(\Theta)$, the calibration reflecting the selected weighting scheme can be defined as search of the parameter set $\Theta^{(opt)}$ that minimizes, e.g., the following objective function:

$$\sum_j w_j \left({PS_j}^{Market} - {PS_j}^{Model}(\Theta)\right)^2,$$

where the sum applies to all swaptions j considered in the calibration process (typically with different values of maturity, tenor and strike).

Until recently, the choice of weights (w_j) often used to be arbitrary: as an illustrative example, e.g., defining it as 1 for ATM swaptions, as well as for AFM swaptions with a specific tenor, while 0 weight is considered for all other swaptions. However, currently the awareness regarding the importance of reasonable aligning of weights with the risk profile is growing. As described in the next part of this paper, this can be approached with a methodology based on multiple recalibrations of a risk-neutral model, for shocks applied to the swaption surface. Taking into account the need for a number of model recalibrations, we propose an innovative methodology based on neural networks, which allows us to significantly improve the speed of the calibration process.

The weights calibration problem

The weights shall reflect the sensitivity of such indicators as the Own Funds (OF). In this specific exercise, the aim is to measure the sensitivity to swaptions, all other risk drivers being set at their central values (including interest rate level and equity volatilities, among others). To illustrate the problem, suppose that we represent OF as an "optimal linear combination" of N swaptions with prices P_i:

$$OF = \mu + \sum_{i=1}^{N} \beta_i P_i.$$

This representation is made in the form of a replicating portfolio, where only swaptions are involved (in other words, all other possible assets are "hidden" in the formula within the constant term μ).

The reference value of OF estimated in a perfectly market-consistent way would reflect market prices as follows:

$$OF^{Ref} = \mu + \sum_{i=1}^{N} \beta_i P_i^{Market}.$$

Using the model prices P_i^{Model} produced by the DDSVLMM model, we also have the following representation of the modelled OF:

$$OF^{Model} = \mu + \sum_{i=1}^{N} \beta_i P_i^{Model}.$$

The aim is to minimise the discrepancy between the reference and the modelled OF, represented by the absolute difference $\left|OF^{Ref} - OF^{Model}\right|$. From the Cauchy–Schwarz inequality, this replication error is bounded as:

$$\left|OF^{Ref} - OF^{Model}\right| \leq \sqrt{N} \sqrt{\sum_{i=1}^{N} \beta_i^2 (P_i^{Market} - P_i^{Model})^2}.$$

This means that in the calibration process involving a mean square error target, the valuation error can be controlled by setting the following weights:

$$\omega_i = \beta_i^2.$$

The problem, therefore, boils down to retrieving the sensitivity parameters:

$$\beta_i = \frac{\partial OF}{\partial P_i}.$$

Those can be obtained in the following steps:

1. Applying shocks for each swaption price or subset of swaption prices (note that we can instead refer to implied volatilities as needed).
2. Calibrating the interest rate model based on shocked market information.
3. Simulating the risk-neutral model with the updated parameters.
4. Calculating the OF based on the risk-neutral scenarios generated, then compared to the original OF to estimate the sensitivity parameter.

The original approach to calibrate the DDSVLMM model has been proposed by Wu & Zhang (2006),[3] based on the original method from Heston (1993)[4] applied to equity prices. In the recent years, we have developed a range of methods to accelerate the DDSVLMM calibration process, including the use of the so-called Gram-Charlier and Edgeworth expansions,[5] efficient approximations based on polynomial processes,[6] and acceleration of the optimisation procedure based on analytical calculation of the gradient.[7] Those methods have proven to be efficient for applications involving intensive calibrations in a variety of contexts. In this paper, we present another novel approach based on neural networks. These networks have been successfully explored in mathematical finance to learn the model parameters from the information provided by the market. In this context, they also appear as a promising approach to the problem of weight design. Indeed, in this study the computational time has been reduced by a factor of 250 by using the neural network, as will be described below.

Modelling interest rates

Like the classical Libor Market Model, the DDSVLMM belongs to the family of interest rate models which define their dynamics for forward rates observable on the market (as opposed to short rate models like Hull-White or the instantaneous forward rate used in the Heath-Jarrow-Morton framework). Its volatility factor is modelled with a stochastic process, which facilitates replication of some market structures (smile, skew). In addition, its distribution is displaced with a shift to handle negative rates.

