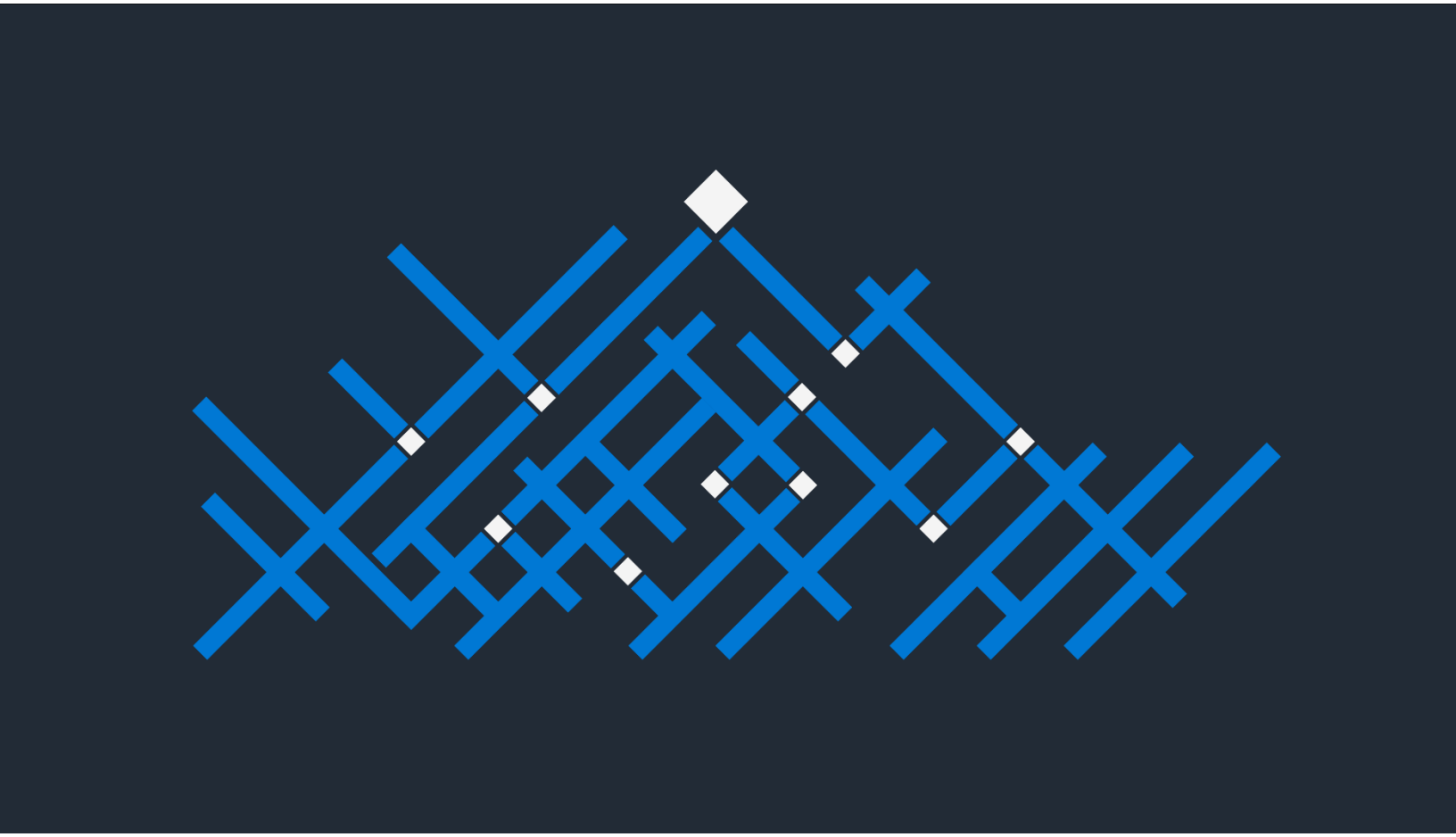
MILLIMAN REPORT

Clustering techniques: Innovations and practical implementation

October 2025

Aatman Dattani
Avi Freedman
Corey Grigg
Jan Thiemen Postema
Karol Maciejewski
Sijbo Holtman



Milliman

Table of contents

In today's evolving insurance landscape, market consolidations and increasingly sophisticated models are rapidly increasing portfolio sizes and computational demands. As regulatory reporting, risk management, and capital management all require timely and accurate model runs, both runtime efficiency and information availability have become critical priorities. This paper explores novel and evolving clustering techniques for model compression, offering significant computational efficiency gains and providing practical solutions to these pressing industry challenges.

Introduction

Part of the job of life insurance actuaries involves performing financial projections on large blocks of contracts. For some projections where it is too time consuming to run a seriatim projection (where each contract is projected separately), model cells are used. The extent to which model cells are used varies, e.g., by region and use case, but the main goal is to approximately reproduce relevant cash flows and other values. For large blocks of business, this approach can reduce runtimes and computational cost and time by a factor of hundreds or more, clearly demonstrating its value to life insurers.

The traditional method for creating model cells is roughly as follows: Contracts are combined into one cell if they match on certain characteristics (e.g., plan code) and also are in the same range for others (e.g., issue age). The input file is created by using one representative cell for each such group of contracts, which uses the characteristics common to all of the contracts in the group (or a representative value if the characteristics vary) and otherwise uses input values that in some way represent an average of the contracts in the group. The cell is also grossed up by a scalar so that it can stand in for the entire group.

This procedure is reasonable for many types of business. However, problems arose in the U.S. during the early 2000s for variable annuities. Because of the large number of scenarios required to assess these products, it may be necessary to use cell models (the same thing can happen for any product where large numbers of sensitivities, rather than scenarios, are to be run). However, largely because of the presence of guarantees, which are option-like (i.e., nonlinear), a cell using values for its inputs created by taking averages will not yield appropriate outputs; rather, the outputs will be systematically biased in one direction, limiting if not eliminating altogether the usefulness of the projection.

To address this problem, Milliman in 2008 introduced another method, known as cluster modeling. Here, model cells are created in a manner designed to approximately reproduce various outputs (in this paper we call these "model fit variables"). The process used to do so uses cluster analysis, hence the name of the method. Each model fit variable is typically a present value of some type (or types) of cash flows under a particular "calibration scenario."[1] The selection of the set of scenarios and variables is done in such a way that the user expects that matching those variables is likely to ensure matching the variables that are important for the task (e.g., a 95 conditional tail expectation [CTE] over a large set of scenarios). Once the user is persuaded that the process is working as designed, it is not necessary, in future periods, to run all of the scenarios on the seriatim model; it is necessary only to run the calibration scenarios, generate the clustered model, and then run the smaller (clustered) model through all scenarios.

