SOCIETY OF
ACTUARIES

Nested Stochastic Modeling for Insurance Companies

November 2016

Nested Stochastic Modeling for Insurance Companies

SPONSOR

Financial Reporting Section
Modeling Section
Committee on Life Insurance Research

AUTHORS

Runhuan Feng, PhD, FSA, CERA
University of Illinois at Urbana-Champaign

Assisted By
Zhenyu Cui, PhD
Steven Institute of Technology

Peng Li, MS
Central University of Finance and Economics

Caveat and Disclaimer

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries or its members. The Society of Actuaries makes no representation or warranty to the accuracy of the information.

NESTED STOCHASTIC MODELING FOR INSURANCE COMPANIES

Contents

Nested Stochastic Modeling for Insurance Companies

1. Executive Summary

Stochastic modeling is commonly used by financial reporting actuaries whenever financial reporting procedures, such as reserving and capital requirement calculation, are performed under various economic scenarios, which are stochastically determined. *Nested stochastic modeling* is required whenever modeling components under each economic scenario are themselves determined by stochastic scenarios in the future. An example might be the stochastic reserving of equity-linked insurance for which a dynamic hedging strategy is employed and the Greeks are stochastically determined. As the insurance industry is moving toward more detailed and sophisticated financial reporting standards and practices, it is expected that the computational burden and technical difficulty will rise with the increasing use of nested stochastic modeling.

Despite the importance of such a topic, literature on applications of nested stochastic modeling in the context of financial reporting has been relatively scarce. The *purpose of this study* is multifold:

(1) We intend to provide a resource to help financial reporting actuaries better understand a variety of nested stochastic techniques available both in the industry and in the academic literature.

(2) With a wide array of competing techniques, we aim to review and perform a comparative analysis of their accuracy and efficiency. Some of these techniques either have never been introduced to financial reporting applications or have not been tested in a fair comparison with other techniques.

It is our hope that the findings of this study can contribute to the expansion of a toolkit available to financial reporting actuaries in the insurance industry. As the old saying goes, "Rome was not built in a day"; there are many issues and circumstances that this report does not address. Further investigations are needed for practitioners who intend to apply these techniques for their company-specific models.

The testing of various techniques under consideration in this report will be carried out with two examples:

- Case I: *Risk-neutral valuation of guaranteed minimum accumulation benefit*

  We take a minimalist approach for this example to capture only the essential structure of a nested simulation. The example is simple enough so that all closed-form solutions can be obtained and used as benchmarks against which the results from other techniques can be tested. The primary focus of this test case is on the *accuracy and validity* of all techniques under investigation.

- Case II: *AG-43 CTE calculation for guaranteed lifetime withdrawal benefit*

  The second example is intended to resemble an actual financial reporting model. While it is still unrealistic to implement a full-fledged model in a single research project, we do include most necessary elements of a financial reporting model on a single cell. We expect that the experimentation with this case will shed some light on common implementation issues of these techniques in more realistic circumstances. Due to the significant increase of structural complexity from Case I, we expect Case II to provide a more realistic contrast on the modeling *efficiency* of various techniques.

We shall review a variety of existing techniques from academic literature and practitioners' publications and propose a few new techniques, all of which are are suitable for financial reporting applications. Note, however, only techniques that are structurally designed for nested simulations are selected for testing. Other techniques, such as importance sampling, which can be used in any non-nested model, are not included in the study, to avoid any potential mixing effect. Nor did we apply any specialized computational tricks or techniques such as parallel computing, or reply on any fast computing hardware such as GPUs, etc. All time consumptions in this report are estimated from experiments performed on a personal laptop, and the results should be interpreted only on relative, not absolute, terms.

Techniques Tested

Method	Brief description
A. Closed-form solutions	Valuations are largely based on closed-form formulas that produce exact values or approximation
B. Crude Monte Carlo	Straightforward simulations based on product design and projection of cash flows
C. Optimal budget allocation	Static allocation of resources between two levels of simulations according to certain criteria
D. Sequential allocation	Dynamic allocation of resources
E. Preprocessed inner loops	Preprocess inner loop results under representative scenarios and infer results under desired scenarios from those under similar representative scenarios.
F. Least-Squares Monte Carlo (LSMC)	Approximate inner-loop items by curve fitting, typically with polynomial approximations
G. Least-Squares Monte Carlo with basis selection	Replace inner-loop items by exponential sum approximations with automatic bases selection.
H. Numerical partial differential equation (PDE) methods	Replace inner-loop items by employing numerical PDE algorithms

Summary of Observations

As each technique has its own advantages and limitations, we do not attempt to identify one technique as universally superior. Rather we intend to showcase the variety of alternative techniques for different models to encourage readers to find the best techniques for their own unique modeling situations. We highlight a few observations from numerical experiments in this research study in the following table.

Method	Pros	Cons
A. Closed-form solution	Very accurate and efficient; Provides benchmarks approximation against which all other techniques can be tested.	Limited to certain stochastic models; Requires expertise to develop solutions.
B. Crude Monte Carlo	Easy to implement; Requires minimal analysis.	Can be extremely time and resource consuming.
C. Optimal budget allocation	Easy to implement formula-based allocation; No more modeling beyond crude MC.	Existing allocation strategies depend on specific risk measures; Can be difficult to generalize.
D. Sequential allocation	Dynamically allocate budget; Ideal use of resources.	Can be slow due to conditional statements in computational algorithms.
E. Preprocessed inner loops	Easy to understand and implement; Modest accuracy in low dimensions.	Difficult to determine boundary points to cover all points for interpolations; Difficult to select grid points in high dimensions.
F. Least squares Monte Carlo (LSMC)	Modest accuracy with small number of inner-loops; Can be used for extrapolation.	Little guidance on basis functions; Difficult to select cross-terms in high dimensions.
G. LSMC with basis selection	Can be more efficient than F due to automatic selection of basis functions.	More analysis involved; Limited literature on high dimensions.
H. PDE methods	Can be highly accurate and efficient; Possible reduction of dimensions to improve efficiency.	Requires expertise for stochastic analysis; Special algorithms for high-dimension PDEs.

2. Acknowledgements

We thank the Modeling Section and Financial Reporting Section of the Society of Actuaries for their generous sponsorship and the Project Oversight Group for their guidance and suggestions throughout the research project. In particular, Mike Leung, Mark Evans and Bruce Rosner shared their insights in the development of test cases and also provided constructive criticism that improved the presentation of this report. We are also indebted to Ronora Stryker for her work behind the scenes to facilitate communications and to promote the dissemination of research findings.

Members of Project Oversight Group:

- Bill Beatty, FSA, FCIA
- Fontaine Chan, FSA, MAAA
- Frank Clapper, FSA, MAAA
- Mark Evans, FSA, MAAA
- Mike Leung, FSA, MAAA (Chair)
- Bruce Rosner, FSA, MAAA
- Ronora Stryker, ASA, MAAA

We also want to thank Sun Feng, FSA, MAAA; Dr. Yang Ho, FSA, MAAA; Yu Feng, FSA, MAAA; Dr. John Manistre, FSA, MAAA, for their comments and suggestions. This research project also received assistance from a number of students at the University of Illinois, including Qianyu Cheng, Tianyi Xing and Yitong Huang. Special thanks should go to an

anonymous actarial software company that generously donated sample spreadsheets illustrating the AG-43 stochastic reserving method, which formed the basis of the second test case in this research study.

While this report is intended to provide sufficient information for replication and further investigation by practitioners, there may have been details unintentionally left out. Many techniques in this report can be extended to more general cases appearing in practice. Should you have questions regarding the content of this report, please feel free to contact us by email at rfeng@illinois.edu.

3. Introduction

Stochastic modeling is used wherever modeling parameters or assumptions vary randomly from one period to the next. While stochastic modeling may be used on any actuarial assumption, its most common use in life insurance business is for interest-sensitive products where financial results are heavily dependent on the economic scenario. Nested stochastic modeling is theoretically required wherever one stochastically calculated parameter is dependent on the value of another stochastically calculated parameter. An example would be a stochastic calculation of required capital where managing a hedging program relies on computations from another stochastic model. As the industry continues to move towards increasingly complex modeling of stochastic components, the computational burden grows exponentially. Insurance companies are seeking ways to avoid or reduce nested simulations.

However, literature on nested stochastic modeling techniques is scarce. Among the limited number of research studies on nested stochastics, most existing techniques are developed in the context of portfolio risk management in the financial industry. To the best knowledge of these researchers, no technique has been specifically developed to address the unique challenges of the financial reporting area.[1] While consulting firms, software vendors and major insurance companies play leading roles in the industry to adopt and commercialize new techniques, their research findings are proprietary and often are not freely accessible to the general actuarial community. The intent of this research study is to fill this persistent gap between the literature and industry practice by addressing the following two questions:

(1) What methods are currently available for nested stochastic modeling?
(2) What techniques can be used to improve accuracy and accelerate the run time for nested stochastic modeling?

We answer these questions by performing an objective and quantitative assessment of various competing modeling techniques. We also intend to make the report self-contained and provide sufficient technical details so that experiments and conclusions in this study can be replicable, verified and further developed by practicing actuaries.

The following eight techniques are either selected from the literature or suggested by project oversight group members and respondents to the accompanying survey:

A. Analytical solutions
B. Crude Monte Carlo
C. Optimal budget allocation
D. Sequential allocation
E. Preprocessed inner loops

[1] For example, nearly all examples of nested simulations in the literature consider only two periods, one of which involves an outer layer of projections with one time step and an inner layer of projections with one time step. However, in the practice of financial reporting, there are always projections of multiple periods, each of which requires further projections into the future. It should be kept in mind that straightforward and repeated applications of existing two-period techniques can be sufficient for the moment, but this approach does not take any potential advantage of a multiple period structure. We intend to address this issue in further research.

F. Least-Squares Monte Carlo (LSMC)

G. LSMC with basis selection

H. Partial differential equation (PDE) method

The inclusion of these techniques is largely based on three criteria:

(1) The technique should address the structure of nested stochastic modeling;

(2) There should be supporting statistical analysis regarding the analytical properties of the statistics used, such as consistency and convergence;

(3) The methodology should have the potential of general applicability to the more complex modeling that an actual financial reporting system requires.

There are, however, a few exceptions. The method of analytical solutions may have limited applicability for criterion (3). However, as we shall demonstrate in the more realistic case, it is sometimes possible to develop analytical approximations that can significantly improve modeling efficiency. The method of preprocessed inner loops may not meet criterion (2), as it does not appear to have been formalized in the statistics literature. However, we include it for comparison due to its known applications in the insurance industry.

The two test cases under consideration are given as follows:

(1) Test case I: We model the dynamics of variable annuity separate accounts by a geometric Brownian motion (independent lognormal model). Consider the insurer's liability from a guaranteed minimum accumulation benefit (GMAB) in five years, which is in essence an European put option on its separate accounts. We are interested in calculating a risk capital based on the present value of one-year Value-at-Risk (VaR) of the GMAB liability as well as the probability function of the GMAB liability.

(2) Test case II: We perform an AG-43 stochastic scenario amount calculation for a single cell of guaranteed lifetime withdrawal benefit (GLWB). To fully employ a nested stochastics structure, we consider the AG-43 calculation for a product line with a delta-hedging strategy. The calculation of the stochastic scenario amount represents outer loops, whereas the delta calculations are carried out from inner loops. To demonstrate the flexibility of testing methods in this report, we include fairly complex product designs such as combinations of roll-up and ratchet options, and we also consider the impact of dynamic policyholder behaviors.

In the remainder of this report, we shall provide a brief description for each technique, including background information and its comparative advantages and disadvantages. However, we try to avoid getting into technical details that may be distracting to readers. Further details can be found in the Appendices and the references at the end of this report.

4. Terminology and Categories

Several sets of terminology appear in the current literature regarding nested stochastic modeling. To avoid confusion, we summarize common terms and define their meanings in this report.

When the simulation is nested, there are typically two levels of sampling procedures, as shown in Figure 1.

- Outer loop/step/stage: The simulation in the first stage of projection. For example, in test case I, an outer loop refers to the sampling of separate account values in one year. We shall call sample paths of equity returns in the outer loops **scenarios**, as outer loops typically represent different economic conditions and scenarios of risk factors under a real-world measure. We often denote the set of n scenarios by $\omega_1, \cdots, \omega_n$.

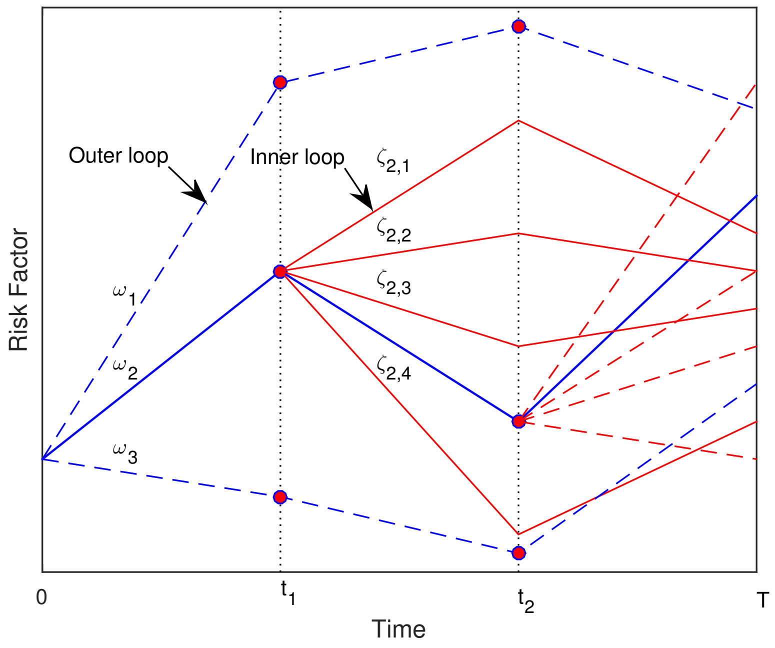

FIGURE 1. An illustration of nested simulations

- Inner loop/step/stage: The simulation in the second stage of projection, for each scenario in the outer loop. For example, in test case I, the inner loop refers to the sampling of GMAB payments in five years. We shall call sample paths of cash flows in the inner loops **paths**, as inner loops typically represent paths of pricing or valuation under a risk-neutral measure conditional on the drawn risk factors from the outer loops. For each scenario in the outer loop, say, ω_k, we denote the subsequent m inner loop paths by $\zeta_{k1}, \cdots, \zeta_{km}$, respectively.

- Time step: Most financial reporting exercises involve recursive evaluation of surplus/earnings over accounting periods. The time step refers to the length of each period for recursive calculation. In test case I, the separate account is modeled by a geometric Brownian motion, which is simulated by repeated multiplications of independent lognormal factors. In this case, the time step is the period for which each lognormal factor is generated. Note that one may use different time steps for outer loops and inner loops.