MODEL DYNAMICS

We consider a fixed horizon T and a tenor structure with annual dates $T_0 = 0, T_1, \ \dots, T_{N-1}, \ T_N = T$. The forward rate prevailing over the period $[T_k, T_{k+1}]$ is defined by:

$$F_k(t) = \frac{P(t, T_k)}{P(t, T_{k+1})} - 1.$$

For a given displacement factor $\delta > 0$, the displaced forward rate $\tilde{F}_k(t)$ is built from the definition of the forward rate $F_k(t)$ by:

$$\tilde{F}_k(t) = F_k(t) + \delta.$$

---

[3] Wu, L., & Zhang, F. (2006). LIBOR market model with stochastic volatility. Journal of industrial and Management Optimisation 2(2), 199.

[4] Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. The Review of Financial Studies 6(2), 327-343.

[5] Devineau, L., Arrouy, P-E, Bonnefoy, P., & Boumezoued, A. (2020). Fast calibration of the Libor Market Model with stochastic volatility and displaced diffusion. Journal of Industrial & Management Optimisation 16 (4) : 1699-1729.

[6] P.-E. Arrouy, A. Boumezoued, B. Lapeyre, & S. Mehalla (2020). Jacobi Stochastic Volatility Factor for the Libor Market Model. Finance & Stochastics.

[7] Andres, H., Arrouy, P. E., Bonnefoy, P., Boumezoued, A., & Mehalla, S. (2020). Fast calibration of the LIBOR Market Model with stochastic volatility based on analytical gradient. arXiv preprint arXiv:2006.13521.

The dynamics of the (displaced) forward rate under the risk-neutral probability can be derived as follows (see Wu & Zhang, 2006):

$$\mathrm{d}\tilde{F}_k(t) = \sqrt{V_t}\tilde{F}_k(t)\boldsymbol{\gamma}_{\boldsymbol{k}}(t) \cdot \left(\mathrm{d}\boldsymbol{W}_t - \sqrt{V_t}\boldsymbol{\sigma}_{\boldsymbol{k+1}}(t)\mathrm{d}t\right).$$

The components in bold refer to vectors; as such, the model involves multiple factors. The volatility pattern is driven by $\boldsymbol{\gamma}_{\boldsymbol{k}}(t)$ and $\boldsymbol{\sigma}_{\boldsymbol{k+1}}(t)$, which are interrelated based on no-arbitrage relationships. The components $\boldsymbol{\gamma}_{\boldsymbol{k}}(t)$ are based here on the three parameters a, b and c that are to be estimated.

Furthermore, $(V_t)_{t \le T}$ is the stochastic variance process following Cox-Ingersoll-Ross (CIR) dynamics under the risk-neutral measure:

$$\mathrm{d}V_t = \kappa(\theta - V_t)\mathrm{d}t + \epsilon\sqrt{V_t}\mathrm{d}Z_t.$$

Finally, the correlation structure between forward rates and their variance is captured through a correlation parameter ρ to be estimated in the calibration process.

As a summary, the set of parameters to be estimated is: $\Theta = (a, b, c, \theta, \kappa, \epsilon, \rho)$.

Calibration using neural networks

The proposed methodology based on fitting neural network can be summarised in four steps:

1. Training and test database construction: Generate a large sample of DDSVLMM typical parameters within prescribed grids (e.g., using Sobol sequences). Then compute the related prices from the standard pricing formulas of the DDSVLMM.
2. Neural network training on swaptions volatilities: The neural network is trained on 80% of the sample to learn the parameters as a function of the volatilities; it is then tested on the remaining 20% of the sample.
3. Neural network evaluation on test and selection of best neural network model.
4. Use of the neural network for instantaneous calibration in the weight design approach.

SIMULATED DATA

The need to generate simulated data is mainly driven by the huge number of data points required to ensure a satisfactory learning of the neural network. If we compile daily observations of the last 20 years, we have 5,040 (252*20) training surfaces, which is a relatively low number. In the simulation exercise, we generate parameters that reflect a reasonable range of swaption price values; the sample of parameters obtained is presented in Figure 1.