1. Rather than a projected cash flow, a model fit variable might also represent, for example, some item known as of the projection date without doing any projection, or a projected balance sheet amount. It is not necessary for a calibration scenario to be fully realistic; it is only necessary that it enables you to determine what policies will perform similarly in important realistic scenarios. In particular, you may want to turn off asset-liability interactions in your calibration scenarios, especially if there are a large number of policies in the block.

In this paper, we describe cluster modeling and also introduce a new way to use it, one that allows the algorithm to target specified tolerances that can be defined for each model fit variable. We also present a novel second method, "replicating policies," which uses an optimization routine to construct a small portfolio of insurance policies that closely matches key characteristics of the full portfolio within specified tolerances.

Our analyses and experience demonstrate that both methods perform well for the intended tasks. Both techniques can be used in various ways within the reporting pipelines of insurance companies; additionally, cluster modeling is currently available within Milliman's software solution Integrate.

Key use cases for compression methods include reducing runtime of stochastic and deterministic valuations or required capital computations for large life insurance portfolios. They can also streamline calculations for sensitivity analyses, what-if scenarios, and calibrations for curve-fitting and replicating portfolios. Projection runtimes and computational cost tend to scale with the compression factor, although the time to prepare and run the compression must be considered as well.

Although this paper focuses on large portfolios of insurance contracts, these compression techniques can be applied to other large sets of observations; examples include asset portfolios and scenario sets, where runtimes are also a constraint. While we do not explore these additional applications in more detail here, the methods discussed here are broadly applicable.

After describing the two compression methods, we apply both to two case studies to illustrate the steps involved in using the methods in practice and give an idea of the results. Although clustering may be applied to blocks of business with millions of cells, the blocks here are much smaller. We contrast the approaches and describe lessons learned from years of experience relating to the practical use of cluster modeling and getting senior management and auditors comfortable with its use, most of which also apply to replicating policies. Finally, we present concluding comments.

Cluster modeling

The cluster modeling process was laid out in a 2008 paper, with some additional remarks in a 2009 article.[2] We give a brief description here of the most common way to use the technique.

For each contract being considered, the following information is required:

- Which segment the contract belongs to, i.e., each cluster will contain contracts from only one segment.
- The values of the "location variables" for each contract. Here we assume that these are the same as the model fit variables as described previously.
- The size of the contract, which might be, for example, amount of insurance or account value. Each location variable will be divided by the size; therefore, we try to use a size variable such that, if, for two policies, the ratios of the location variables are all close to N:1, we expect that the same will be true of the size variable.

The algorithm used in cluster modeling since the 2008 paper is described in more detail in Figure 3, which can be found in Appendix I. Like most methods of cluster analysis, it uses a pairwise distance function and aims to put in the same cluster contracts that are relatively "close" to each other, i.e., the distance between them is small. For purposes of setting up this distance function, cluster modeling requires the user to specify a weight for each location variable, where a higher weight means that the user wants to prioritize the variable more in the clustering process.

2. Freedman, A. & Reynolds, C. (August 1, 2008). Cluster analysis: A spatial approach to actuarial modeling. Milliman. Retrieved October 16, 2025, from https://www.milliman.com/en/insight/cluster-analysis-a-spatial-approach-to-actuarial-modeling; Freedman, A. & Reynolds, C. (2009). Cluster Modeling: A New Technique To Improve Model Efficiency. CompAct, 32, 4–8, Retrieved October 16, 2025, from https://www.soa.org/globalassets/assets/library/newsletters/compact/2009/july/com-2009-iss32.pdf.

Once the clusters are determined, a representative contract is selected from each cluster. The contract selected is the one closest to the size-weighted average of all contracts in the cluster.[3] To create the cell model, each representative contract is scaled based on the size variable, i.e., by the ratio of the total size of contracts in the cluster, divided by the size of the representative policy.[4]

The clustering process depends on the selection of weights. A higher weight on a variable can be expected to lead to a smaller error in the variable. However, it is appealing to be able to specify tolerances for each variable (that is, the percentage error in the variable that is considered acceptable) and have an algorithm selecting weights to try to meet those tolerances.

In this paper, we describe a way to directly incorporate tolerances.[5] The clustering is run with an initial set of weights. For each location variable, we calculate the ratio of (a) the absolute value of the percentage error in the variable over (b) the percentage tolerance. If any ratios are above 100%, we take some of those variables, increase the weight of those variables by a specified percentage, and redo the clustering. The iteration continues until all variables are within tolerance, or at some other stopping point.

For the case studies in this paper, we proceed at each iteration based on how many variables have a ratio over 100%. If this number is positive, we increase by x% the weights of half of those variables in the next iteration; the variables with the largest ratios are used for this purpose. Otherwise, the single variable with the largest ratio has its weight increased by y% in the next iteration. The process is cut off after a maximum number of iterations, and the results are reviewed. Information about the fit for all iterations is saved; we do not simply assume that the last iteration will provide the best fit. In practice, it does not always happen that the set of ratios always improves in some sense from one iteration to another. Future research may develop modifications to the method that tend to converge on a better set of weights more quickly.

The iterative-weights algorithm requires multiple clustering runs and, therefore, takes more time than if the weights are simply specified and not changed. However, intuitively, it is reasonable to suppose that, at future valuation dates for the same block, the weights will not need to change much, and fewer iterations will be necessary.

Replicating policies

The replicating polices approach has emerged from the evolution of model compression techniques. At its core, the method uses an optimization routine to select a subset of scaled policies, or cells, from a portfolio of interest. The objective is to closely match key characteristics of the full portfolio using this subset.

The term “replicating policies” is inspired by the concept of replicating portfolios, where a limited set of market instruments is used to approximate the behavior of a large portfolio. A major advantage of the replicating policies technique, as compared to replicating portfolios, is its ability to efficiently produce sensitivities to both market parameters and non-market risk parameters. This capability is particularly valuable for (regulatory) reporting, as well as for risk and capital management.

The specific implementation of this approach varies on the use case. Variations may include the selection of model fit variables, acceptable replication error, choice of optimization function and algorithm, and technical implementation. The required inputs are similar to those used in cluster modeling, although scaling by a size variable is typically not applied. Model fit variables may include aggregate present values and aggregate cash flows, often across multiple sensitivities. Additional targets, such as martingale tests, may also be included in the optimization function.

3. An alternative, not further discussed here, is to use error correction such that the target is offset from the size-weighted average by an amount that would offset deviations from previously selected representative cells.

4. An alternative, not further discussed here, is to adjust the scalars; where there are more clusters than model fit variables, this can be done so as to exactly match the model fit variables, perhaps also meeting some least-squares criterion. However, problems may result if there is a variable that is a linear combination of the others; additionally, if there is a variable that is close to a linear combination of the others, the resulting scalars may not be meaningful. In a sense, this is similar to the replicating- portfolio approach but likely to be less robust.