For consistency, we shall denote the number of scenarios by n and the number of paths by m_k for the k-th scenario where $k = 1, 2, \cdots, n$. In the case of a crude Monte Carlo simulation with a uniform sampling scheme, we shall suppress the subscript, i.e., $m_1 = m_2 = \cdots = m_n = m$.

While all methods aim to speed up nested stochastic modeling, we can summarize their distinctive natures in three categories:

(1) Optimal allocation of resources between outer and inner loops.

There is no structural change to the procedure of nested stochastic modeling. Optimal allocations of a fixed computation budget between outer and inner loops are developed to minimize statistical errors of the ultimate estimator of nested simulation. The optimal budget allocation (method C) and sequential allocation (method D) are in this category

(2) Replace inner loops by closed-form or numerical approximations.

Since the inner-loop calculation is conditioned on each outer-loop scenario, computational efforts resulting from inner-loop calculations grow exponentially for an improvement of accuracy on outer-loop calculations. The intent of this approach is to avoid Monte Carlo simulations in the inner loops by closed-form or numerical approximations, which typically reply on analytical properties of the underlying stochastic models. Analytical solutions (method A) can be considered the most ideal case in this category. The numerical PDE approach (method H) is widely applicable in most stochastic models.

(3) Reduce inner loops by curve-fitting techniques.

The purpose of inner-loop calculation is to create a mapping between items calculated by inner loops and those calculated by outer loops. It is often implicitly assumed that the item calculated by inner loops, say, an insurer's liability, is a continuous function of items calculated from outer loops, say, risk factors such as equity returns.

If we reduce the number of inner-loop simulations for efficiency, only a smaller collection of ordered pairs between equity returns and liability values can be obtained. Many curve-fitting techniques are introduced to connect the ordered pairs in order to produce a continuous mapping between arbitrary equity returns and their corresponding liability values. Preprocessed inner-loops (method E) employ multivariate *interpolation* techniques, whereas Least-Squares Monte Carlo (method F) is based on *smoothing techniques* such as polynomial fitting, and its modified version (method G) is based on exponential fitting.

5. Study Findings on Case I

5.1. **Closed-form solution.** Case I is intended for the calculation of an insurer's economic capital using a one-year mark-to-market approach of a five-year guaranteed minimum accumulation benefit (GMAB) written on a separate account. Assume that the evolution of the separate account value is driven by a geometric Brownian motion, $\{F_t, t \geq 0\}$, under the real-world measure P,

$$\mathrm{d}F_t = \mu F_t \mathrm{d}t + \sigma_0 F_t \mathrm{d}B_t, \qquad F_0 > 0,$$

where $\{B_t, t \geq 0\}$ is a standard Brownian motion. Suppose that the risk-free interest rate is r per time unit. In the Black-Scholes model, the separate account value is determined by

$$\mathrm{d}F_t = rF_t \mathrm{d}t + \sigma_1 F_t \mathrm{d}\tilde{B}_t,$$

where $\{\tilde{B}_t, t \geq 0\}$ is also a standard Brownian motion under the risk-neutral measure Q. Here we use different volatility coefficient σ_1 than the original coefficient σ_0, because the risk-neutral valuation and real-world valuation are typically done at different time points in the nested stochastic model, and we intend to allow different assumptions of volatilities at these two time points.

Risk-neutral valuation

The GMAB offers the greater of a minimum maturity benefit, denoted by G, and the then-current separate account value at the maturity $T = 5$. In other words, the cost of such a benefit from the insurer's perspective is given by

$$\text{max}\{G - F_T, 0\}.$$

For notational brevity, we treat the five-year survival probability as 1. Observe that the GMAB is indeed a European put option. Then the insurer's liability t years from now ($t = 1$) is given by risk-neutral pricing of the put option from the Black-Scholes formula

$$L := e^{-r(T-t)} G\Phi(-d_2(F_t)) - F_t\Phi(-d_1(F_t)), \tag{5.1}$$

where Φ is the cumulative distribution function of a standard normal random variable,

$$\begin{aligned} d_1(F) &:= \frac{\ln(F/G) + (r + \sigma_1^2/2)(T-t)}{\sigma_1\sqrt{T-t}}, \\ d_2(F) &:= d_1(F) - \sigma_1\sqrt{T-t}. \end{aligned}$$

Because of the strong Markov property of the underlying process, the one-year projection of the insurer's liability L is essentially represented as a function of the then-current separate account value F_t. It is well known that the delta of a put option is negative. Hence, L is a strictly decreasing function of F_t. In Section 5.8, we shall illustrate the functional relationship between L and F_t in Figure 6.

Real-world valuation:

Now we consider the economic capital for the guaranteed benefit as the $90th$ percentile of the present value of the one-year net liability,

$$E := \text{VaR}_p(e^{-rt}L), \qquad p = 0.9,$$

where VaR is the Value-at-Risk defined by

$$\text{VaR}_p := \text{inf}\{V : \mathbb{P}(L > V) \leq 1 - p\}, \qquad p \in (0, 1).$$

Observe that in this case, S_t is log-normally distributed and L is a strictly decreasing function of F_t. Therefore, we have a closed-form solution to the economic capital

$$E = Ge^{-rT}\Phi(-d_2(f_p)) - f_p e^{-rt}\Phi(-d_1(f_p)), \tag{5.2}$$

where the number f_p is given by

$$f_p := F_0 \,\text{exp}\left\{\left(\mu - \frac{\sigma_0^2}{2}\right)t + \sigma_0\sqrt{t}\Phi^{-1}(1-p)\right\},$$

and Φ^{-1} is the inverse normal distribution function.

Nested stochastics:

In summary, the inner loops are evaluated under the risk-neutral measure as a conditional expectation

$$L = \mathbb{E}^Q\left[e^{-r(T-t)}(G - F_T)_+ \middle| F_t\right],$$

and the outer loops are evaluated under the real-world measure

$$E = e^{-rt} l_p := e^{-rt} \mathsf{VaR}_p(L).$$

Advantages

- This is ultimately the most efficient and the most accurate approach, as no simulation is involved and hence there is statistical sampling error in final results.
- We often rely on convergence test to confirm accuracy for Monte Carlo simulations (increasing the number of sample points to see if there is any pattern of a converging sequence of results). However, this method fails to demonstrate any potential problem of inaccuracy if the underlying statistics are biased. Analytical solutions often provide benchmarks against which other simulations can be tested for accuracy. Analytical solutions do often exist in special cases of a general model.

Disadvantages

- The more realistic and flexible the model, the less likely that there exist analytical formulas for either the outer-loop valuations or the inner-loop valuations.
- It often requires advanced mathematical techniques to develop closed-form solutions, which are usually represented in terms of special functions. Although it is not a disadvantage per se, many practitioners are not comfortable using these unfamiliar techniques.

5.2. **Crude Monte Carlo.** In a crude Monte Carlo simulation, we carry out calculations in two steps. For the outer loops, we project n sample paths of the separate account up to time t under the real-world measure. We shall label the sample paths by $\{F_t(\omega_1), F_t(\omega_2), \cdots, F_t(\omega_n)\}$. Under each real-world scenario, say, the path of $F_t(\omega_k)$, we further project m sample paths of the separate account values up to time T under the risk-neutral measure, denoted by $\{F_T(\zeta_{k1}), F_T(\zeta_{k2}), \cdots, F_T(\zeta_{km})\}$. Under each real-world scenario, say, ω_k, we look for the true value of the insurer's liability $L(\omega_k)$ resulting from the cash flows generated by separate account values $\{F_T(\zeta_{k1}), F_T(\zeta_{k2}), \cdots, F_T(\zeta_{km})\}$. Thus, we estimate the risk-neutral value of the liability by a sample mean

$$\hat{L}_k := \frac{1}{m}\sum_{j=1}^{m} \hat{Z}_{kj}, \qquad Z_{kj} := e^{-r(T-t)} \max\{G - F_T(\zeta_{kj}), 0\}, \quad k = 1, 2, ..., n. \tag{5.3}$$

Estimating the probability distribution:

We want to estimate the probability distribution function $\alpha = \mathbb{P}(L < V)$ via simulation for a given V. Since the inner-loop simulation generates independent and identically distributed (i.i.d) paths, we use the unbiased estimator

$$\hat{\alpha} := \frac{1}{n}\sum_{k=1}^{n} \mathbb{1}(\hat{L}_k < V). \tag{5.4}$$

Estimating the VaR:

To estimate the VaR, we sort the random sample $\{\hat{L}_1, \cdots, \hat{L}_n\}$ from the smallest to the largest, denoted by $\{\hat{L}_{(1)}, \cdots, \hat{L}_{(n)}\}$, and we use its order statistic to estimate the VaR, which is a percentile. Therefore, an estimator is given by

$$\hat{E} := e^{-rt} \hat{L}_{(\lceil np \rceil)}, \tag{5.5}$$

where $\lceil x \rceil$ is the least integer greater than x.

Advantages

- It is easy to implement, which requires only minimal training on stochastic models.

- It is very flexible in modeling and accommodates all product designs in the market.

Disadvantages

- Many practitioners call it brute force Monte Carlo, which colorfully describes the computational burden of this procedure. It is well known that in a non-nested setting the sampling error of Monte Carlo simulation in general decreases at the rate of $1/\sqrt{n}$ with n being the sample size. In other words, the sample size has to increase a hundredfold in order for the estimate to improve one significant digit. Since sampling errors arise from various sources, it may be harder to control for nested simulations. For example, if most of the sampling errors are caused by inner loops, increasing the number of outer-loop scenarios by 100-fold does not necessarily improve the accuracy of the outer-loop statistics to the next significant digit. On the other hand, if most of sampling errors come from outer loops, increasing the sample size of inner loops by 100-fold while holding the number of outer-loop scenarios constant does not improve the accuracy of estimates either.

5.3. **Optimal budget allocation.** As the computation cost often imposes a binding constraint on the size of nested simulation, the work of Gordy and Juneja [9] presents a strategy to allocate a fixed budget between inner loops and outer loops in order to minimize mean-squared errors of crude Monte Carlo estimators.

Let γ_1 be the computation cost of each inner loop, γ_0 be the cost of each outer loop and Γ be the total computation budget. The goal of optimal allocation is to find the optimal m and n such that the mean-squared error $\mathbb{E}[(\hat{L}_{(\lceil Mp \rceil)} - \text{VaR}_p(L))^2]$ is minimized, given the budget constraint

$$n(m\gamma_1 + \gamma_0) = \Gamma. \tag{5.6}$$

Note that in financial reporting applications the cost of each outer loop depends on the size of the time step for the projection of various risk factors, whereas the cost of each inner loop relies on the size of the time step for cash flow projections, typically under risk-neutral measures. For example, if the projection in an inner loop is on a quarterly basis for a total of 10 years with a total of two risk factors, we consider the computation cost of $4 \times 10 \times 2 = 80$. If the projection in an outer loop is on a quarterly basis for one year for a total two risk factors, we consider the computation cost of $4 \times 1 \times 2 = 8$.

Note that for each given scenario ω_k, the sample average $\hat{L}(\omega_k)$ is an estimator of the true value of the liability $L(\omega_k)$. As the sample average is a random variable by itself, the difference $\hat{L}(\omega_k) - L(\omega_k)$ is also a random variable, called a “pricing error,” which is scenario-dependent. An essential element in determining the allocation is the conditional variance of the pricing error, $\Theta(l)$, given the true value $L(\omega_k) = l$.

When γ_0 is relatively small in comparison with γ_1 and Γ is very large, the optimal n^* and m^* can be determined by

$$m^* \approx \left(\frac{2\theta_p^2 \Gamma}{p(1-p)\gamma_1} \right)^{1/3} \tag{5.7}$$

and

$$n^* \approx \left(\frac{p(1-p)}{2\theta_p^2 \gamma_1^2} \right)^{1/3} \Gamma^{2/3}, \tag{5.8}$$

where $\theta_p := -\Theta'(l_p)$.

Let us consider an example. Suppose that the computation cost of each inner loop is $\gamma_0 = 80$, and the computation cost of each outer loop is $\gamma_1 = 8$. We are interested in estimating the 90% VaR of a desired quantity from the nested stochastic model, i.e.,$p = 0.9$. The computing facilities allow us to run a total computation budget of $100,000$. We would first run a small sample of estimates and determine a rough estimate of l_p, which is the quantity we try to estimate, as well as the outer-loop scenario ω_k from which the VaR is obtained. Then we run additional inner-loop calculations under this particular scenario ω_k. The sample variance is used to estimate $\Theta(l_p)$. Again consider a similar scenario ω_{k+1} and find its corresponding estimate of the desired quantity. Then the derivative θ_p is estimated by a difference quotient based on the two scenarios ω_k and ω_{k+1}. Now suppose that some calculations show that $\hat{\theta}_p = 19$. Then according to formulas (5.7) and (5.8), we obtain the approximate optimal number of scenarios $n = 5.796$ and the approximate optimal number of paths $m = 215.643$. A detailed example of how all components are determined can be found in Section 5.8.

One should keep in mind that solutions to optimization problems rely on the analytical properties of risk measures under consideration. Gordy and Juneja (2008) provided optimal allocations of m and n for estimating the probability function, VaR and conditional tail expectation.

Advantages

- Rather than blindly assigning numbers of inner loops and outer loops in crude Monte Carlo, this method provides a strategic allocation of resources.

Disadvantages

- It is generally difficult to obtain the exact distribution of pricing error. It is reasonable to make a normality assumption due to the Central Limit Theorem. However, it might not be easy to obtain the variance of the pricing error.
- The approximate optimal allocation in (5.7) and (5.8) requires the exact value of l_p, which is precisely the unknown quantity to be estimated by nested simulation. A practical solution is to replace the unknowns with estimates as suggested in the example above.

5.4. **Sequential allocation of inner loops.** The aforementioned two methods are both based on a uniform distribution of computation budget for each outer-loop scenario. In other words, each estimator from inner loops employs the same number of random paths. For the purpose of risk management, we are often interested in extreme events where large losses occur. Therefore, it is inner-loop paths generated from adverse scenarios that actually count in the estimation of desired risk measures. It would be computationally more efficient to dedicate more resources to the most severe cases rather than those with little chance of inclusion in risk measure calculations. Broadie, Du and Moallemi [2] proposed a nested simulation scheme to allocate additional computational effort to scenarios with greater marginal changes to risk measures.