FIGURE 1: BOX PLOT OF THE DDSVLMM PARAMETERS

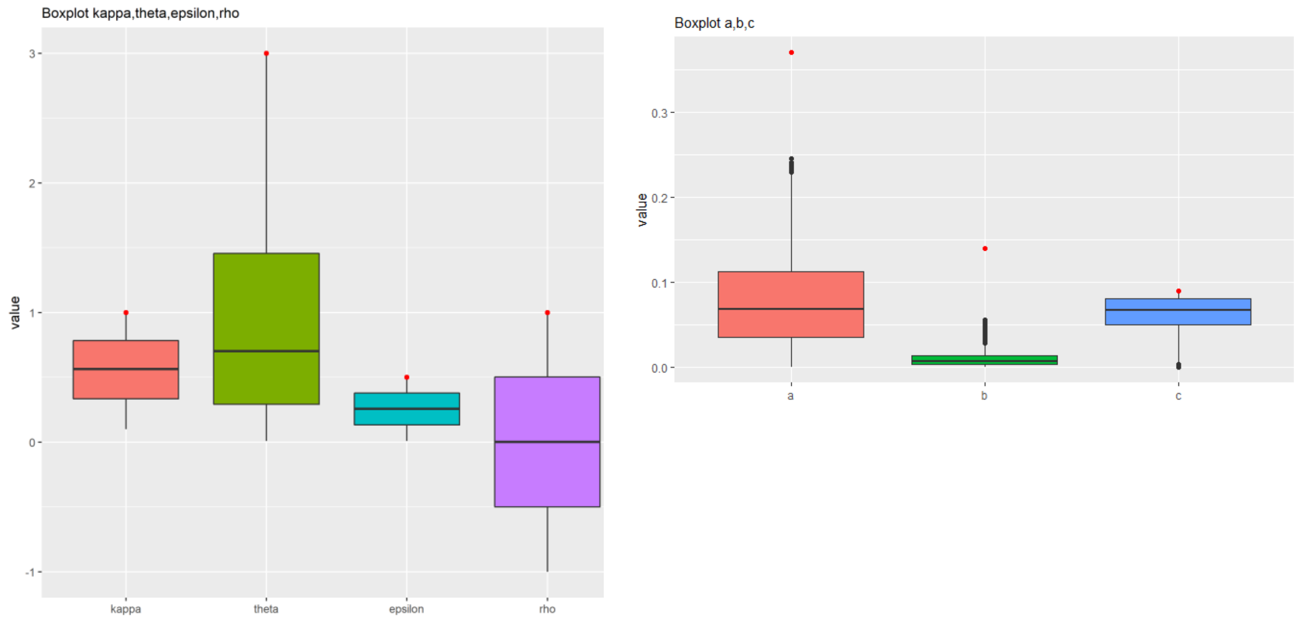


FIGURE 2: NEURAL NETWORK ARCHITECTURE

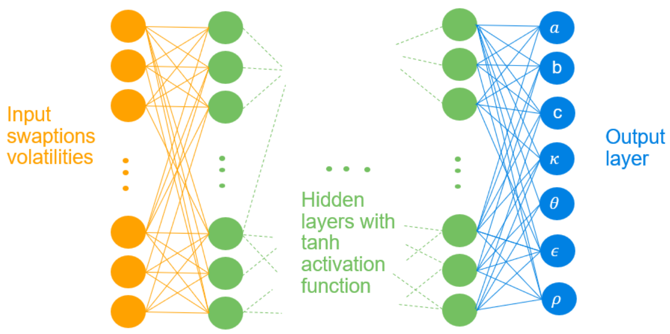
HYPER-PARAMETER	VALUE
NUMBER OF HIDDEN LAYERS	$\{3, \ldots, 11\}$
NEURONS PER LAYER	$\{64, 128, 256\}$
EPOCHS	$\{10, 20, 30 \ldots, 200\}$
BATCH SIZE	$\{8, 16, 32 \ldots, 256\}$
ACTIVATION FUNCTIONS	$\{elu, relu, tanh, softmax, linear\}$
DROPOUT RATE	$\{10\%, 20\%, \ldots, 50\%\}$

Before training the neural network, an optimal architecture must be found. The table in Figure 2 summarises the range of hyper-parameters that are studied and compared:

The number of layers and neurons per layer will drive the complexity of the neural network, for which we want to reach a trade-off between stability and replication accuracy. Epochs and batch sizes are also tuned to optimise the learning process. Different activation functions are tested to monitor the shape of the resulting response function (obtained as a composition of linear operations and the activation functions). Finally, a dropout rate is considered to manage overfitting.

The retained neural network relies on $tanh$ activation functions and is illustrated in Figure 3.

FIGURE 3: CHOSEN NEURAL NETWORK ARCHITECTURE



The table in Figure 4 summarises the different hyper-parameters obtained as a result of the grid search algorithm.