5. Since the 2008 paper, cluster modeling has been available in MG-ALFA and, more recently, in the Integrate Calculation Engine, which subsumes MG-ALFA. The iterative process described here uses a proof-of-concept version of the calculation engine that is not yet available to Integrate clients. This feature, as well as other enhancements to model compression as a distinct engine within Integrate, is being considered for further development. In the meantime, it may be simulated manually.

A highly effective implementation involves a reasonably straightforward two-step approach:

- Segmentation: The portfolio is divided into blocks of policies that should not be mixed.
- Optimization: For each block, an optimization routine is used to select a minimal subset of scaled policies or cells, such that, in aggregate, the model fit variables of the compressed portfolio match those of the full portfolio within a specified tolerance.

For very large portfolios, an optional preliminary clustering step can be applied using k-means clustering or the method used for cluster modeling or any other clustering method, though it is often not necessary. To illustrate, a portfolio containing 800,000 policies with 1,000 location variables could still feasibly be processed in-memory on a consumer-grade laptop. The final optimization step requires a robust optimization algorithm to ensure that number of selected cells is minimized. In practice, we typically employ an established linear programming algorithm, which is also used in the case studies presented in this paper.

More specifically, the optimization process rephrases the problem as a linear programming task, where location variables for each policy are scaled by a factor. The objective is to minimize the total scaling factors, the number of nonzero factors, and the difference (error) between the total original and the total scaled location variables. Optionally, the error can be weighted, allowing users to adjust its importance during optimization. The approach performs the optimization subject to the constraint that the error must not exceed a predefined tolerance.

The approach can be integrated easily into existing reporting pipelines with various user interface options. Our typical implementation uses Python, though other programming languages may also be employed.

Case studies

Here, we illustrate the compression procedures using two case studies: first, a portfolio of fixed indexed annuities (FIAs), and second, a portfolio comprising endowments and annuities.

CASE STUDY: FIA PORTFOLIO

In the first case study, we focus on a stochastically valued portfolio composed of various insurance policies of a typical U.S. annuity insurer. The portfolio consists of fixed indexed annuities with guaranteed lifetime income. About 30,000 policies were issued over multiple years, and they include different variants (plan codes), distinguished by factors such as underlying assumptions and specific product structures. The portfolio is backed by an asset portfolio of typical bonds and commercial mortgages that are synonymous with those held by typical U.S. insurance companies.

The goal of the compression is to match the present value of cash flows over multiple time buckets with the present value over 30 years to be the most important one. The reason to match cash flows over multiple shorter time buckets is to aid with asset liability management.

The cash flows for the portfolio are calculated and subjected to the more than 1,000 U.S. statutory scenarios plus shocks that are both economic and non-economic in nature. We choose to calibrate the compression across five economic scenarios and test it across 1,400 stochastic scenarios. In other words, the five in-sample scenarios are used to train or calibrate the models, whereas the other scenarios serve as a validation set to evaluate how well the compressed set of model cells generalizes previously unseen conditions.

We define the net present value (NPV) of liability cash flows as a location variable and capture it across various time buckets for each policy for each of the five economic scenarios. In total, we capture five location variables per scenario, yielding 25 location variables. We also defined a compound variable based on the 25 location variables. The goal of this variable is to help ensure a tighter asset liability match. Defining a compound location variable is not typical practice; however, given the research nature of this analysis, we created this location variable.

To compare compression and replication quality in this case study, we use the following metrics:

- Compression ratio: percentage of policies removed from the compressed portfolio
- Goodness-of-fit (GoF) ratio: ratio of compressed to original portfolio[6] NPVs for the full 30 years, shown for the average, and two CTE confidence levels, each generated using 1,400 scenarios
- Coefficient of determination (R^2): calculated for the NPV over 30 years across scenarios

The GoF ratios for CTE showcase the levels of fit not only at the average but also at the extreme tail.

Next, we describe the application and results for both compression methods.

Cluster modeling approach

The size variable was a net of cash flows from one of the calibration scenarios.[7]

Typically, clusters are produced with compression ratios of 90% to 95% for production runs by insurers. On occasion, we have also produced clusters at 99% compression levels if enough homogenous data points within an in-force file exist. Irrespective of the compression level, the reserves or capital calculated via clustering typically falls in a GoF range of 99% to 101% when compared to full seriatim calculations.

Given this was a research project, we chose to test the boundaries of clustering and replicating policies producing compression rates of between 99.8% and 99.9%.

Replicating policies approach

For the replicating policies approach, all five in-sample scenarios were again used for calibration. As is standard practice, all available location variables informed the compressed portfolio. A uniform weight of one was assigned to each variable, and an error tolerance of 10^{-19} was applied.

The resulting compression ratio follows from the optimization approach restricted by the predefined in-sample error of tolerance. Given the relatively small number of variables in this case study, the achieved compression ratio was high, at 99.9%, corresponding to a 30-cell model. Despite the limited number of location variables, the fit to the original portfolio was strong, as demonstrated in Figure 1.

For illustrative purposes, we combined the two compression approaches to show that can also be used to achieve good results.

Results comparison for the first case study

Figure 1 presents the results for the metrics introduced earlier for both compression methods.

FIGURE 1: KEY METRICS FOR CLUSTER MODELING AND REPLICATING POLICIES FOR THE FIRST CASE STUDY

METRIC	CLUSTER	REP. POLICIES	CLUSTER + REP. PORTFOLIO
Compression ratio	99.8%	99.9%	99.8%
GoF ratio | average	100.1%	100.2%	100.1%
GoF ratio | CTE 75	100.3%	100.6%	100.2%
GoF ratio | CTE 98	100.4%	101.3%	100.6%
R^2	99.97%	99.82%	99.92%

6. We measured the model fit against a 90% compressed cluster model as it was too time consuming to run the seriatim in-force file across 1,400 scenarios.

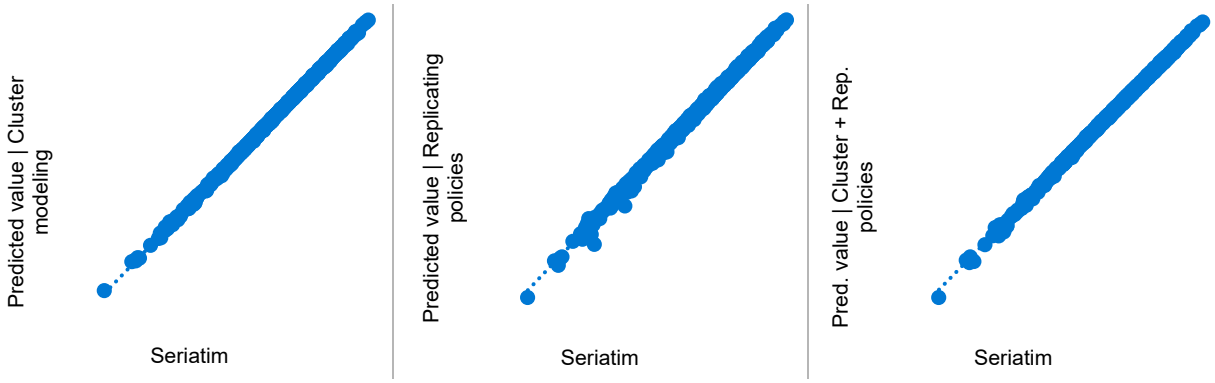
7. Typically, we would use account value as the size variable for FIAs. Given the research nature of this analysis, we experimented with using a variable that was derived from the location variables. For this case study, the outcomes aligned with our expectations.