Here we provide a brief description of the strategy. Some technical details can be found in Appendix A. Let $L(\omega_1)$ and $L(\omega_k)$ denote an insurer's liabilities under two outer-loop scenarios ω_1 and ω_k, and $\hat{L}_1$ and $\hat{L}_k$ represent their respective estimators. Take the estimation of the probability function in (5.4) as an example. Whether or not a sample $\hat{L}_k$ affects the estimator $\hat{\alpha}$ relies on its relative position to the threshold V. As shown in Figure 2, the estimators $\hat{L}_1$ and $\hat{L}_k$ are random variables centered around the true values $L(\omega_1)$ and $L(\omega_k)$. However, it is much less likely for $\hat{L}_1$ to reach the threshold V than it is for $\hat{L}_k$. Recall that an increased sample size typically reduces the variance of the estimator. Therefore, it makes sense to allocate more computational resources to $\hat{L}_k$ than to $\hat{L}_1$, as a more accurate estimate of $\hat{L}_k$ may affect the estimator $\hat{\alpha}$, and wasting efforts on that of $\hat{L}_1$ may not have any impact on $\hat{\alpha}$.

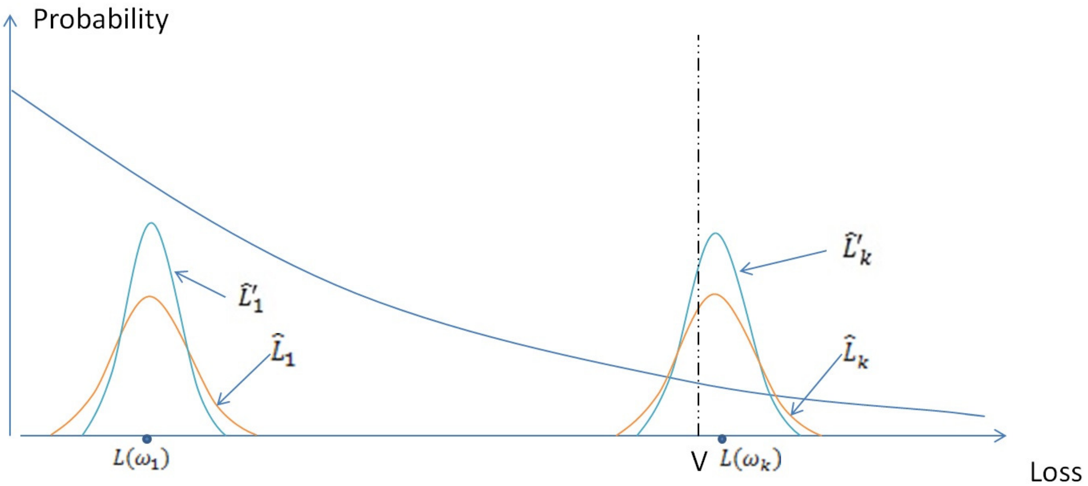

FIGURE 2. Nonuniform sampling of inner loops

The next natural question is how to develop an objective rule of allocating the resources among all scenarios. The sequential algorithm developed by Broadie, Du and Moallemi [2] makes a decision on the allocation of inner paths one at a time. Suppose that the total computation budget is again given by (5.6). In this algorithm, we let $m_k, \hat{L}_k, \sigma_k$ be the current sample size of inner loops, the estimate of liability and the conditional standard deviation of any new sample, for the k-th outer-loop scenario. Here is a summary of their algorithm in three steps:

(1) Allocate m^0 inner loop paths to each outer-loop scenario. (The unused budget $\Gamma - n(m^0\gamma_1 + \gamma_0)$ will be allocated in the remaining steps.)

(2) Search for

$$k^* \in \operatorname*{argmin}_{k=1,2\cdots,n} \left\{ m_k \cdot \frac{|\hat{L}_k - V|}{\sigma_k} \right\}. \tag{5.9}$$

Allocate resources to generate one additional sample to the scenario ω_{k^*}. Update m_{k^*} and $\hat{L}_{k^*}$.

(3) If the total used budget $n\gamma_0 + \sum_{k=1}^n m_k\gamma_1 < \Gamma$, then repeat Step (2).

(4) Compute the risk measures based on $(\hat{L}_1, \cdots, \hat{L}_n)$.

Here is a heuristic argument for choosing k^* according to the criterion (5.9). Recall from (5.3) that

$$\hat{L}_k = \frac{1}{m_k}\sum_{j=1}^{m_k} \hat{Z}_{kj}.$$

Keep in mind that we cannot observe the true values $L(\omega_1), \cdots, L(\omega_k)$ in reality. But we can make our estimator $\hat{L}_1, \cdots, \hat{L}_k$ more accurate by including more inner-loop simulations. The Central Limit Theorem tells us that $\hat{L}_k$ converges to $L(\omega_k)$ as the sample size m_k goes to infinity, i.e., the probability density function of $\hat{L}_k$ will become more and more concentrated around $L(\omega_k)$ as m_k increases. However, we want to conserve resources for scenarios that improve the accuracy in an efficient way. Assume that we have already observed that $\hat{L}_k \geq c$, and if we were to generate an additional inner sample for the outer-loop scenario ω_k, the new estimator is given by

$$\hat{L}'_k = \frac{1}{m_k+1}\sum_{j=1}^{m_k+1} \hat{Z}_{kj}.$$

The key observation is that this additional inner sample to outer loop k will affect only the estimate $\hat{\alpha}$ defined in (5.4) if it is possibly true that $\hat{L}'_k < V$, i.e., $\hat{L}'_k$ has an increased probability to be on the opposite side of $\hat{L}_i$ with respect to V. The increased chance of "sign change" is illustrated in Figure 2. Note that even though $\hat{L}'_k$ has the same mean as $\hat{L}_k$, it has a greater probability mass to the left of V than $\hat{L}_k$. In contrast, the more concentrated density function of $\hat{L}'_1$ makes it less likely that $L(\omega_1) > V$. Therefore, it is not worthwhile to make even more accurate estimate of $L(\omega_1)$. To myopically maximize the impact of the single additional sample, the goal is to choose the scenario ω_i that maximizes the *probability of such a sign change*, i.e.,

$$\mathbb{P}(\hat{L}'_k < V | \hat{L}_k > V).$$

Broadie, Du and Moallemi [2] applied the one-sided Chebyshev inequality to show that the probability is maximized under the scenario k^* identified in (5.9). A simple illustration of a Chebyshev inequality is shown in Appendix A.2. The quantity $m_k|\hat{L}_k - V|/\sigma_k$ itself also provides some insight about the strategy. The minimization procedure for the quantity favors scenarios whose estimators $\hat{L}_k$ are close the threshold V. Among estimators that are all close to the threshold, the minimization procedure favors those with relatively bigger variance σ_k and those with fewer paths m_k.

Observe that the procedure requires knowledge of the conditional standard deviation σ_k for each scenario ω_k. In Case I, we can calculate the exact value of σ_k by $\sqrt{\mathsf{Var}(\hat{Z}_{kj})}$, which is given in Appendix A.1. However, this value may not be known in more complex cases in practice. Nevertheless, σ_k can be estimated from the sample variance of inner-loop paths.

Advantages

- An efficient approach to allocate computational resources to scenarios where the accuracy of inner-loop calculation has the most impact on the overall risk measures.

Disadvantages

- The procedure can be quite time-consuming, as the method requires distributing one inner-loop sample at a time. In comparison with crude Monte Carlo simulation, additional resources are spent on the search algorithm to determine the optimal outer loop to which the next inner loop is to be added. Nonetheless, some remedial measures have been proposed in subsequent publications by Broadie and coauthors. Further research should be pursued if one intends to speed up the procedure.

5.5. **Preprocessed inner loops.** This technique is commonly practiced in the insurance industry. The essence of this method is to preprocess inner-loop calculations (typically risk-neutral valuation of liabilities) with a small set of outer-loop scenarios and then use interpolation to compute other values for desired scenarios outside the preprocessed set. This approach is also introduced and referred to as a factor-based approach in Hardy [10, p. 189].

Preprocessed inner loops

For example, suppose there are two quantifiable risk factors $X^{(1)}$ and $X^{(2)}$ to be considered in the outer-loop simulation. There are many ways in which we can create a set of partition points $(x^{(1)}_1, x^{(1)}_2, \cdots, x^{(1)}_n)$. For instance, $\{x^{(1)}_k = \mathsf{VaR}_{(k-1)/(n-1)}(X^{(1)}), k = 1, \cdots, n\}$ using percentiles, or an equidistant partition $\{x^{(1)}_k = a + (b-a)(k-1)/(n-1), k = 1, \cdots, n\}$ if $X^{(1)}$ falls roughly in a bounded domain (a, b). Similarly, we can create a set of partition points for $X^{(2)}$. These pairs are then tabulated to form a grid system as shown in Table 1. At each grid point, an inner-loop calculation is carried out to determine the corresponding liability (or other quantities of interest under risk-neutral measure), which is denoted by $\hat{L}_{ij}$ corresponding to the i-th scenario of the risk factor $X^{(1)}$ and j-th scenario of the risk factor $X^{(2)}$.

		equity			
		$x_1^{(2)}$	$x_2^{(2)}$	$\cdots$	$x_n^{(2)}$
interest rate	$x_1^{(1)}$	$\hat{L}_{11}$	$\hat{L}_{12}$	$\cdots$	$\hat{L}_{1n}$
	$x_2^{(1)}$	$\hat{L}_{21}$	$\hat{L}_{22}$	$\cdots$	$\hat{L}_{2n}$
	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
	$x_{n'}^{(1)}$	$\hat{L}_{n1}$	$\hat{L}_{n2}$	$\cdots$	$\hat{L}_{n'n}$

TABLE 1. Preprocessed grid

Note that the sizes of partitions n, n' are determined by an insurer's preference of granularity, which is often a compromise between accuracy and efficiency. As the main purpose of such an exercise is to reduce run time, the size of the grid is not expected to be very large.

Interpolation for outer loops

In the outer-loop simulation, a wide range of outer-loop scenarios are generated to reflect the insurer's anticipation of market conditions. When an outer loop requires the liability evaluated with various levels of risk factors, $(x^{(1)}, x^{(2)})$, which are typically not on the grid $\{(x_i^{(1)}, x_j^{(2)}), i = 1, \cdots, n, j = 1, \cdots, n'\}$, approximations are made by interpolating liability values at neighboring points on the table. While there are many multivariate interpolation methods available, the most common one appears to be the bilinear interpolation. Suppose $x_i^{(1)} < x^{(1)} < x_{i+1}^{(1)}$ and $x_j^{(2)} < x^{(2)} < x_{j+1}^{(2)}$. Then a first linear interpolation is done in one direction:

$$\hat{L}(x^{(1)}, x_j^{(2)}) = \frac{x_2^{(1)} - x^{(1)}}{x_2^{(1)} - x_1^{(1)}} \hat{L}_{ij} + \frac{x^{(1)} - x_1^{(1)}}{x_2^{(1)} - x_1^{(1)}} \hat{L}_{(i+1)j},$$

$$\hat{L}(x^{(1)}, x_{j+1}^{(2)}) = \frac{x_2^{(1)} - x^{(1)}}{x_2^{(1)} - x_1^{(1)}} \hat{L}_{i(j+1)} + \frac{x^{(1)} - x_1^{(1)}}{x_2^{(1)} - x_1^{(1)}} \hat{L}_{(i+1)(j+1)}.$$

Then the desired estimate follows from a second linear interpolation:

$$\hat{L}(x^{(1)}, x^{(2)}) = \frac{x_2^{(2)} - x^{(2)}}{x_2^{(2)} - x_1^{(2)}} \hat{L}(x^{(1)}, x_j^{(2)}) + \frac{x^{(2)} - x_1^{(2)}}{x_2^{(2)} - x_1^{(2)}} \hat{L}(x^{(1)}, x_{j+1}^{(2)}). \tag{5.10}$$

There are many other more sophisticated interpolation techniques, such as stochastic kriging. See Liu and Staum [12] for details.

Advantages

- The inner-loop calculation can be reduced tremendously due to the few numbers of inner-loop calculations. It is very easy to implement.
- Most computing software packages provide built-in interpolation functions. It is relatively easy to implement an interpolation procedure.

Disadvantages

- This approach suffers a common phenomenon called “the curse of dimensionality”. Its accuracy deteriorates as one introduces more risk drivers. We shall demonstrate with Case II in Section 6.5 that it is reasonably accurate when we use only two variables, whereas its results are no longer credible after moving to four variables.
- There has been no literature on convergence properties of preprocessed inner loops by linear interpolation, although there are related studies of interpolation techniques such as stochastic kriging for nested simulation.
- Since items required by outer loops are stochastically determined, one has to prepare a large enough grid so that required items can be interpolated from the table of preprocessed inner loops. When risk drivers reside on a very large domain, choosing appropriate boundaries can be tricky, especially in high dimensions.
- In the insurance industry, it was generally believed that the preprocessed grid does not consider path dependency, given that the outer loops typically consist of non-overlapping scenarios. This can underestimate the delta for a step-up benefit. However, it is in fact possible to add additional dimensions to the preprocessed grid to handle path dependency issues. Case II in this report provides an example of handling path-dependency with a step-up benefit.

5.6. **Least-Squares Monte Carlo (LSMC).** The regression-based nested simulation technique was first proposed in Longstaff and Schwartz [13] for the pricing of American options. The work of Broadie, Du and Moallemi [2] laid out a theoretical analysis of LSMC for applications to risk measures.

The idea of LSMC is to replace the inner-loop calculation by an analytical approximation. Curve-fitting techniques should be considered in this category. Suppose an insurer’s liability L depends on a number of risk factors/drivers $\mathbf{F} = (F_1, F_2, \cdots, F_d)$, such as equity return and interest rates. In other words, there exits an unknown function g such that for any scenario ω:

$$L(\omega) = g(F_1(\omega), F_2(\omega), \cdots, F_d(\omega)).$$

Note that in practice L may be obtained from a discrete-time model under each scenario, and hence the function g is usually not known explicitly. Because the graph of g can be viewed as a curve on the $\mathbb{R}^d$ space, there are many curve fitting techniques that can be used to approximate the unknown g by a mixture of known functions, often called *basis functions*. In the LSMC, we consider a set of real-valued basis functions $\phi_1(\cdot), ..., \phi_s(\cdot)$, which can be written as a row vector

$$\Phi(\cdot) = (\phi_1(\cdot), ..., \phi_s(\cdot)) \in \mathbb{R}^s.$$

Some typical examples of basis functions are polynomials. For examples, in a model with two risk factors, one might use $\phi_1(x_1, x_2) = x_1, \phi_2(x_1, x_2) = x_2, \phi_3(x_1, x_2) = x_1 x_2$. Then the liability function L is to be approximated by a linear combination of these basis functions, for some vector $\boldsymbol{\beta} = (\beta_1, \cdots, \beta_s)^\top$ to be determined,

$$L(\omega) = g(\mathbf{F}(\omega)) \approx \Phi(\mathbf{F}(\omega))\boldsymbol{\beta} = \sum_{l=1}^{s} \beta_l \phi_l(\mathbf{F}(\omega)),$$

where $\top$ denotes the transpose of a vector. Ideally, basis functions should be easy to evaluate and capture main features of the functional relationship g. It is common that practitioners use lower term polynomials when the functional relationship is entirely data driven.