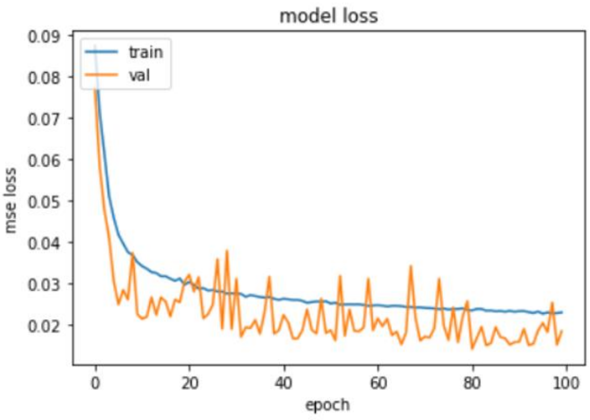
FIGURE 4: RESULTING HYPER-PARAMETERS

HYPER-PARAMETER	VALUE
Number of hidden layers	3
Neurons per layer	64
Epochs	100
Batch size	8
Activation functions	$tanh$
Dropout rate	50%
Early stop	2 epochs with no validation error improvement (2×10^{-5})

NEURAL NETWORKS TRAINING

The neural network is trained using the backpropagation algorithm. Weights and biases of the algorithm are updated using the stochastic gradient minimisation. The learning curve obtained is presented in Figure 5.

FIGURE 5: LEARNING CURVE



CALIBRATION RESULTS

Because the neural network outputs the DDSVLMM parameters, we start testing the goodness of fit of the neural network by evaluating the error on the outputted parameters. In Figure 6, we depict the values of the output parameters as a function of 100 swaption surfaces (here randomly selected due to visualisation considerations), among the 20,000 surfaces from the test data:

- In blue, the predicted values by the neural network
- In orange, the test data

The smallest errors are observed for the parameters a, b and c. The volatility surfaces are indeed very sensitive to the variation of these three parameters.

FIGURE 6: PREDICTION RESULTS ON TEST DATA PARAMETERS

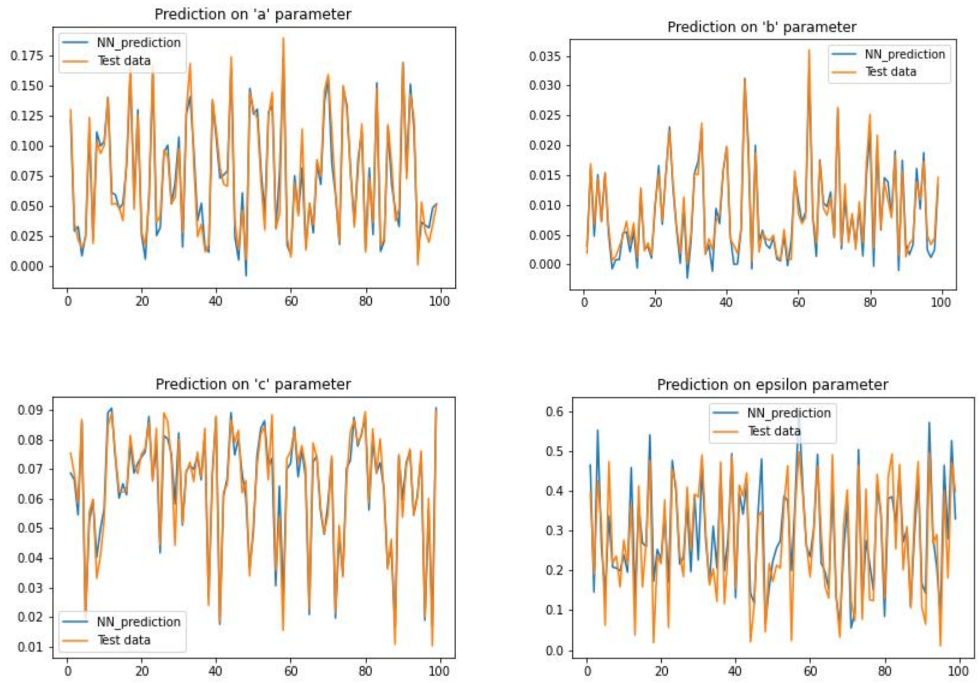
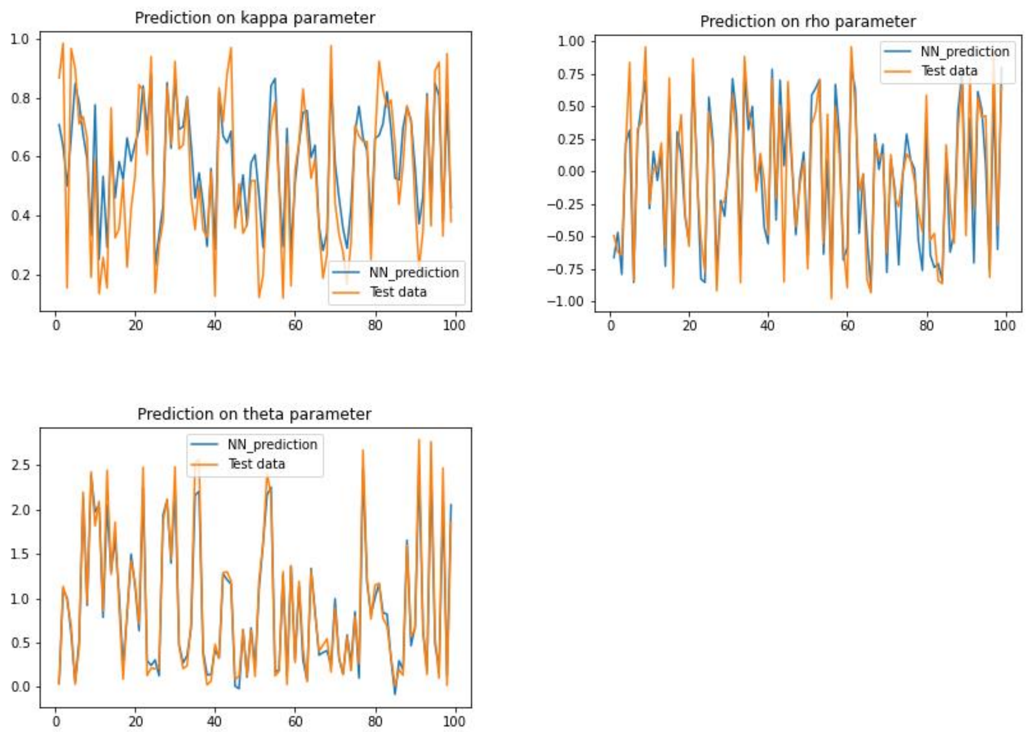