The three sets of results are based on (1) a 50-cell cluster, (2) a 30-cell replicating policies portfolio, and (3) a 50-cell replicating policies portfolio based on a 250-cell cluster.

Figure 1 shows that all three methods produce very good results even at high compression levels. Combining the cluster and replicating policies methods yields a better fit in some but not all cases. In Figure 1, the tail metric is noisier than only clustering; however, we believe that if we were to try and jointly optimize, we may get a tighter fit.

While these results look good, we analyze results across all scenarios in more detail in a GoF graph and analyze the trendline when reviewing results for the 1,400 stochastic scenarios. The corresponding R^2 numbers are given in Figure 1

FIGURE 2: SERIATIM VS. COMPRESSED PORTFOLIO VALUES FOR THE VARIOUS TESTED COMPRESSION METHODS



While the present value for 30 years looked good, we saw more noise when analyzing results for smaller time tranches. In those tranches, clustering performed better than replicating policies, but given clustering is at a slightly lower compression level, that outcome was within our expectations. In practice, the application of either method could be refined, e.g., through additional calibration scenarios or alternative calibration settings, to achieve the desired GoF.

CASE STUDY: ENDOWMENT AND ANNUITY PORTFOLIO

In the second case study, we shift our focus to the other side of the Atlantic, toward a deterministically valued portfolio composed of various insurance policies of a typical Belgian life insurer. Unlike typical practice, where an initial segmentation step is performed before applying any compression method, we intentionally omit this step to demonstrate that effective replication can be achieved even without prior segmentation.

The portfolio consists of two main product types: endowment and annuity policies. The endowment product represents a significantly larger portion of the portfolio, with 50,000 policies, while the annuity product consists of 5,000 policies. For the annuity product, only a single variant exists. In contrast, the endowment product includes 17 different variants, distinguished by factors such as underlying assumptions and specific product structures. Figure 6 provides background information on these variants, including their respective sizes within the portfolio. As shown in Figure 6, there is considerable variation among the variants, both in terms of the insured sums and the number of policies. The portfolio is a synthetic portfolio; the product characteristics of the portfolio are modeled after those of an average, mid-sized, western European life insurer. A broader set of statistics can be found in Appendix II.

The cash flows for the portfolio are calculated and subjected to the standard formula Solvency II shocks, as well as to exacerbated shocks at 150% of their standard levels. This results in 21 scenarios: one central scenario and 20 shocked scenarios. In addition to the central scenario, four shocks are considered in-sample: catastrophe, mortality, interest down, and expense. The remaining shocks are treated as out-of-sample. In other words, the five in-sample scenarios are used to train or calibrate the models, whereas the other scenarios serve as a validation set to evaluate how well the compressed set of model cells generalizes previously unseen conditions.

The dataset contains NPVs and cash flows for up to 150 projection years, covering a total of 14 different variables. However, most policies mature before reaching the maximum projection horizon. This results in a theoretical total of 2,114 location variables per scenario, a number consistent with implementations observed at insurers.

For analysis, we focus on 14 key variables related to NPVs of different underlying cash flows. Additionally, a single net variable is constructed. We refer to Appendix II for details on these 14 key variables and the construction of the net variable.

To compare compression and replication quality in this case study, we use the following metrics:

- Compression ratio: percentage of policies removed from the compressed portfolio
- Root mean squared error (RMSE): calculated for the single net variable, relative to its base scenario value
- Coefficient of determination (R^2): calculated for the 14 NPV variables across out-of-sample scenarios

Next, we outline the application and results for both compression approaches.

Cluster modeling approach

For the clustering approach, we used the five in-sample scenarios for calibration and generated a 100-cell model corresponding to a 99.8% compression ratio.

While the endowment policies contained a logical size variable given by the sum assured, the annuity contracts did not. Therefore, no size variable was introduced. As mentioned, for a real project we would have treated the two blocks separately, but here we treated them without segmentation.

There are 92 location variables, primarily the net of the NPV and cash flow variables in the base scenario and the net of the NPV variables in the other four calibration scenarios. The key net NPV variables match closely, within or slightly outside of tolerance, for the in-sample scenarios, as these were also given the most weight. Further details and commentary are provided in Appendix II.

Replicating policies approach

For the replicating policies approach, all five in-sample scenarios were again used for calibration. As is standard, all available location variables informed the compressed portfolio. A uniform error tolerance of 0.001% was applied, with default weighting: Each variable received equal weight, except NPVs, which were scaled according to the number of years with nonzero cash flows. The framework also allows for manual adjustment of weights if preferred.

The replication policies approach resulted in a decent compression ratio of 98.5%, which corresponds to 840-cell model. The compressed results fit the original portfolio well, both in- and out-of-sample.

There are also cases when it is relevant to look at individual variables. When doing so, it is interesting to consider the maximum relative error. In this case, the maximum error can be attributed to the NPV of the maturity benefit outgo in the out-of-sample lapse up 150% scenario, where the error was 3.85%.

Selling TBAs provides the lender with price certainty at the time of sale. Lenders can sell forward into TBAs and deliver a pool of mortgages or lenders can trade TBAs (selling at the time of lock and repurchasing the securities prior to delivery) to hedge price risk. Selling forward requires a specified amount of delivery at a specified date and introduces additional risks to lenders. Therefore, most lenders trade TBAs to manage price risk using a duration matching strategy.

Results comparison for second case study

Figure 3 presents the results for the metrics introduced earlier for both compression methods.

FIGURE 3: KEY METRICS FOR CLUSTER MODELING AND REPLICATING POLICIES FOR THE SECOND CASE STUDY

METRIC	CLUSTER MODELING	REPLICATING POLICIES
Compression ratio	99.8%	98.5%
RMSE net variable (in-sample)	0.05302%	0.00078%
RMSE net variable (out-of-sample)	0.36%	0.88%
R^2 of 14 key variables (out-of-sample)	0.9995	0.9999

We have the following observations:

The replicating policies approach achieves a lower compression ratio, which is driven by the higher number of location variables in combination with the chosen tolerance. This is a natural outcome of the difference in the usual treatment of location variables between the approaches. For cluster modeling, the typical approach is to work with a manageable set, although there is no hard limit on the number. For replicating policies, it is typical to use all available variables.