The unknown vector $\boldsymbol{\beta}$ is typically determined by minimizing mean-squared error

$$\boldsymbol{\beta}^* \in \operatorname*{argmin}_{\boldsymbol{\beta} \in \mathbb{R}^s} \mathbb{E}[(L - \Phi(\mathbf{F})\boldsymbol{\beta})^2]. \tag{5.11}$$

As it is not possible to directly compute $\boldsymbol{\beta}$ without the exact distribution of L, which is unknown in the context of LSMC, we often consider the statistical analog of the optimization problem (5.11),

$$\hat{\boldsymbol{\beta}} \in \underset{\beta \in \mathbb{R}^s}{\operatorname{argmin}} \frac{1}{n} \sum_{k=1}^{n} (\hat{L}_k - \Phi(\mathbf{F}(\omega_k))\boldsymbol{\beta})^2, \tag{5.12}$$

which is a standard ordinary least-squares problem. Recall that we write $\hat{L}_k = \hat{L}(\omega_k)$ for short. If we let $\mathbf{Y} = (\hat{L}_1, \cdots, \hat{L}_n)$ and $\mathbf{X} = (\Phi(\mathbf{F}(\omega_1))^\top, \cdots, \Phi(\mathbf{F}(\omega_n))^\top)^\top$, then

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

Once the vector $\hat{\boldsymbol{\beta}}$ is determined, then inner-loop calculations in the nested simulation will be replaced by evaluations of the analytical function $\Phi(\mathbf{F}(\omega))\boldsymbol{\beta}$. For example, in the estimation of the probability function, we may use

$$\hat{\alpha} := \frac{1}{n} \sum_{k=1}^{n} \mathbb{1}(\Phi(\mathbf{F}(\omega_k))\boldsymbol{\beta} < V).$$

Advantages

- The method is known to be efficient to reduce computational time. In the nested setting of a crude Monte Carlo with n outer loops and m inner loops, the total number computation units is $n(\gamma_0 + m\gamma_1)$ where γ_0 is the computation cost of an outer loop and γ_1 is that of an inner loop. LSMC finds an approximation of inner-loop calculation in a procedure separate from outer-loop simulation. If we use m inner loops to find the approximation and n outer loops to generate an empirical distribution of approximated inner-loop values, then the computation cost would be $n\gamma_0 + m\gamma_1$ plus the cost of least-squares estimation, which is typically significantly less than that of crude Monte Carlo.

- The method has a formal mathematical basis of convergence.[2] In the fitting scenario, although the fitting is subject to a degree of sampling error from the randomness of the fitting scenarios, its convergence has been formally proved in Stentoft (2004). As we add more scenarios and basis functions, the estimators converge to the actual functional relationship.

- After inner loops are replaced by polynomial approximations, the computation requirement is often reduced to the extent that it is affordable to use a very large number of outer-loop scenarios, significantly reducing errors from the outer-loop stage.

Disadvantages

- The asymptotic and convergence analysis in the literature has been done only on risk measures such as the probability of large loss or expected excess loss. There has not been rigorous analysis with regard to the convergence of mean-squared error for common risk measures in practice such as VaR and conditional tail expectation.

- Error analysis requires asymptotics, which can be difficult to use.

- The choices of basis functions such as polynomials, particularly those used in practitioners' publications, appear to be arbitrary. Why would the polynomials x, x^2, x^3 be any better than x^3, x^4, x^5 for approximations? Here is an excerpt from the Barrie Hibbert report, Koursaris [11]:

[2] In particular, Theorem 1 in Stentoft [15] provides the mathematical foundation for using the LSMC method to price options involving multiple stochastic factors or with path-dependent payoff functions, such as Asian options.

> *A drawback of the polynomial regression technique is the large number of potential terms in the basis function. For even moderate numbers of risk drivers this can quickly increase into the thousands and may be greater than the number of fitting scenarios. In this case, a method is needed to select a subset of the polynomial terms that describes the liability function well without over fitting to the random liability valuations. The Akaike Information Criterion is a robust statistical test for this use.*

5.7. **LSMC with basis selection.** While there are other ways of selecting basis functions, we propose a new method based on a technique from the applied harmonic analysis literature. The essence of the methodology is to approximate the unknown liability function by a mixture of complex-valued exponential functions. The original idea was developed in Beylkin and Monzon [1] for approximating complex-valued functions and was adapted in Feng and Jing [6] for actuarial applications. The technique has been extended to multivariate cases, but we shall illustrate the procedure in a single-variable case in this report. Technical details on how the algorithm works can be found in the Appendix Section A.3.

Given $2N+1$ values of a function $f(x)$ on a uniform grid in $[0,1]$ and a target level of error $\epsilon > 0$, the goal is to find the minimal number M of complex weights w_m and complex nodes γ_m such that

$$\left| f\left(\frac{k}{2N}\right) - \sum_{m=1}^{M} w_m \gamma_m^k \right| \leq \epsilon, \qquad \text{for all } 0 \leq k \leq 2N. \tag{5.13}$$

Then we use the same set of complex weights and complex nodes to construct a linear combination of exponential functions as a smooth approximation

$$f(x) \approx \sum_{m=1}^{M} w_m e^{t_m x}, \qquad \text{for all } x \in [0,1], \qquad t_m = 2N \ln \gamma_m. \tag{5.14}$$

Inner-loop approximation

In the case of capital calculation, we consider the insurer's liability as an unknown function of a certain risk driver, say, equity values. We shall consider the domain of the risk factor to be finite, say, $[a,b]$ with $-\infty < a < b < \infty$. We re-scale the domain of the liability function to be

$$f(x) = g\left(\frac{x-a}{b-a}\right).$$

Then we create an equidistant partition of the range $[a,b]$:

$$\{x_k = a + (b-a)k/(2N), \qquad k = 0, \cdots, 2N\}.$$

On each of the partition points, we run an inner-loop calculation to determine the corresponding liability values, which are denoted by $(\hat{L}_0, \hat{L}_1, \cdots, \hat{L}_{2N})$. The rest of the calculation is to use the mapping between x_k and $\hat{L}_k$ to find a smooth function relation between the risk driver and the liability (the item calculated by inner loops).

Consider the $(N+1) \times (N+1)$ Hankel matrix **H** defined as follows:

$$\mathbf{H} = \begin{bmatrix} \hat{L}_0 & \hat{L}_1 & \cdots & \hat{L}_{N-1} & \hat{L}_N \\ \hat{L}_1 & \hat{L}_2 & \cdots & \hat{L}_N & \hat{L}_{N+1} \\ \vdots & & & & \vdots \\ \hat{L}_{N-1} & \hat{L}_N & \cdots & \hat{L}_{2N-2} & \hat{L}_{2N-1} \\ \hat{L}_N & \hat{L}_{N+1} & \cdots & \hat{L}_{2N-1} & \hat{L}_{2N} \end{bmatrix}.$$

For the practical purpose of this application, we shall consider only the case where the Hankel matrix is real-valued. Then we can solve for the eigenvalue problem

$$\mathbf{Hu} = \sigma \mathbf{u}, \tag{5.15}$$

where σ is a real eigenvalue and $\mathbf{u} = (u_0, \cdots, u_N)$ is the corresponding eigenvector. To find the approximation, we rank all the eigenvalues of **H** in a decreasing order:

$$\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_N. \tag{5.16}$$

If we choose an eigenvalue σ_M $(0 \leq M \leq N)$ smaller than the level of error tolerance ϵ, then the rest of the algorithm will enable us to find a set of complex nodes $\{\gamma_1, \gamma_2, \cdots, \gamma_M\}$ and a set of complex coefficients $\{w_1, \cdots, w_M\}$ such that the error of approximation is controlled for all $0 \leq k \leq 2N$,

$$\left| \hat{L}_k - \sum_{m=1}^{M} w_m \gamma_m^k \right| \leq \sigma_M.$$

We shall carry out the computation in the following steps:

(1) (Identify eigenvalue and eigenvector) Construct an $(N+1) \times (N+1)$ Hankel matrix **H** with elements $\hat{L}_k$ where $k = 0, 1, \cdots, 2N$. Find all eigenvalues of the eigenvalue problem (5.15), which are ranked from the largest to the smallest as in (5.16). Choose the largest σ_M smaller than the level of error tolerance. Find the corresponding eigenvector $\mathbf{u} = (u_0, u_1, \cdots, u_N)$.

(2) (Determine complex nodes) Construct the eigen-polynomial $P_{\mathbf{u}}(z) = \sum_{k=0}^{N} u_k z^k$ and find all of its roots. Find the M roots with smallest absolute values $\{\gamma_1, \gamma_2, \cdots, \gamma_M\}$.

(3) (Determine complex weights) Use the method of least squares to determine all the unknowns $\{\rho_1, \rho_2, \cdots, \rho_M\}$ in the equation $L_k = \sum_{n=1}^{M} \rho_n \gamma_n^k$, for $0 \leq k \leq 2N$. In other words, if we set $\mathbf{Y} = (\hat{L}_0, \hat{L}_1, \cdots, \hat{L}_{2N})^\top$ and construct the Vandermonde matrix

$$\mathbf{X} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_M \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \gamma_1^{2N} & \gamma_2^{2N} & \gamma_3^{2N} & \dots & \gamma_M^{2N} \end{bmatrix},$$

then the vector $\hat{\boldsymbol{\rho}}$ can be found by

$$\hat{\boldsymbol{\rho}} = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{Y}.$$

Note, however, if the level of error tolerance is set to be very low, then there could be too many terms included in the approximation. If one plots the sizes of σ's ranked from the largest to the smallest connected with line segments, it is typical that the line segments decline sharply for the first few and tend to flatten afterwards. A "rule of thumb" is to include only the first few σ's right before the line segments flatten.

The end product of the above-mentioned algorithm will produce the desired approximation (5.14),

$$g(x) \approx \sum_{m=1}^{M} \hat{\rho}_m e^{t_m x}, \qquad x \in (0, \infty),$$

where $t_m = (2N/b) \ln \gamma_m$. This concludes the step of inner-loop approximation.

Advantages

- Similar to the LSMC method, the inner-loop approximation and the outer-loop simulation are done in a non-nested setting. It is much more efficient than the nested setting with the crude Monte Carlo method.
- The method provides an automated procedure to find an optimal set of basis functions. The analytical approach of determining basis function can be much more efficient than the regression-based variable selection methods, such as the selection based on the AIC mentioned in the Barrie-Hibbert report, as it does not require repeated calculation of coefficients.
- It allows users to approximate the desired liability function up to a pre-described level of error tolerance. According to the theory of Hilbert space, the approximation up to an arbitrary level of error can be achieved by orthogonal polynomials as well. However, in practice, this can be difficult to materialize with only a predetermined set of candidate polynomials. In contrast, the LSMC based on the Hankel matrix approximation provides an objective approach to determine optimal exponential functions. Unlike the LSMC approach, the error analysis does not require asymptotics, which can be difficult to explain. The error bound can be easily obtained and controlled, as it is calculated in the intermediate step. Users can have a rough estimate of the error level without any additional analysis or computation.

Disadvantages

- The method does not appear to work well for unbounded functional relationships between response variables and dependent variables.

To the best of the researchers' knowledge, this is the first time that the Hankel matrix approximation has been used for nested simulations. Further studies are necessary to better understand applications of this approach.

5.8. **Numerical results.** Here we list all aforementioned techniques for nested stochastic modeling to be compared for Case I. Method H is not included here because analytical solutions are already available. If necessary, a numerical PDE method can be easily developed for this case.

A. Analytical solutions
B. Crude Monte Carlo
C. Optimal budget allocation
D. Sequential allocation
E. Preprocessed inner loops
F. Least-Squares Monte Carlo (LSMC)
   (a) inner-loop fitting based on outer scenarios
   (b) inner-loop fitting based on equidistant grid
G. LSMC with basis selection

All techniques are tested under the following valuation assumptions:

- Risk-free interest rate $r = 0.05$ per annum
- Expected rate of return under the real-world measure $\mu = 0.09$ per annum
- Volatility coefficient under the real-world measure in the outer-loop simulation $\sigma_0 = 0.2$ per annum

- Volatility coefficient under the risk-neutral measure in the inner-loop simulation $\sigma_1 = 0.3$ per annum
- Length of projection for risk-neutral valuation (inner-loop simulation) $T = 5$
- Length of projection for capital calculation (outer-loop simulation) $t = 1$
- Initial fund value for separate accounts $F_0 = 100$
- Total investment guarantee $G = 110$.

Since an exact closed-form formula for the Value-at-Risk is available in Section 5.1, its results are used here as benchmarks against which the accuracy and efficiency of all other techniques are tested.

In the following example, estimates of two risk measures are determined simultaneously for all cases.

- Distribution of the one-year liability (loss function)

$$\alpha = \alpha(V) := \mathbb{P}(L > V)$$

,

- Value-at-Risk of the one-year liability

$$\text{VaR}_\alpha := \text{inf}\{V : \mathbb{P}(L > V) \leq 1 - \alpha\}, \qquad \alpha \in (0, 1).$$

When the closed form is available as in the Section 5.1, VaR_α is a right-inverse function of $\alpha(V)$. Although it may appear repetitive to consider both quantities, they are in fact estimated by different statistics in Monte Carlo simulation techniques. For example, the loss function is often estimated by (5.4), and the VaR of the present value of one-year liability is often estimated by (5.5) or some variations. These statistics have different statistical properties when measured by mean-squared error (MSE) to be introduced below. Hence above-mentioned techniques may behave differently for these two risk metrics.

Measurements of accuracy to be used here are (1) mean-squared errors and (2) bias. For example, in the case of $\hat{\alpha}$ defined in (5.4) with a sample size of N, the MSE is given by

$$\text{MSE}(\hat{\alpha}) = \frac{1}{N} \sum_{i=1}^{N} (\hat{\alpha}_i - \alpha)^2.$$

The sample size N for the statistic $\hat{\alpha}$ is the number of times $\hat{\alpha}$ is estimated, not to be confused with n in (5.4), which is the number of outer-loop scenarios to generate each estimate $\hat{\alpha}$. The bias is defined as

$$\text{Bias}(\hat{\alpha}) = \frac{1}{N} \sum_{i=1}^{N} \hat{\alpha}_i - \alpha.$$

Warning:

We have chosen the risk measures carefully in this section so that all aforementioned techniques have been analyzed analytically in the literature, with the exception of preprocessed inner loops. However, one needs to exercise caution when applying these techniques to other risk measures, such as conditional tail expectation, particularly optimal allocation methods. Keep in mind that each optimal allocation formula is based on the minimization of asymptotic mean-squared errors of a specific risk measure. In the case of the probability distribution and the case of Value-at-Risk, they lead to the same optimal numbers of inner loops and outer loops, as shown in Gordy and Juneja [9]. However, the results would not necessarily be the same for other risk measures, such as conditional tail expectation. A blind use of the same allocation would lead to a suboptimal solution or possibly even worse than a crude Monte Carlo simulation.