FIGURE 6: PREDICTION RESULTS ON TEST DATA PARAMETERS (CONTINUED)



The neural network can then be used to infer parameters based on market implied volatility quotes of swaptions as of 30 June2020. The parameters can then be converted into model swaption volatilities. The comparison between parameters obtained from the neural network and those obtained from the classical calibration approach[8] is depicted in the table in Figure 7.

FIGURE 7: PARAMETER COMPARISONS

PARAMETER	NEURAL NETWORK	CLASSICAL CALIBRATION
a	0.0072	0.0053
b	0.008	0.030
c	0.058	0.094
κ	0.044	0.089
θ	0.41	0.27
ϵ	0.23	0.21
ρ	0.87	0.062

Some parameters are close, while others appear to differ. In order to provide an assessment of the calibration accuracy by the neural network, we measure the difference between the model volatilities for the neural network calibration and the market volatilities. The average absolute error on ATM swaption volatilities is 12 basis points (bps). This is to be compared to the average error of 5 bps between model volatilities from classical calibration and market volatilities. As could be expected, the neural network provides a higher error magnitude. Recall, however, that this provides, once fitted on the training data, an instantaneous parameter calibration (no computational cost), which is here the objective.

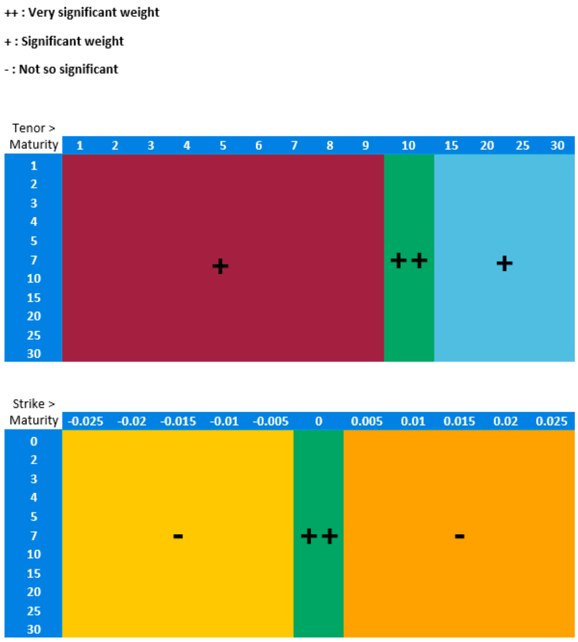
[8] Note that the parameters depicted are for purely illustration purposes and do not reflect Milliman CHESS calibration standards and any related recommendation for use in practice.