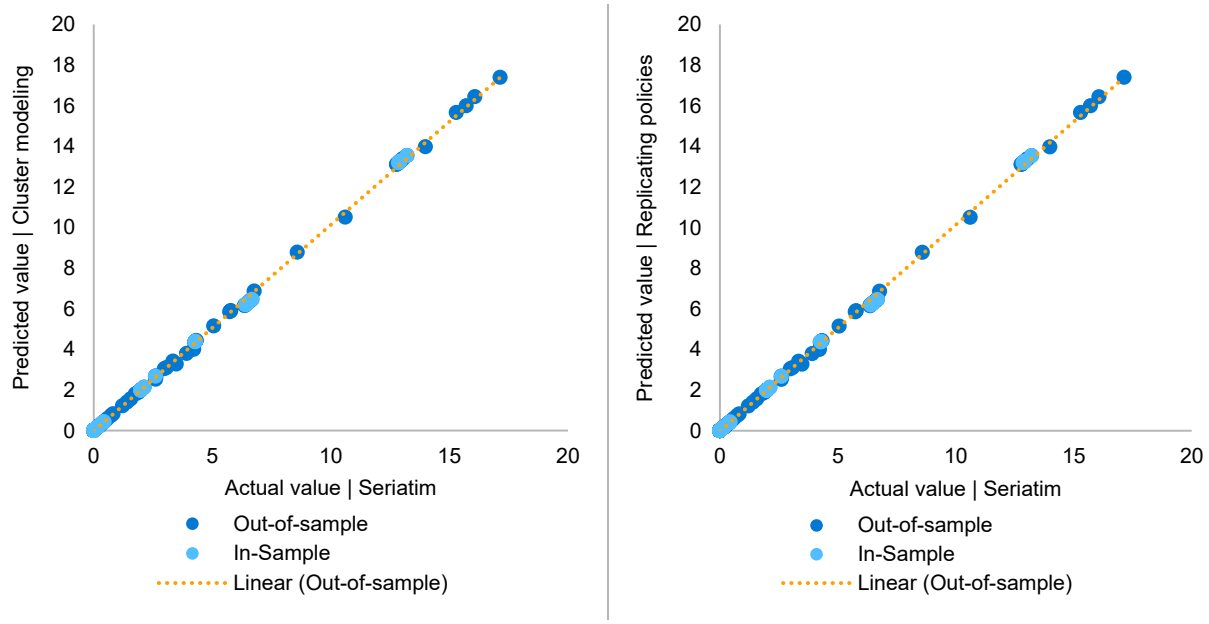
Both techniques achieve near-zero relative in-sample RMSE, consistent with their calibration tolerances.

The relative out-of-sample RMSE is low for both approaches, though somewhat higher for the replicating policies approach.

The out-of-sample R^2 results are high for both methods and align with the RMSE results.

Figure 4 visualizes the GoF graph for both techniques, displaying all NPVs, including a trendline that is fitted to the out-of-sample data points. The accompanying R^2 is given in Figure 3.

FIGURE 4: ACTUAL VS. COMPRESSED PORTFOLIO VALUES FOR CLUSTER MODELING ON THE LEFT AND REPLICATING POLICIES ON THE RIGHT WITH NUMBERS IN EUR BLN



Best practices for practical implementation

The successful application of model compression techniques necessitates meticulous validation, regular calibration, strategic automation, effective communication with stakeholders, and a critical evaluation of its cost-benefit balance. This section delves into these considerations to ensure that compression methodologies deliver optimal value.

MODEL COMPRESSION USE CASES

Grids or cloud computing may make a line-by-line or seriatim approach feasible for some projects, but this is not always the case, especially when either the modeling involves (1) asset-liability interactions, (2) a huge number of stochastic scenarios, or (3) nested stochastic modeling. In these situations, model compression techniques, such as clustering and replicating policies, offer advantages over classical cell models.

Model compression is particularly valuable when a seriatim approach is not feasible. Classical cell compression based on grouping rules may be preferred over more complex methods due to its simplicity. However, such an approach may not be capable of generating a cell model that is both sufficiently small and sufficiently accurate, especially where the important variables are nonlinear, as is the case with products featuring minimum guaranteed benefits. In this case, more complex compression techniques should be exploited.

Compression techniques can also complement or replace other proxy techniques to create computational efficiency. For example, in stochastic economic or required capital calculations, replicating functions or loss functions are often used to make the large number of scenarios, frequently nested, computationally feasible. Compression techniques can streamline these processes, e.g., by efficiently producing calibration data for the proxies or even by directly substituting existing proxy methods.

More generally, the fundamental requirement for applying compression is that the computationally expensive problem involves the sum of a large number of vectors with location variables, in which case compression reduces this to the sum of a much more manageable number of representative scaled vectors.

SELECTING A COMPRESSION TECHNIQUE

Clustering and replicating policies have certain things in common. Most importantly, both are well suited to the task of portfolio compression. They share a focus on model fit variables and their respective tolerances. However, there are also differences, which often come down to user preference and practical considerations.

For example, the replicating policies method is designed to work with many variables, up to many hundreds, and ensure reasonable tolerances for all of them simultaneously. By contrast, although cluster modeling can mechanically handle as many variables as desired, it generally works better if they are pruned to a more manageable set of variables—perhaps by using present values for fewer time periods and combining different types of cash flows. This requires more thinking up front about which variables to use and which to place the highest weight on than is the case with the replicating policies method. In either case, it is important to validate that the process is capable of generating realistic values for the quantities that are important to the user.

In cluster modeling, the number of cells that the clustered cell model is meant to contain is set as a parameter. With replicating policies, the tolerances are set, and the model size is determined by the algorithm. A user might prefer one of these approaches over the other.

A distinction between the two approaches is how their results connect to the underlying policies. In cluster modeling, the link is built into the method: each specific cell directly represents a set of policies that are similar in meaningful ways. In contrast, the replicating policies method produces scalars that allow the replication of aggregate portfolio behavior. These scalars can be linked back to specific policies, but this connection is not intrinsic to the method. Again, a user might prefer one of the methods due to this distinction.

VALIDATING COMPRESSED MODEL RESULTS

Validation is a critical step in the process to ensure that a compressed model is a reasonable representation of the seriatim model, given the purposes of the analysis. Evidence of validation of the compressed model is a key output necessary for stakeholder communication and model governance, particularly when the model is used for financial or regulatory reporting or for strategic applications, such as financial planning and analysis.

Approaches to validation of these models can be classified into two categories: external validation and internal validation. External validation utilizes data other than the information supplied to the algorithm, typically the performance of compressed and uncompressed models under additional scenarios, whereas internal validation leverages only the data available from the inputs or outputs of the algorithm to assess the quality of a model. Although external validation is often the more robust approach, there are benefits to also conducting internal validation to investigate the stability of the results and in cases where it is difficult or expensive to obtain sufficient data for external validation.