Method	Outer loop	Inner loop	Mean	Bias	MSE	Time (secs)
A	-	-	0.9500	-	-	0.00000
B	100	100	0.9262	-0.02433	0.00132	0.01014
C	150	67	0.9233	-0.02667	0.00072	0.01565
E	100; 1000	100	0.9438	-0.00627	0.00026	0.01248
F(a)	100; 1000	100	0.9548	0.00482	0.00010	0.01092
F(b)	100; 1000	100	0.9441	-0.00595	0.00013	0.00624
G	101; 1000	100	0.9535	0.00345	8.955×10^{-5}	0.00990

TABLE 2. **(Budget $\approx 10^4$)** Probability of large loss for 20 repetitions.

Method	Outer loop	Inner loop	Mean	Bias	MSE	Time (secs)
A	-	-	25.4792	-	-	0.00000
B	100	100	26.5808	1.10164	3.85057	0.01014
C	150	67	26.6173	1.13813	1.49494	0.01564
E	100; 1000	100	25.8917	0.41259	1.31196	0.01248
F(a)	100; 1000	100	25.1624	-0.31685	0.46592	0.01092
F(b)	100; 1000	100	25.8458	0.36666	0.59492	0.00624
G	101; 1000	100	25.3459	-0.13327	0.25677	0.00993

TABLE 3. **(Budget $\approx 10^4$)** Value-at-Risk (VaR) for 20 repetitions.

Testing accuracy and efficiency

In this numerical example, we use the following benchmarks for testing. The analytical method in Section 5.1 shows that the probability that the present value of the one-year liability is greater than $V = 25.4792$ is given by

$$\alpha(V) = 0.95.$$

Conversely, the 95% Value-at-Risk of the present value of the one-year liability is given by

$$\text{VaR}_{0.95} = 25.4792.$$

The average computation time for each estimate of risk measure is reported in seconds. We run a total of 20 repetitions of statistics from each technique, and then estimate their MSEs and biases from the 20 estimates.

Implementation

In Tables 2 – 5, we compare the accuracy and efficiency of all techniques given different computation budgets. Here are the details regarding the implementation of each technique. We use the budget of $\Gamma = 10^6$ for illustration purposes.

A. Analytical solutions

The computation of the VaR is based on (5.2). There is no simulation required for its evaluation.

Method	Outer Loop	Inner Loop	Mean	Bias	MSE	Time
A	-	-	0.95000	-	-	0.00000
B	1000	1000	0.94665	-0.00335	8.155×10^{-5}	0.12012
C	3224	311	0.94541	-0.00459	2.8866×10^{-5}	0.30428
D	1000	-	0.95105	0.00105	5.225×10^{-5}	31.9342
E	1000; 10000	1000	0.94879	-0.00121	1.2134×10^{-5}	0.16926
F(a)	1000; 10000	1000	0.94989	-0.00011	5.2608×10^{-6}	0.14071
F(b)	1000; 10000	1000	0.94323	-0.00677	5.1034×10^{-5}	0.13182
G	201; 10000	5000	0.94993	-0.00007	2.9645×10^{-6}	0.10975

TABLE 4. **(Budget $\approx 10^6$)** Probability of large loss for 20 repetitions.

Method	Outer Loop	Inner Loop	Mean	Bias	MSE	Time
A	-	-	25.4792	-	-	0.00000
B	1000	1000	25.6343	0.15503	0.38696	0.12012
C	3224	311	25.7571	0.27783	0.09828	0.30428
D	1000	-	25.4532	-0.02626	0.33254	31.9342
E	1000; 10000	1000	25.5582	0.07895	0.05499	0.16926
F(a)	1000; 10000	1000	25.4813	0.00179	0.02439	0.14071
F(b)	1000; 10000	1000	25.9106	0.43137	0.21108	0.13182
G	201; 10000	5000	25.5083	0.02906	0.02567	0.10975

TABLE 5. **(Budget $\approx 10^6$)** Value at Risk (VaR) for 20 repetitions.

B. Crude Monte Carlo

We manually allocate the budget to $n = 1,000$ outer loops and $m = 1,000$ inner loops. The probability and the VaR of the one-year liability are computed with the statistics in (5.4) and (5.5).

C. Optimal budget allocation

The budget is allocated and fixed according to asymptotically optimal numbers in (5.8) and (5.7). The number θ_p is determined in Appendix Section A.1. Although in this case lengths of projection are different for inner and outer loops, calculations involve only terminal values F_t and F_T. Hence we set $\gamma_0 = \gamma_1 = 1$. The estimators of the risk measures are given in (5.4) and (5.5). Keep in mind that the optimal allocation was developed based on the minimization of mean-squared errors. As shown in Table 4 and 5, the bias of the resulting statistics may not necessarily be smaller than crude Monte Carlo.

D. Sequential allocation

The algorithm is given an initial allocation of $n = 1,000$ outer loops, and $m^0 = 800$ inner loops. Then it dynamically updates the number of inner-loop paths for each outer-loop scenario in order to myopically maximize the chance of a sign change of the inner loop estimator. As shown in Figures 3 and 4, the total number of inner loops for each scenario varies drastically, with an average $\bar{m} = 1,000$. In Figure 3, we plot the number

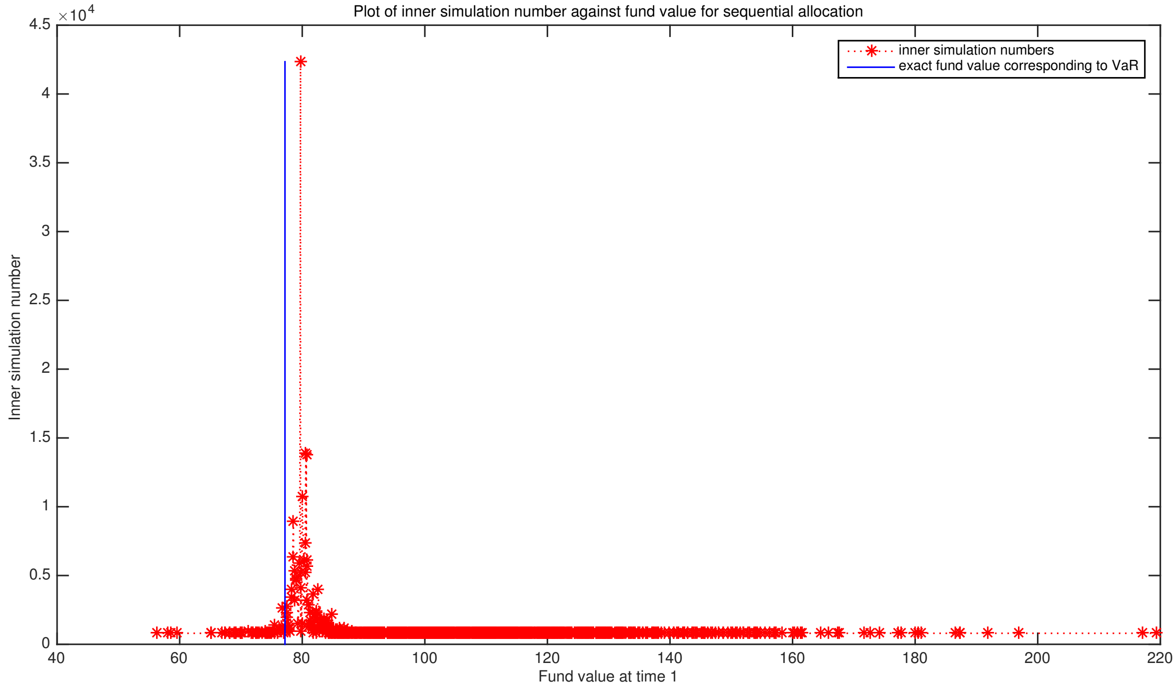

FIGURE 3. Inner sample number versus equity value

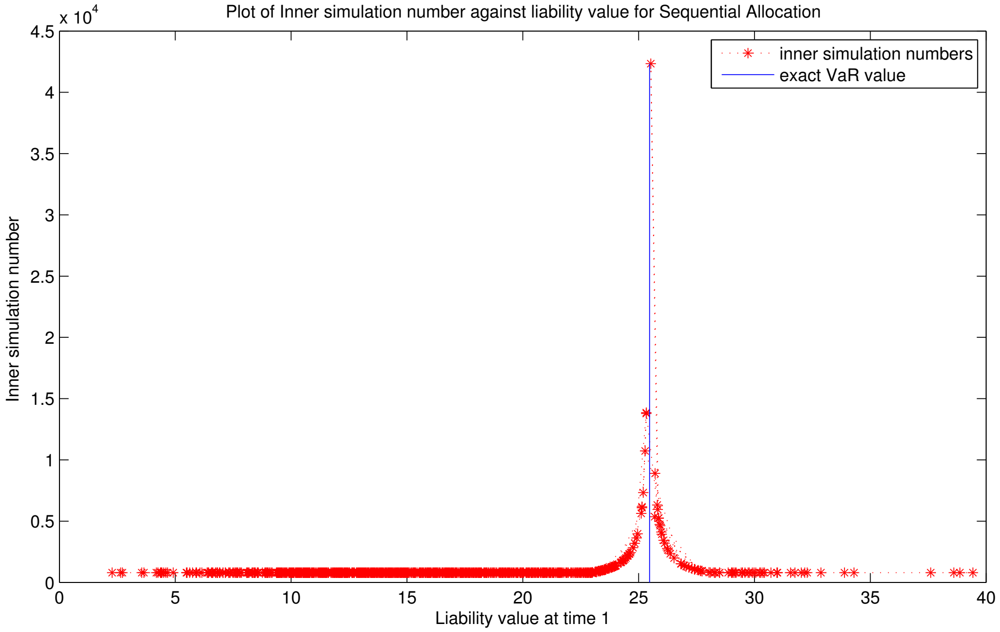

FIGURE 4. Inner sample number versus liability value

of inner-loop paths for each scenario against the corresponding equity value. The distribution of budget for inner-loop calculations is shown against estimated liability values in outer-loop scenarios in Figure 4. It is clear in Figure 3 that there are more inner simulations employed around $F_t = 77.1846$ (represented by the blue vertical line), which corresponds to the 95% VaR of the one-year liability. Represented on a different scale, Figure 4 shows that there are more inner simulations employed around the VaR of the liability distribution given by $\text{VaR}_{0.95} = 25.4792$ (represented by the blue vertical line). This observation is consistent with the principle of the “sequential allocation” method, which is to allocate more inner loops (resources) to the scenarios that may have the most impact on the estimation of risk measures. The observation is in essence a real data realization of the idea shown in Figure 2.

E. Preprocessed inner loops

We use an equidistant grid on $[a = 40, b = 250]$[3] with $n_0 = 200$ nodes. Because a predetermined grid is used, one can think of these equity values as fixed “outer-loop scenarios.” Generate $m = 5,000$ sample points of future cash flows on each node to determine liability values in (5.3). Thus we produce n_0 ordered pairs of equity values and liability values. In the outer-loop simulation stage, another set of $n = 10,000$ scenarios is generated to form an empirical distribution of equity values. In the case of probability α, we map each equity value to a liability value using the linear interpolation outlined in Section 5.5. Note that since there is only a single risk driver, the one-dimensional analogue of (5.10) is used. Once all liability estimates are collected for n scenarios, the probability α is estimated by the statistic (5.4). Similarly, the Value-at-Risk is estimated by applying the statistic (5.5) to interpolated liability values.

F(a). LSMC with fitting based on outer scenarios

For the inner loop approximation, we randomly generate $n_0 = 200$ outer loop scenarios of F_t, denoted by $\hat{F}_1, \cdots, \hat{F}_{n_0}$. Note that these values are not equidistant, in contrast with the method described below. Then we approximate the functional relationship between equity value and liability value by a linear combination of polynomial basis functions $\phi_1(x) = x, \phi_2(x) = x^2, \phi_3(x) = x^3$. In other words, we use a least-squares estimator to determine β such that

$$\hat{L}_k \approx \beta_0 + \beta_1 \hat{F}_k + \beta_2 \hat{F}_k^2 + \beta_3 \hat{F}_k^3, \qquad k = 1, \cdots, n_0. \tag{5.17}$$

In the outer-loop simulation stage, we take a larger sample $n = 10,000$ of time-t-scenarios and approximate the liability under each scenario by the fitted polynomial function. Then apply statistics (5.4) and (5.5) to the approximated liability values.

F(b). LSMC with fitting based on an equidistant grid

In a second trial, we use an equidistant grid on the range of equity values $[a = 40, b = 250]$ with $n_0 = 200$ nodes (time t-scenarios). Generate $m = 5,000$ sample points of future cash flows on each node to determine liability values in (5.3). Thus we produce n_0 ordered pairs of equity values $\{\hat{F}_k = a + k(b-a)/n_0, k = 0, 1, \cdots, n_0\}$ and liability values $\{\hat{L}_k, k = 1, \cdots, n_0\}$. The rest of the calculation is identical to that described in F(a).

[3] We choose this range, because $\mathbb{P}(40 \leq F_1 \leq 250) = 0.999987982608598$.