Weights design results

Shocks are applied to the swaption volatilities to determine which swaption areas induce significant OF variations, in line with the weight design methodology described above. For each shock, a new swaption volatility surface is built, hence a new calibration of the DDSVLMM model is required. In this study, this is achieved through the use of the neural network, which makes the weight design approach faster as opposed to using the classical calibration method. In this study the computational time has been reduced by a factor of 250 by using the neural network.

A preliminary step in the approach is to group swaption volatilities according to distinct areas that remain interpretable in line with company's risk profile. Those volatilities relate in particular to maturity and tenor distribution, and to optionality (positive or negative strikes in the AFM area). Then the shocks are performed and the OF sensitivity is measured accordingly for each area. The OFs are calculated based on a simplified ALM model involving classical savings contracts. The results of the weight design approach are depicted in Figure 8, where the different areas are displayed for both ATM and AFM swaptions and highlighted in different colours, and where the weight values are disclosed according to their magnitudes on a scale $\{-;+;++\}$.

FIGURE 8: AREAS WEIGHTS RESULTS



As we see in Figure 8, different weights have been designed to areas depending on maturity, tenor and strike. The weight design process exhibits in particular the relative importance of ATM market information as opposed to AFM market data. Interestingly, the 10-year tenor plays a central role as it conveys the highest weights. This seems interpretable regarding the way the underlying cash flow model works, because it involves the 10-year rate in three main mechanics: on the liability side, the credited rate calculation and dynamic lapses trigerring function; on the asset side, it is the reference maturity for reinvestments in bonds.

Finally, we can compare the valuation of the OF based on a classical calibration in two cases:

- Considering uniform weights, i.e., without paying attention to the optimal weight design and OF heterogenous sensitivity to subsets of market swaption volatilities
- Considering the optimal weights as considered in this study

The relative variation of the OF in the two cases has been proven to be nonnegligible in this study, as it can also be the case in a variety of contexts (depending on liability portfolio structure and economic conditions). In particular, the materiality of this impact can increase when the interest rate model replication capabilities are lowered or when the weight distribution becomes even more heterogenous.

Concluding remarks

We presented a calibration technique of the DDSVLMM relying on neural networks. The particular application we considered was the design of swaption weights in the calibration process. The use of neural networks in a context of such exercises, which require multiple recalibrations, is promising as it significantly reduces computational time, in this case up to a factor of 250.

Of course, calibration accuracy is to be monitored because the neural network can be considered only as a proxy of the full calibration result, by overcoming the minimisation of the discrepancies between market data and classical model pricing. In this study, we have seen how an optimal weight design allows us to retrieve a better assessment of risk-neutral valuation.

Other applications may be explored using such a faster calibration technique, for instance Nested Simulations as well as for calibration of any proxy methods for Solvency II Internal Models or the forecast of solvency ratio in a Standard Formula framework.

How can Milliman help?

Milliman has a depth of experience and expertise related to Economic Scenarios Generators (ESGs), including the following services:

- Milliman CHESS™. We offer a “one-size-fits-all” ESG software-as-a-service (SaaS) solution, with a large spectrum of best-in-class models. We provide advice on the best model in each context and the relevant tailored calibrations in line with each specific risk profile.
- Delivery of economic scenarios. We provide scenarios as a service, including custom calibrations and scenario generation at required frequencies
- Production support. Secure the process by outsourcing production work, with the assistance of our teams in the generation of economic scenarios during production campaigns.
- ESG review. Get updates and comfort on your in-house solution through a detailed external review of the modelling framework, including embedded parameters, implementation and validation of the results.
- ESG implementation support. We help you move to an in-house ESG implementation and get the relevant support: thorough choice of models and assumptions, technical specifications, functional specifications and core code implementation.

If you have any questions or comments on this paper or any other aspect of Economic Scenario Generators, please contact your local Milliman consultants or use the contact link below.

CONTACT
chess@milliman.com