External validation

External validation generally involves partitioning the training data to create an out-of-sample dataset; this was done in the case studies by using some scenarios or sensitivities to create the location variables and then analyzing the output from the seriatim and reduced models under other scenarios or sensitivities. If it appears appropriate, post-model adjustments can be applied to correct deviations. An important consideration for the external validation data is that it should cover the types of sensitivities or scenarios under which the model is expected to be used.

Since this process requires potentially expensive runs of the seriatim model, it may be beneficial to try to reduce the number of out-of-sample scenarios by using techniques designed to generate well-spaced scenarios; this may include using the results from the reduced model itself.

Once the external validation scenarios have been determined and results from the seriatim and reduced models obtained, traditional regression metrics such as RMSE can be calculated to assess the fit in aggregate. Additionally, it may be appropriate to specify metrics for tolerances, e.g. “no more than 5% of validation scenarios have an error greater than 2%.” Best practice would be to consider materiality thresholds, as well as the purpose of the analysis, in specifying these metrics and to communicate compressed model accuracy to stakeholders to factor in decision making.

Additionally, the error distribution (i.e., the residuals) should be examined for potential bias or other anomalies that would indicate that it is not simply random noise. Several statistical tests can quantify the degree to which the errors are normally distributed, such as the Kolmogorov-Smirnov or Anderson-Darling tests. If the reduced model appears to contain a bias, a post-model adjustment can be applied; for example, a compressed model that consistently underestimates the result obtained from the seriatim model could benefit from a bias-correction adjustment, such as adding the average error or baseline error to the compressed model result. As a result, metrics that measure absolute fit, such as the RMSE, can improve.

Internal validation

In the case of cluster modeling, internal validation generally involves analysis of the clusters themselves or comparison of the clusters to the original data prior to clustering. Ideally, clusters are both distinct (separate) and compact (cohesive), and various metrics have been developed to score a cluster based on these criteria. For example, the silhouette score measures how different the constituents of a cluster are from their own cluster compared to other clusters. A low silhouette score indicates that clusters may be overlapping and possibly that some points may be better assigned to a different cluster. It may be that more clusters are warranted because diverse points are grouped together or that the number of clusters could be reduced to create more distinction by grouping similar clusters together. A shift in the rate of improvement of the silhouette score versus the number of clusters can indicate an optimal number of clusters, which can be visualized as an “elbow” in the graph of the silhouette score against the number of clusters.

Stability analysis

It may also be of interest to analyze the stability of clusters when the dataset is perturbed. This can be done by adding or removing data from the set, for example, by using bootstrap resampling or by using a different scenario or set of scenario results as the basis for clustering. This perturbation is followed by re-clustering and comparing the new cluster set to the unmodified cluster set to see if similar clusters are formed. For example, when removing a single seriatim point from the dataset, a harmonious result would be that no clusters are modified except for the cluster containing the removed point.

Externally, this can be extended to the period-to-period stability of clusters: In the case of a set of insurance policies, policy terminations or small changes in the market state should not have a significant impact on the cluster assignments. This can be measured by applying the prior period's cluster groupings and assessing model fit via external validation methods or by assessing the number of cluster grouping changes from period to period. However, for products where policyholder behavior or other events can result in significant state changes, e.g., activation of a guaranteed minimum withdrawal benefit, index fund rebalancing, or transition from active to disabled, the expectation of cluster stability may not hold, and policies with observed state changes may need to be excluded from stability analysis.

While this discussion is framed in the cluster modeling context, the same stability analysis principles can also be applied to the replicating policies approach. In that setting, stability checks can help ensure that the reduced model continues to approximate the liability profile consistently over time, even as minor changes occur in the underlying policy set or market environment.

With respect to internal validation and stability analysis, it is worthwhile to remember that unlike in some other applications of model compression, here the purpose is not to generate a similar compressed model per se but rather to approximately replicate the performance of the compressed model under a variety of scenarios. These tools may reduce the occurrence of surprises due to random noise.

Frequency of validation

Regular calibration and validation are essential due to changing market conditions, policyholder behavior, and regulatory environments in order to maintain the relevance and accuracy of financial projections or valuation over time. A schedule should be established (e.g., annually) to validate the process when there is time and grid availability to perform the seriatim runs that may be necessary. This process may conclude that it is necessary to change the weights on the location variables or other parts of the compression approach design if this is not automated as part of the process. It is also possible to partially validate a compressed model more frequently, for example, performing a baseline seriatim valuation for comparison with the baseline compressed model results can indicate whether it is still well aligned with the seriatim model.

For purposes of roll-forward attribution analysis when using cluster modeling, it may be beneficial to retain mappings of representative cells from the beginning of period to the end of period to avoid introducing noise, with reassignments of weights being an explicit step in the attribution.

Significant events, such as mergers, economic shifts, new product launches, or changes in underwriting guidelines, may provide a reason to validate or calibrate off schedule. These changes may justify the selection of new segments or calibration scenarios. However, in the case of the last two causes, the immediate effect may be small since only new business is affected.

AUTOMATION OF COMPRESSION AND VALIDATION

Automation can significantly enhance efficiency, reduce manual errors, and ensure consistency in the compression and validation processes. In particular, the cost-benefit tradeoff between the human expertise required to maintain and understand a compressed model and the computational savings from using it can be tipped in favor of compression if the process is streamlined and efficient.

Aspects that may be automated include:

- Compression modeling pipelines: Raw data may be processed in multiple steps to produce compressed in-force files without manual intervention, such as (in the case of replicating policies) using a preliminary clustering step. This may extend to automated tuning of key model hyperparameters to improve performance, typically using approaches such as iteration or grid search—for example, the iterative weights adjustment described previously (in the case of clustering).
- Validation scripts: Scripts can be written that automatically calculate validation metrics post-compression to expedite the validation process. Alerts can be set up to flag unusual validation results for further investigation.
- Integration with data systems: Linking the automated compression system with existing data warehouses and actuarial models ensures seamless data flow and the potential for near real-time updates if the compressed model is fast enough and model execution and reporting are part of the automation.
- Reporting automation: Generating automated reports and dashboards facilitates quick dissemination of results and validation metrics to stakeholders.

COMMUNICATING WITH AUDITORS AND MANAGEMENT

Effective communication is vital to gain buy-in from management and satisfy audit requirements. Although lower computing cost certainly can resonate with management, and it is often the main selling point for a compressed model, there are other reasons why a compressed model can be useful. Compressed models can cut processing times significantly and thus allow management to perform more frequent analysis. If management is interested in metrics that require a Monte Carlo simulation, then utilization of a compressed model can help overcome uncertainty in the results by most efficiently utilizing the information in the entire system.