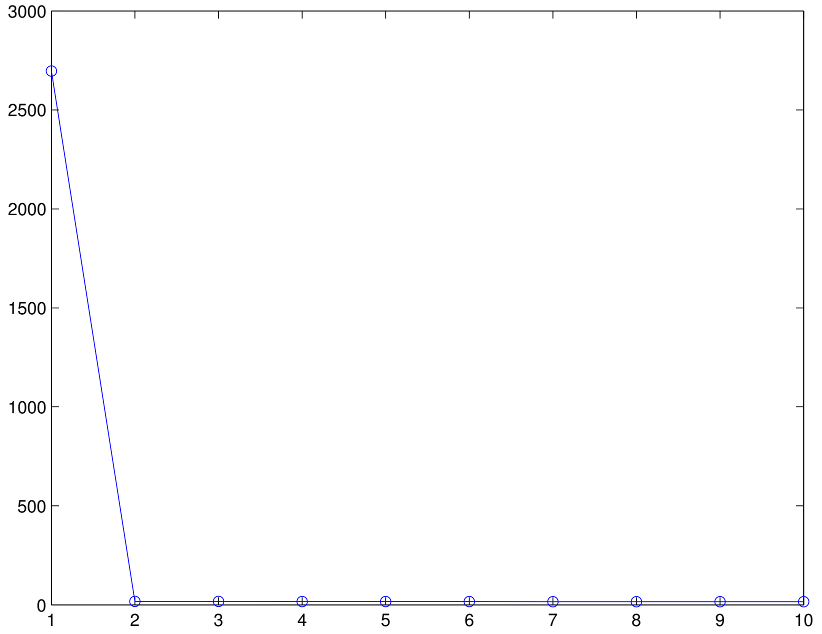

FIGURE 5. Magnitude of 10 largest eigenvalues

G. LSMC with basis selection

We follow the steps outlined in Section 5.7 and use the same grid as in the procedure for F(b). Instead of polynomials, we are effectively using a mixture of exponential functions

$$\hat{L}_k \approx \rho_1 \gamma_1^{2nx_k} + \rho_2 \gamma_2^{2nx_k} + \cdots + \rho_M \gamma_M^{2nx_k}, \qquad k = 1, \cdots, n_0, \tag{5.18}$$

where $\{\gamma_1, \gamma_2, \cdots, \gamma_M\}$ are determined by the eigen-polynomial described in Section 5.7 and for $k = 1, 2, \cdots, 2n$,

$$x_k = \frac{\hat{F}_k - a}{b - a}.$$

However, in this case the number of exponentials, M, is selected so that the maximum error of the approximation is controlled by the eigenvalue σ_M. Figure 5 shows magnitudes of 10 largest eigenvalues obtained from (5.15). It is clear that there is a sharp decline in magnitude from the largest eigenvalue to the second largest eigvenvalue. As pointed out earlier, Figure 5 indicates that there is very little gain from using more than one node in the exponential approximation.

Observations

We compare all techniques in two cases of a budget with various sizes: a small budget of roughly 10^4 and a large budget of roughly 10^6, relatively speaking given the simplicity of the nested model itself.

In the case of a small budget $(\approx 10^4)$ in Tables 2 and 3, it is expected that all other methods outperform the crude Monte Carlo by multiple times in terms of both bias and MSEs. Optimal allocation strategies offer modest improvement on MSEs, although the bias reduction appears to be rather limited. Note that a lower MSE means that one tends to receive a more accurate estimate given a fixed number of simulations. The LSMC methods, including methods F(a), F(b) and G, tend to perform the best among all competing techniques.

In the case of a large budget $(\approx 10^6)$ in Tables 4 and 5, most of the aforementioned techniques perform better than crude Monte Carlo, although the extent of improvement is less than that in the case of a small budget. Note that method G (LSMC method with basis selection) achieves the same level of accuracy as method F (LSMC method with only predetermined polynomials). However, keep in mind that method G produces a rather simple approximate function with only one term of exponential, whereas method F uses a function with three terms of power functions.

5.9. **Error analysis of inner-loop approximation.** As described at the beginning of Section 5, methods E, F and G are in the category of techniques that replace actual inner-loop evaluations by closed-form approximations. Any estimation error of inner loop quantity can carry over and contribute to estimation errors of statistics (5.4) and (5.5). Here we estimate magnitudes of errors introduced by each of these techniques in the earlier numerical examples.

B. Crude Monte Carlo

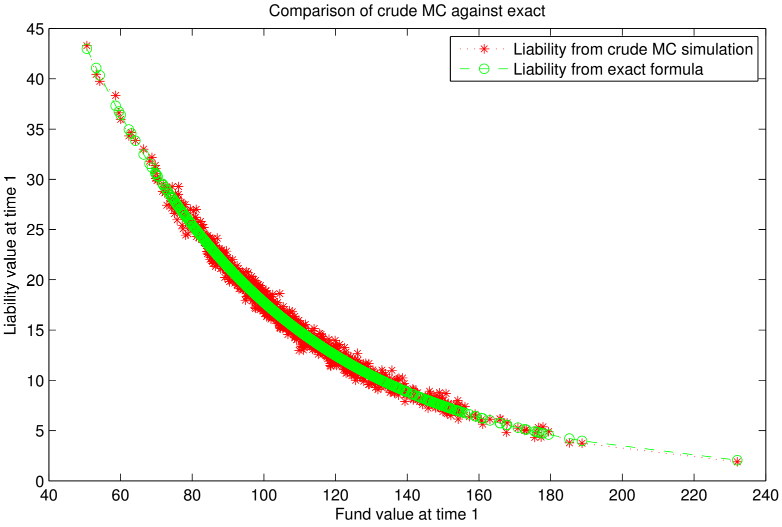

FIGURE 6. Comparison of crude MC versus exact

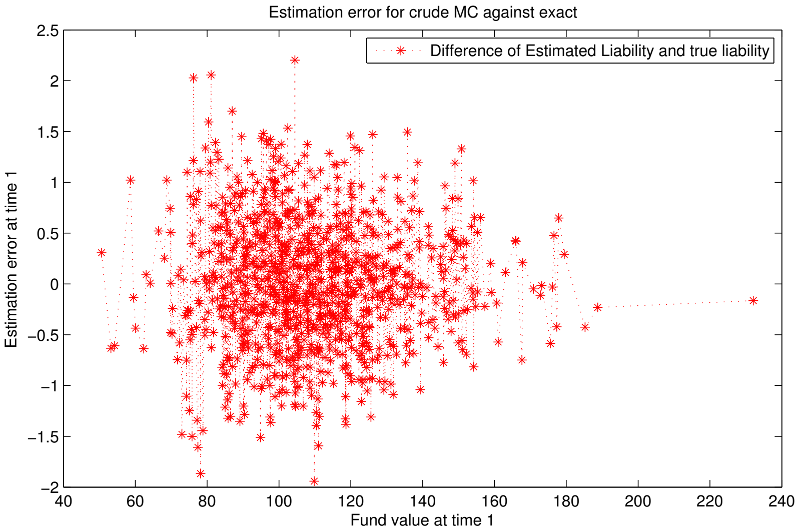

FIGURE 7. Estimation error for crude MC

C. Optimal budget allocation

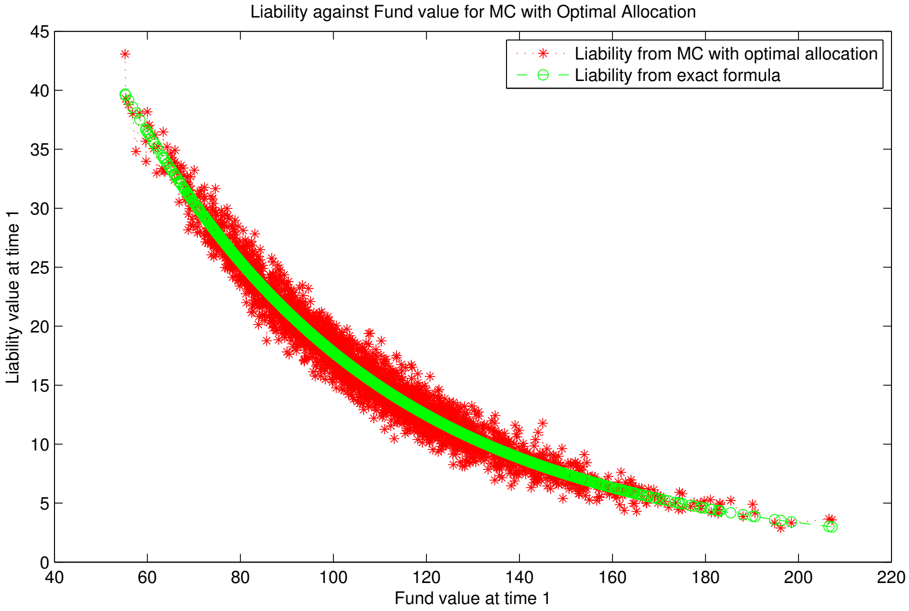

FIGURE 8. Monte Carlo with optimal allocation

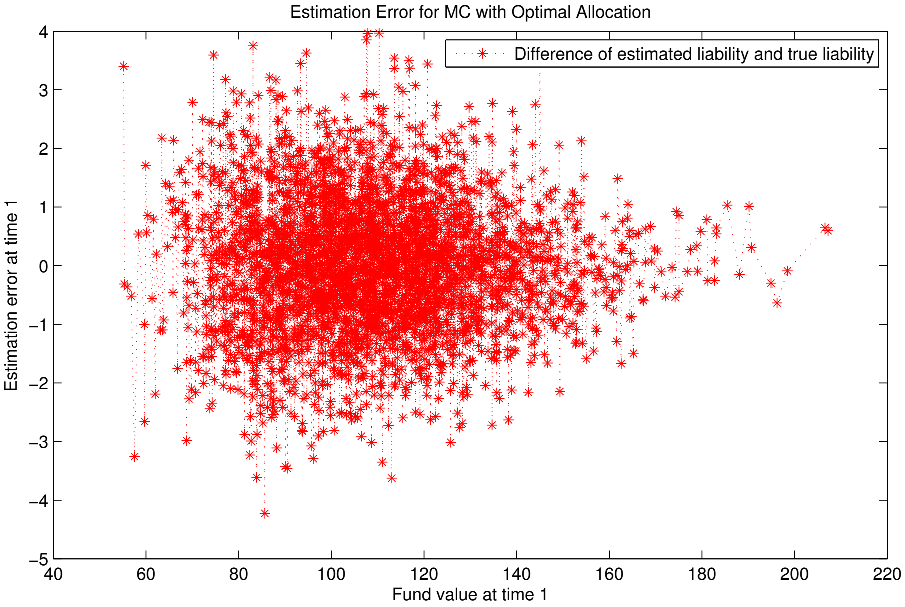

FIGURE 9. Estimation error for optimal allocation

D. Sequential allocation

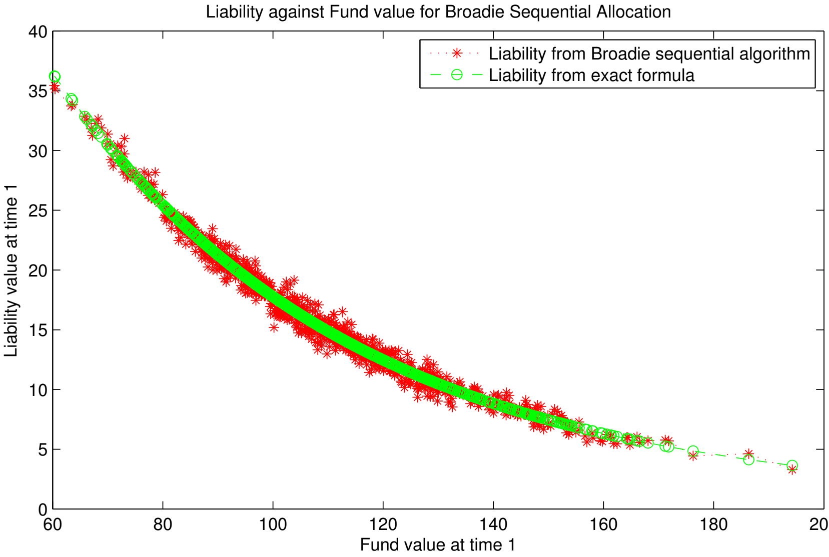

FIGURE 10. Sequential allocation

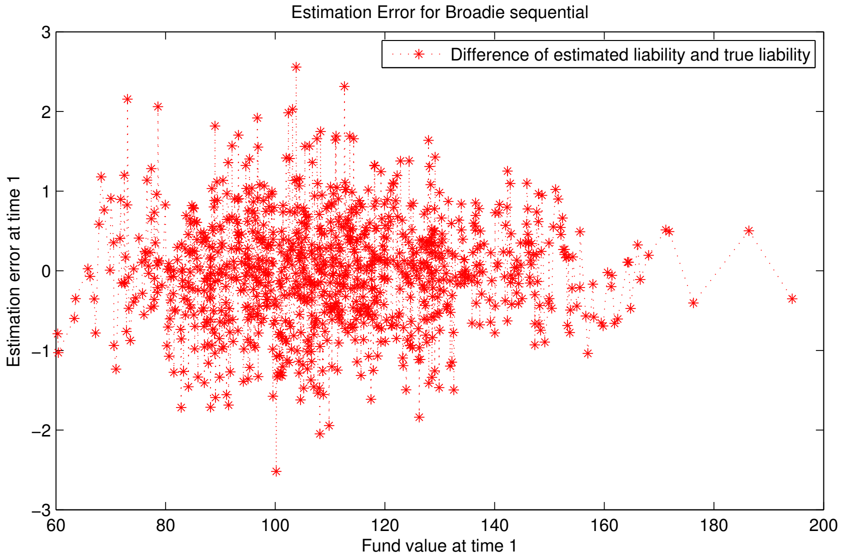

FIGURE 11. Estimation error for sequential allocation

E. Preprocessed inner loops

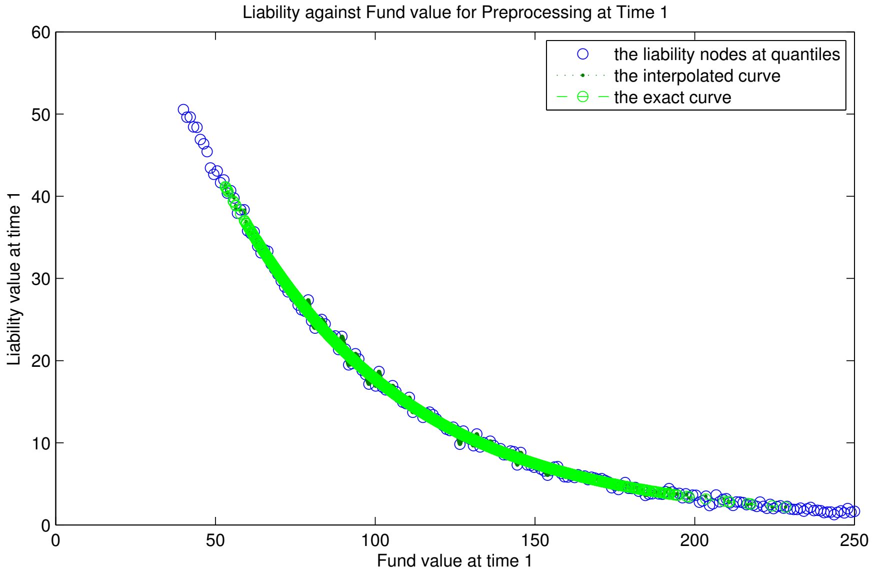

FIGURE 12. Preprocessed inner loops (E)

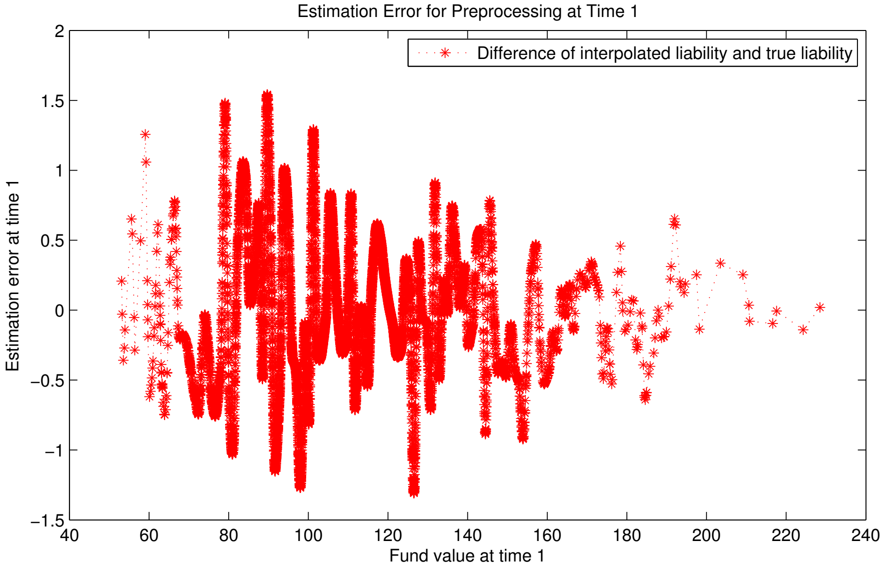

FIGURE 13. Estimation error for preprocessed inner loops (E)

F(a). Least-Squares Monte Carlo with fitting based on outer scenarios

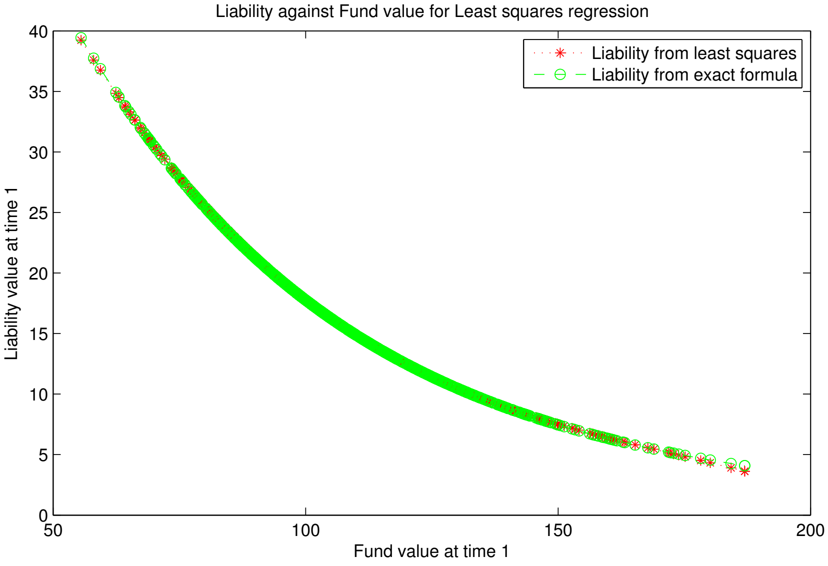

FIGURE 14. LSMC based on outer scenarios (F(a))

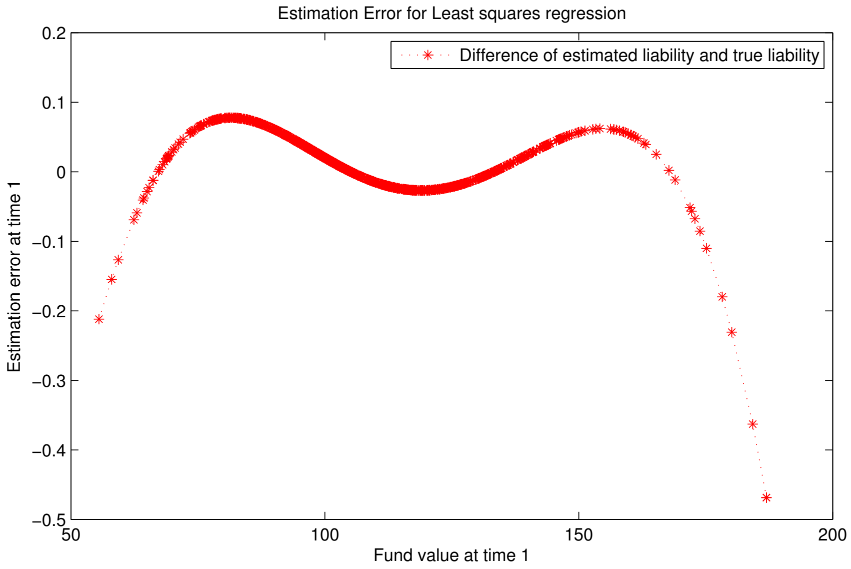

FIGURE 15. Estimation Error for LSMC (F(a))

F(b). Least-Squares Monte Carlo with fitting based on equidistant grid

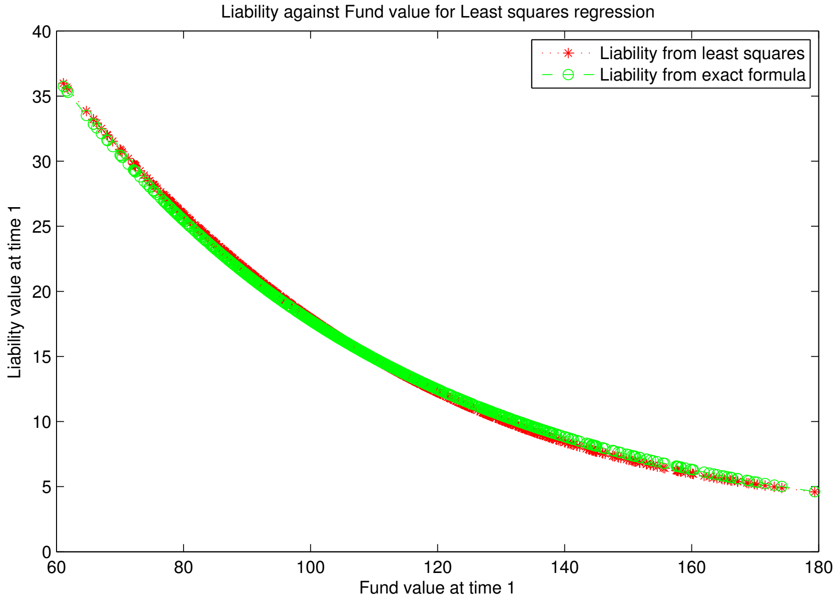

FIGURE 16. LSMC based on equidistant grid (F(b))

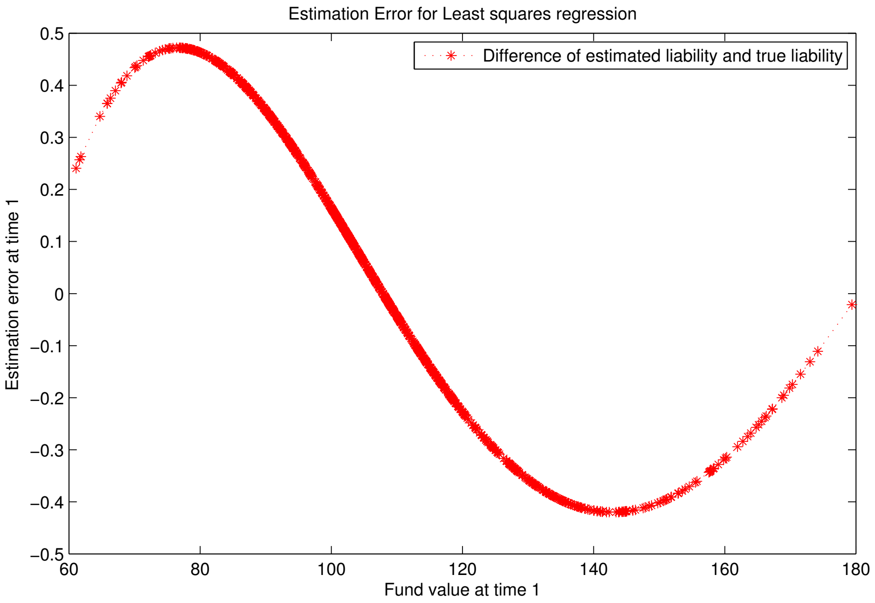

FIGURE 17. Estimation error for LSMC based on equidistant grid (F(b))

G. Least-Squares Monte Carlo with basis selection

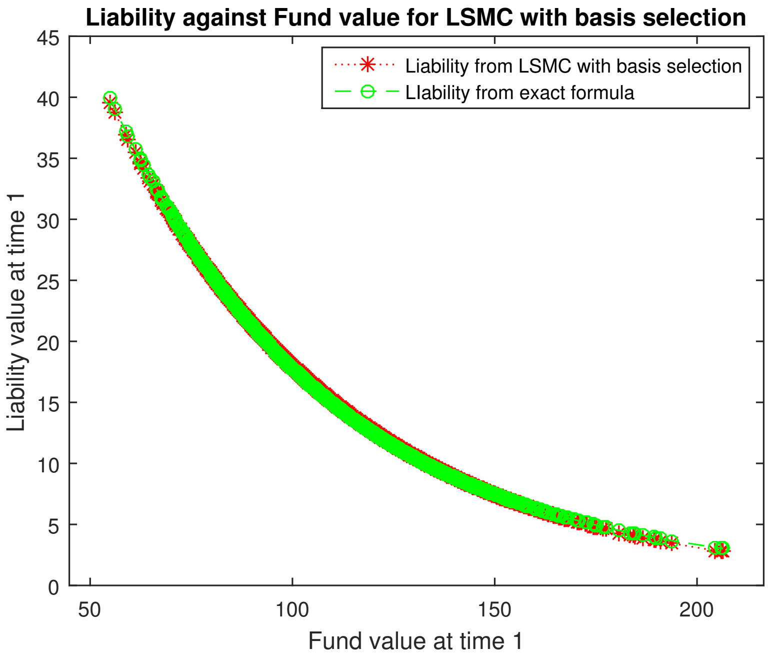

FIGURE 18. LSMC with basis selection (G)

We want to point out that in this example method G can achieve roughly the same level of accuracy as method F(a) with only one exponential term, as shown in Figure 20. The thick green line represents estimation errors resulting from fitting the three term polynomials (x, x^2, x^3) to the data set generated from inner loops. The thin blue line measures

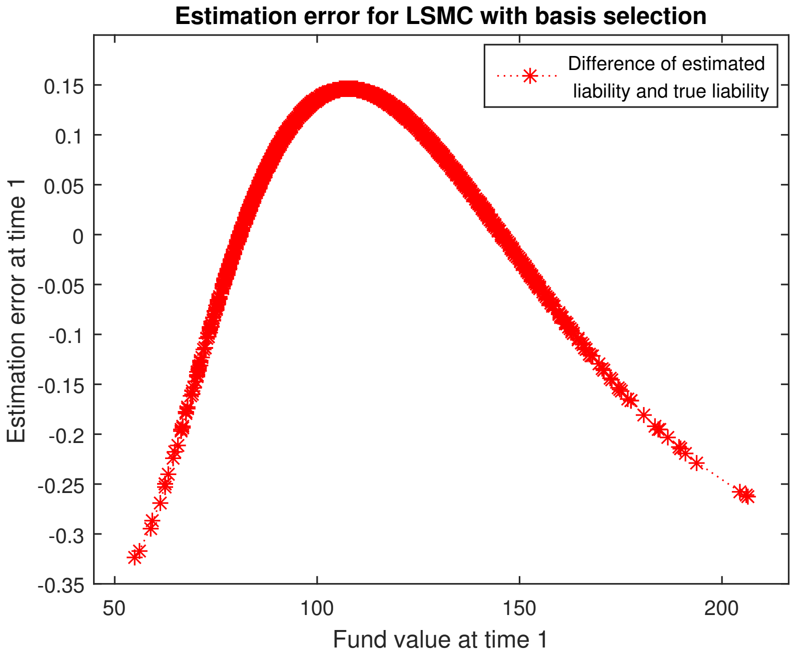

FIGURE 19. LSMC with basis selection (G)

estimation errors resulted from fitting a single exponential function to the same data set, where the exponent and weight are determined by the Hankel matrix approximation. Although the accuracy is not improved, there is a significant savings on the use of basis functions.

From the previous examples, we observe that the Least-Squares Monte Carlo method in general performs well with a small budget and less so with a large budget. If we fix the polynomial function, then estimation errors in the inner-loop evaluation would no longer decline given a sufficient large budget. The Least-Squares Monte Carlo with basis selection can in general avoid this problem, as more and more accurate approximations can be found with an increased budget. In the following, we shall further compare the performance of the LSMC with a fixed polynomial function and the LSMC with exponential basis functions.

In the fitting stage, we choose a fixed equidistant grid $[40, 250]$ with 300 grid points, each of which corresponds to a particular equity value. Then we simulate $m = 1,000,000$ inner loops for each node and use the estimated liability values to fit function forms in (5.17) and (5.18). In test case I, there exists a closed-form formula for the function relation between F_t and L in (5.1). Hence we can actually plot the curve corresponding to this exact function relation. Similarly, we can plot the curves derived from the polynomial function in (5.17) by LSMC and the mixture of exponential functions in (5.18) by LSMC with basis selection.

Figure 21 shows that the estimation errors of LSMC with basis selection (exponential approximation) are on the order of 0.1, whereas those of LSMC (polynomial approximation) are on the order of 1 for equity values.