As a simple example, suppose a simulation using 1,000 scenarios of a seriatim model can achieve a 95% confidence interval of +/-2% for the fair value of a seriatim in-force. A 99% compressed version of model is found to be very highly correlated with the seriatim model, and its errors in out-of-sample scenarios compared with the seriatim model are unbiased. A simulation of 100,000 scenarios using the compressed model could be expected to reduce the uncertainty in the simulation result by a factor of 10 (assuming errors are normally distributed) for the same computing cost. Thus, in this case, although the compressed model is a less accurate representation of the seriatim model for a single scenario, it ends up providing a more accurate answer due to a reduction in stochastic measurement error.

Stakeholders should be informed of the limitations of a compressed model, such as its degree of accuracy, the scope of validation, and whether it was designed or fit to a particular use case, to ensure that users of the model can assess whether the model is fit-for-purpose.

Comprehensive documentation should be prepared, covering:

- Sources of data
- A description of the algorithm
- Parameter settings
- Validation process (including the types of scenarios or sensitivities used for out-of-sample testing, e.g., insurance risks and economic risks) and results
- Compliance with regulatory standards

When not to employ compression

Compression techniques can add value, but they may not be optimal when maintenance costs outweigh benefits or when more efficient alternatives exist. One must ensure the model meets its intended purpose and that actuaries can explain, evaluate, and monitor model risk. This often requires specialized expertise, either in-house or via third party, and time for ongoing calibration and validation. Besides the direct costs of maintenance, one must also consider opportunity costs, as similar specialist personnel may also be useful in other areas, such as building predictive models for policyholder behavior, anomaly detection, and development of AI capabilities.

The cost-benefit balance can be shifted by either increasing the value of the model or by reducing the maintenance burden. For the former, this could mean avoiding overfitting to a specific use case and aiming to support multiple model metrics. A model that can support fair value, reserving, and capital calculations both instantaneously and on a projected basis is clearly superior to one that is more narrowly focused, but such a model may be difficult to calibrate. Expansion of the usage of a model through more frequent analysis or a greater number of sensitivities or scenarios can also add value, unlocking analysis that was previously too expensive or even completely infeasible. Examples include shifting from monthly to daily balance sheet monitoring or moving from a deterministic valuation to stochastic valuation. For the latter, the previously discussed automation approaches to calibration and validation can streamline the process. Models that require less frequent calibration and validation due to stability of the compressed model will also have lower maintenance costs, and thus stability versus accuracy becomes a potential tradeoff to evaluate.

In some cases, the methods described in this paper may not be the ideal method for reducing computational costs. Other efficient techniques can be worth investigating in lieu of or in conjunction with them. These include hardware and software optimization, scenario-reduction techniques for stochastic models, curve-fitting models, and neural networks.

Next steps

Compression algorithms offer insurers powerful tools for managing large datasets and enhancing the efficiency of financial projections and valuations. To maximize the value of these techniques, their implementation must be accompanied by rigorous validation, ongoing calibration, strategic automation, and open communication with stakeholders. It is also critical to recognize when alternative approaches might be more suitable. By thoughtfully applying compression techniques and carefully evaluating their impact, insurers can optimize their reporting and monitoring processes, delivering more robust, reliable, and actionable insights.

For further inquiries on how compression techniques and other advanced solutions can improve your reporting efficiency, please contact your Milliman consultant. We are dedicated to helping you achieve greater operational excellence and drive meaningful results for your organization.

Appendix I – Mathematical formulation of cluster modeling

Figure 5 complements the earlier explanation of cluster modeling.

FIGURE 5: OVERVIEW OF CLUSTERING ALGORITHM

G_i is the segment containing contract i

S_i is the size of contract i

$L_{i,x}$ is the value of location variable x for contract i

$L'_{i,x} = L_{i,x} / S_i$ is the unitized value of location variable x for contract i

W_x is the weight of location variable x

$A_x = \frac{\sum S_i L'_{i,x}}{\sum S_i}$ is the average value of location variable x

$SD_x = \sqrt{\frac{\sum S_i (L'_{i,x} - A_j)^2}{\sum S_i}}$ is the standard deviation of location variable x

$L''_{i,x} = \frac{L'_{i,x}}{SD_x} * W_x$ is the unitized and standardized value of location value x for contract i

$D_{i,j} = \sqrt{\sum (L''_{i,x} - L''_{j,x})^2}$ is the distance between contract i and contract j

NN_i, the nearest neighbor of i, is the value of j minimizing $D_{i,j}$

over all j with $j \neq i$ and $G_i = G_j$

$I_i = S_i * D_{i,NN_i}$ is the importance of i

The clustering process starts with each contract in its own cluster. At each iteration, the contract i with the smallest importance is identified and "mapped into" its nearest neighbor, i.e., contract i is eliminated, and the size of contract j is updated by adding in the size of former contract i (so that, at any stage, the sum of the sizes of all contracts that have not been eliminated equals the sum of the sizes of the original contracts). The process continues until the desired number of remaining contracts equals the desired number of clusters.

Appendix II – Details on endowment and annuity portfolio case study

The endowment and annuity portfolio case study is modeled after that of an average, mid-sized European life insurer. In Figure 6, some background statistics are given on the size of the portfolio in terms of the number of policies, the average duration, and the average size of the policies.

FIGURE 6: BACKGROUND STATISTICS ON THE ENDOWMENT AND ANNUITY PORTFOLIO

VARIANT	# POLICIES	AVERAGE DURATION (MONTHS)	AVERAGE SUM ASSURED	AVERAGE ANNUITY
1	7	411	€309,074	
2	7018	345	€478,580	
3	3836	342	€357,667	
4	4072	362	€523,409	
5	1050	317	€374,332	
6	4268	287	€465,438	
7	1209	248	€337,998	
8	2398	267	€541,512	
9	1265	323	€330,917	
10	1114	310	€335,014	
11	12442	274	€590,154	
12	2974	194	€672,419	
13	1185	202	€342,759	
15	1365	205	€698,796	
17	5777	173	€687,859	
18	1	146	€38,460	
19	19	272	€6,715	
20	5000	223	€ -	€3,086

This case study analyzes 14 key variables, each available both as annual cash flows and as an NPV at the start of the projection period. In addition to these variables, an aggregate net variable is derived by combining the individual variables. For context, Figure 7 lists the variables and illustrates how they are integrated into the net variable.

FIGURE 7: VARIABLES USED IN THE ENDOWMENT AND ANNUITY PORTFOLIO AND THEIR SIGN IN THE AGGREGATE VARIABLE

VARIABLE	CONTRIBUTION TO NET VARIABLE
Premium Income	+
Taxable Income and Gains	+
Non-Taxable Gains	+
Death Benefits Paid	-
Surrender Payments	-
Partial Surrender Payments	-
Maturity Benefits	-
Annuity Payments	-
Rider Costs	-
Total Commissions	-
Total Expenses	-
Cash Bonuses Paid	-
Taxes Before Bonus	+
Increase in Active Initial Expenses	+

Scenarios

The case study is based on a range of scenarios comprising a base case and a series of standard Solvency II shock scenarios. See Figure 8. To test model robustness, an additional set of out-of-sample scenarios is constructed by scaling each shock by a factor of 150%.

FIGURE 8: AN OVERVIEW OF THE TYPE OF SCENARIOS IN THE EUROPEAN CASE STUDY

SCENARIO NAME	DESCRIPTION
Catastrophe	0.15% one-year increase in mortality rates
Expense	Maintenance expenses +10% and future expense inflation +1%
Interest Down	Yield curve lowered significantly at short terms, smaller drop at long terms
Interest Up	Yield curve raised significantly at short terms, smaller rise at long terms
Mortality	Permanent +15% to mortality rates for death-benefit contracts
Disability	+35% incidence rates, −25% recovery rates for disability/critical illness
Lapse Down	Lapse rates reduced by 50%
Lapse Up	Lapse rates increased by 50%
Lapse Mass	One-off lapse of 40% (retail) / 70% (group)
Longevity	Mortality rates reduced by 20% for annuity-type business

Clustering approach

Next, we provide more detail on the application of the clustering approach, with the calibration based on the five in-sample scenarios as described earlier. All projections were as of year-end 2023. Contracts were grouped purely based on the policy-level cash flows and reserves.

- For the base case: 32 variables consisting of (1) reserves in-force, (2–15) 14 NPVs per variable as of 2023, (16) net of the 14 NPV variables, and (17–32) net of the cash flows for each year 2023–2038.
- For each of the other four scenarios: 15 variables consisting of the 14 separate NPV variables and the net NPV variable as in the base case. This is illustrated in more detail in Figure 9.

Illustrative details of the model fit variables for the base case are provided in Figure 9 and for the other four scenarios in Figure 10. They show the initial weight, the desired tolerance, and the iterated weight. The highest importance is placed on the net NPV variables, which are bolded, and these are matched very closely, but the individual components match less closely. In fact, the desired tolerances were not met. It would be important, therefore, to make sure that the net NPV is actually a good representation of what is most important to match and that the calibration scenarios chosen reflect the types of variation that are expected to be important. This is what we assume for the case study.

FIGURE 9: ILLUSTRATIVE DETAILS ON THE IN-SAMPLE BASE SCENARIO WITH NUMBERS IN EUR MLN

YEAR	VARIABLE	INIT. WEIGHT	ITER. WEIGHT	TOLERANCE	SERIATIM	CLUSTER	RATIO
2023	Reserves in-force	1.00	1.00	0.5%	22,452	22,643	100.85%
2023	NPV Premium income	1.00	1.00	2.0%	1,989	1,992	100.12%
2023	NPV Tax. inc. and gains	1.00	1.00	2.0%	4,291	4,384	102.16%
2023	NPV Inc. init. expenses	1.00	5.36	2.0%	7	7	94.97%
2023	**Net NPV**	**10.00**	**40.43**	**0.1%**	**(13,688)**	**(13,693)**	**100.04%**
2023	Net cash flow	1.00	1.00	1.0%	-	-	--
2024	Net cash flow	3.00	12.17	0.3%	(1,283)	(1,290)	100.55%
2038	Net cash flow	3.00	10.58	0.3%	(482)	(495)	102.75%

FIGURE 10: ILLUSTRATIVE DETAILS ON THE OTHER FOUR IN-SAMPLE SCENARIOS WITH NUMBERS IN EUR MLN

SCENARIO	VARIABLE	INIT. WEIGHT	ITER. WEIGHT	TOLERANCE	SERIATIM	CLUSTER	RATIO
Catastrophe	NPV Surrender out	1.00	1.00	1.0%	12,903	13,231	102.54%
Catastrophe	NPV Tax. inc. and gains	1.00	1.00	2.0%	-	-	--
Catastrophe	NPV Inc. init. expenses	1.00	5.36	2.0%	7	7	94.94%
Catastrophe	**Net NPV**	**5.00**	**20.21**	**0.1%**	**(13,680)**	**(13,685)**	**100.03%**
Expense	NPV Surrender out	1.00	1.00	2.0%	1,989	1,992	100.12%
Expense	NPV Tax. inc. and gains	1.00	1.00	2.0%	4,291	4,384	102.16%
Expense	NPV Inc. init. expenses	1.00	5.36	2.0%	7	7	94.97%
Expense	**Net NPV**	**5.00**	**20.21**	**0.1%**	**(13,688)**	**(13,693)**	**100.04%**
Interest Down	NPV Surrender out	1.00	1.00	2.0%	2,126	2,131	100.24%
Interest Down	NPV Tax. inc. and gains	1.00	1.00	2.0%	2,611	2,673	102.37%
Interest Down	NPV Inc. init. expenses	1.00	4.05	2.0%	3	3	99.60%
Interest Down	**Net NPV**	**5.00**	**17.57**	**0.1%**	**(15,778)**	**(15,790)**	**100.08%**
Mortality	NPV Surrender out	1.00	1.00	2.0%	1,981	1,983	100.10%
Mortality	NPV Tax. inc. and gains	1.00	1.00	2.0%	4,272	4,363	102.15%
Mortality	NPV Inc. init. expenses	1.00	8.14	2.0%	7	6	91.73%
Mortality	**Net NPV**	**5.00**	**15.28**	**0.1%**	**(13,628)**	**(13,621)**	**99.95%**

Replicating policies

The application of the replicating policies approach in this case study adopts a starting point consistent with that of the clustering approach, using the same variables from the five in-sample scenarios for calibration. Experience shows that the replicating policies method performs well when all available information is incorporated into the calibration process. Accordingly, the full set of NPVs and annual cash flows for all variables in each in-sample scenario was used, without relying on aggregated net cash flow or net NPV values. This approach eliminates the need for any pre-processing of the raw data. The fit quality of the replicating policies approach is discussed together with the results presented earlier in this document.

Solutions for a world at risk™

Milliman leverages deep expertise, actuarial rigor, and advanced technology to develop solutions for a world at risk. We help clients in the public and private sectors navigate urgent, complex challenges—from extreme weather and market volatility to financial insecurity and rising health costs—so they can meet their business, financial, and social objectives. Our solutions encompass insurance, financial services, healthcare, life sciences, and employee benefits. Founded in 1947, Milliman is an independent firm with offices in major cities around the globe.

milliman.com

CONTACT

Aatman Dattani
aatman.dattani@milliman.com

Avi Freedman
avi.freedman@milliman.com

Corey Grigg
corey.grigg@milliman.com

Jan Thiemen Postema
JanThiemen.Postema@milliman.com

Karol Maciejewski
karol.maciejewski@milliman.com

Sijbo Holtman
Sijbo.Holtman@milliman.com