For Figure 22, we increase the number of inner loops for each scenario by 10-fold, i.e., $m = 10,000,000$, which leads very accurate estimates of liability values. It is clear that the polynomial approximation no longer improves, as it has reached the best possible outcome of the fixed function form. On the other hand, the basis selection method based

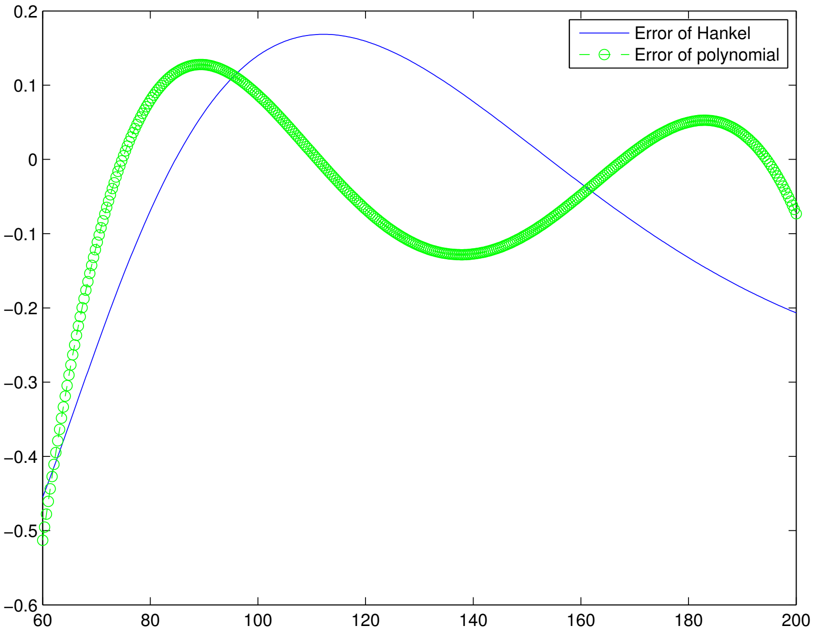

FIGURE 20. Comparison of estimation errors for methods F(a) and G

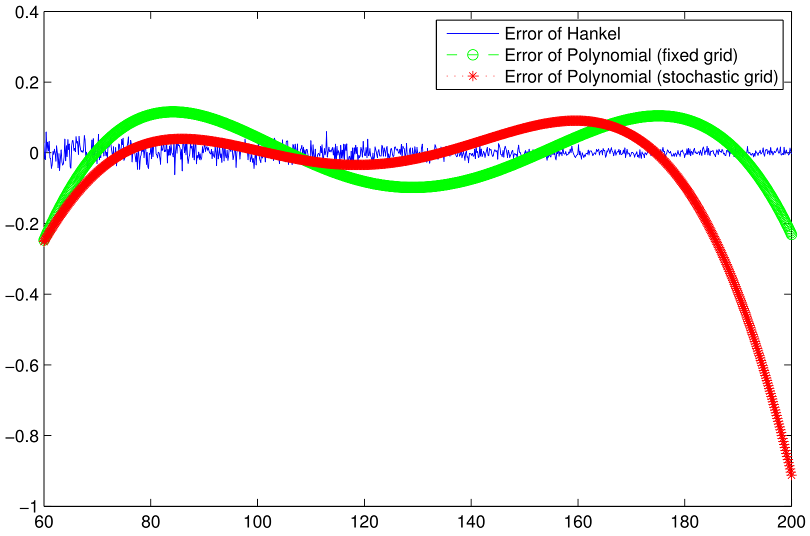

FIGURE 21. Estimation errors for polynomial versus exponential approximations with $m = 1,000,000$

on Hankel matrix approximation continues to improve with the more accurate inner-loop evaluation. The estimation error is on the order of 0.01 for the latter in Figure 22, while that remains the same for the polynomial approximation.

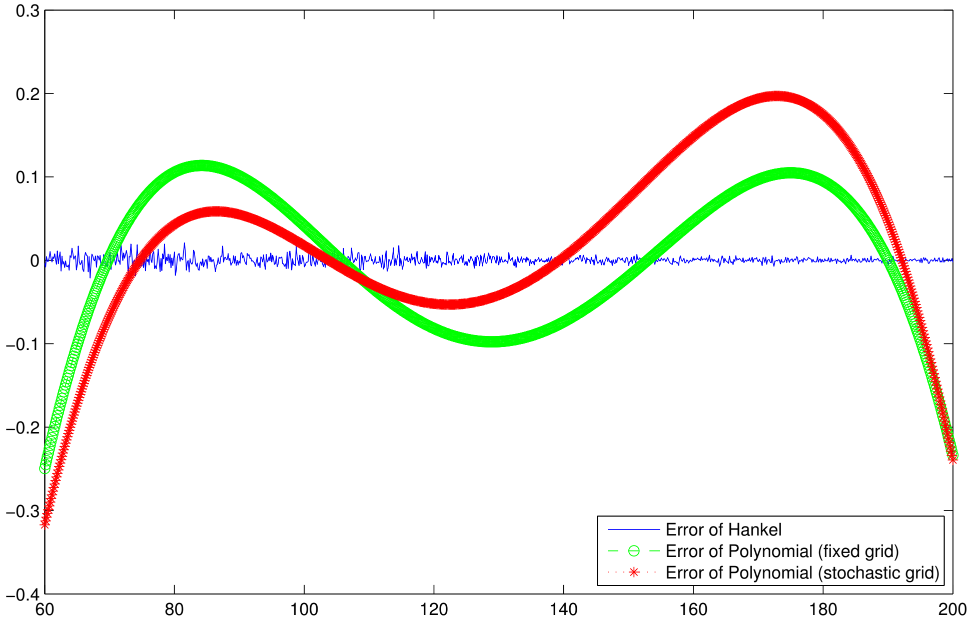

FIGURE 22. Estimation errors for polynomial versus exponential approximations with $m = 10,000,000$

6. STUDY FINDINGS ON CASE II

6.1. **Overview of nested structure.** In the second case, we shall perform an AG-43 stochastic scenario amount calculation for a line of business with a "clearly defined hedging strategy." A detailed account of AG-43 standards can be found in Gorski and Brown [3]. To make the report self-contained, we briefly outline the general procedure of the AG-43 reserving method before identifying the structure of the nested simulation. A full AG-43 calculation requires both standard scenario amount (SSA) and stochastic scenario amount (CTE) calculations. Keep in mind that the purpose of this study to test out various nested simulation techniques. We shall not consider the SSA calculation in this study, as it is a projection based on a single scenario and does not involve any stochastic component. The stochastic scenario amount under the AG-43 is calculated in three key steps. The projection begins with starting assets consisting of hedging portfolio, general account assets and separate account assets.

(1) (Projection of scenarios) Use either AAA's prepackaged scenarios or internal scenario generators to project cash flows of a line of business (on both the liability and asset sides).

(2) (Pathwise accounting procedure) Calculate the accumulated surplus/deficiency at the end of each projection period and determine the greatest present value of accumulated deficiencies (GPVAD) for each scenario. The *scenario amount* is determined by the sum of starting assets and the GPVAD.

(3) (Application of risk measures) Collect scenario amounts for all scenarios and apply the 70% conditional tail expectation risk measure to the sample of scenario amounts to determine the *stochastic scenario amount*.

Hedging programs are implemented and should be included in cash flows if the reporting company has a "clearly defined hedging strategy" (CDHS). A precise definition of the CDHS can be found in Gorski and Brown [3]. The stochastic scenario amount is determined by

$$\text{CTE (best efforts)} \times E + \text{CTE (excluding CDHS)} \times (1 - E).$$

The CTE (best efforts) is calculated from the above-mentioned steps where cash flows reflect the implementation of the hedging program, whereas the CTE (excluding CDHS) is calculated from the same steps except that all existing assets are assumed to be held to maturity and no further actions of hedging are taken. The "effectiveness factor" E is determined based on the sophistication of its hedging projections and must be capped at 70%. As we shall see below, nested simulations appear only in the AG-43 procedure with the consideration of a hedging program, where most computational burden rises from the calculation of the CTE (best efforts). Therefore, we shall focus on the efficiency of computation with CTE (best efforts).

In this test case, we perform CTE calculations for variable annuity contracts with a guaranteed lifetime withdrawal benefit (GLWB) rider. All policyholders in the cohort under consideration are 61 years old at the time of issue, and the projection is performed at the end of four years after issue. All policyholders invest their purchase payments (premiums) into a single fund, which will be linked to an equity index, such as the S&P 500. Each policyholder is provided with a *nominal account* to keep track of their investments and equity returns as well as a *guarantee base* to be used as the basis of calculating free withdrawal amounts. Starting from the valuation date, policyholders take withdrawals up to the maximum allowable amount, 4% of its guarantee base per year. Withdrawals are taken out of policyholders' nominal accounts and do not reduce the guarantee base. Upon the death of a policyholder, his/her beneficiary will receive the remaining balance of the policyholder's nominal account. The GLWB rider also offers a combination of a roll-up option, under which the guarantee base accrues interest, and step-up (ratchet) options, under which the guarantee base would be always matched to the account value, should the latter exceeds the former. For simplicity, we shall consider a flat deterministic yield curve for interest rate. A numerical example of the dynamics of a policyholder's account and guarantee base will be provided in Section 6.3. From an insurer's standpoint, the investments from the cohort are aggregated and considered as a single fund under stochastic scenarios. In the following discussion, we shall refer to the aggregated investment as *separate account*, used in contrast with an insurer's own general account.

The computation of the CTE (best efforts) will be carried out in a nested setting as follows.

(1) Outer loop (AG-43 reserving: CTE (best efforts))

We use a geometric Brownian motion (independent lognormal model) to project equity returns over the next 30 or 50 years. Under each scenario of equity returns, we determine cash flows from separate accounts (withdrawal payments, interest on surplus, rider charges, management fees) and the cash flows from the hedging portfolio (buy and sell of index futures and bonds). The change in surplus is determined by the following recursive relation over each period:

(6.1)

$$\begin{aligned}\text{Change in surplus} = {} & \text{Fee income} + \text{Surrender charge} - \text{GLWB withdrawals} - \text{Expenses}\\ & + \text{Investment income on cash flows} - \text{Change in asset values}\\ & + \text{Investment income on surplus.}\end{aligned}$$

Asset values are determined by starting assets, which are assumed to be long-term bonds, and the value of a hedging portfolio. An inner-loop calculation is invoked every time a dynamic hedging portfolio is rebalanced.

In the example, we also consider dynamic lapse rates, which vary with the account value and guarantee base. The stochastically determined lapse rates are included to capture policyholder behaviors where persistence is high when the guarantee is much higher than account value.

(2) Inner loop (Hedging program)

For simplicity, the hedging program under consideration is based on a delta neutral strategy, which only utilizes index futures and bonds. In the ensuing numerical example, we consider both biweekly and quarterly hedging, with the former aimed at illustrating accuracy of hedging and the latter aimed at reducing computational efforts. The initial net value of the hedging portfolio is assumed to be zero. Keep in mind that the risk-neutral value of the GLWB rider at any given point in time depends on four factors: account value, guarantee base, time to maturity and (dynamic) survivorship. At each point in time, we project the future evolution of account value, guarantee base conditional on the then-current account value and guarantee base. We record two components of benefit payments to policyholders:

(a) Quarterly withdrawals: This amount is always assumed to be 4% of the then-current guarantee base.

(b) Return of account values upon death and return of cash values upon surrender.

The delta, $\Delta,$ of the GLWB rider, which determines how many units of index futures to hold in the hedging portfolio, is calculated as follows: (1) Evaluate the risk-neutral value of the GLWB rider with the then-current account value and the then-current guarantee base. (2) Shock the then-current account value by 1% and evaluate the risk-neutral value of the GLWB rider. (3) Determine the delta by the difference quotient of risk-neutral values.

In the corresponding outer loop, the hedging portfolio always consists of Δ shares of index futures and a certain number of bonds. Over each period, we shall short-sell or buy just enough one-month bonds in order to finance Δ shares of the futures.

(3) Final result

In each outer-loop calculation, we observe the evolution of the present value of the accumulated deficiency over the projection period of 50 years. Record the largest value over time for each outer loop, which is the scenario amount. The CTE(best efforts) is determined by the average of 30% of the largest scenario amounts.

6.2. **Conceptual comparison of Monte Carlo and PDE methods.** Before we dive into the technical details of the GLWB modeling, it is helpful to compare the general methodology of Monte Carlo simulations with the PDE method to be introduced in this section.

In general, the Monte Carlo method of risk-neutral and real-world valuation can be summarized in the following three steps:

Step 1: Use either prepackaged scenarios or internally built stochastic models (called economic scenario generators) for all risk factors driving the insurer's asset and liability portfolio. These stochastic models have to be calibrated to meet regulatory standards. Generate a variety of sample paths of the stochastic models over a projection period.

Step 2: Under each scenario of account values, follow certain accounting standards to determine the accumulated profit/deficiency for the entire projection length.

Step 3: Repeat Step 2 for each scenario many times to generate an empirical distribution of accumulated surplus/deficiency or other performance metrics. Apply order statistics to estimate certain risk metrics, such as quantile/conditional tail expectation, which form the basis of reserve or capital requirements.

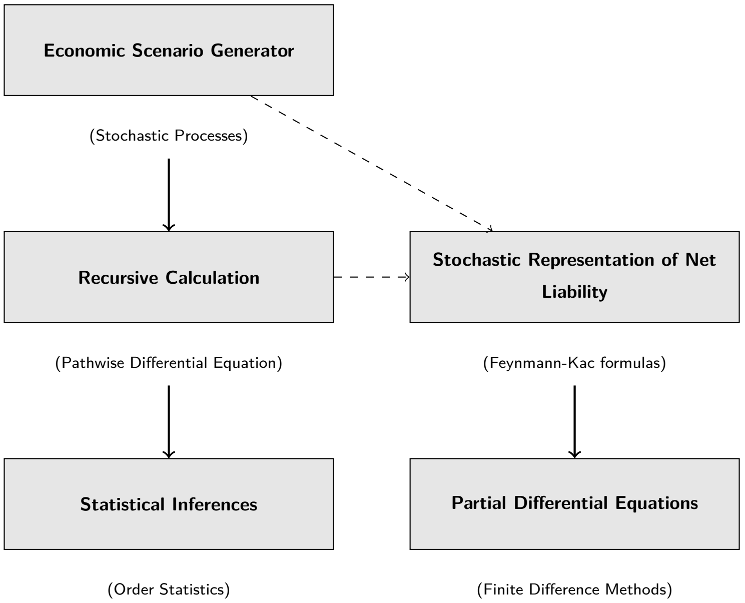
Economic Scenario Generator
(Stochastic Processes)
Recursive Calculation
Stochastic Representation of Net Liability
(Pathwise Differential Equation)
(Feynmann-Kac formulas)
Statistical Inferences
Partial Differential Equations
(Order Statistics)
(Finite Difference Methods)

FIGURE 23. Two computational methods

The logical flow of the Monte Carlo method is shown in the left column of Figure 23. The current market practice and regulatory standards such as the variable annuity reserving method (VA-CARVM) are largely developed with the implementation of Monte Carlo methods at the heart of their designs. As the Monte Carlo method is the tool for implementation by default, designs of accounting rules make it difficult to use other alternative deterministic methods. Therefore, alternative deterministic methods are largely viewed as approximations or “technical tricks.”

Nevertheless, many deterministic methods can provide cost-efficient solutions, even under the framework designed for Monte Carlo simulations. For example, discrete-time calculations used in financial reporting are often based on pathwise -defined recursive relations. We can integrate this information with the underlying stochastic models to determine a stochastic representation of an insurer’s accumulated profit/deficiency.

As shown in the right column of Figure 23, we could utilize this stochastic representation to find an alternative deterministic method to compute the corresponding risk measures. This goal can be achieved through a technique, known as the Feynmann-Kac formula in the physics and applied mathematics literature. The resulting partial differential equations (PDEs) can be solved by either analytical methods or numerical methods. This approach is outlined in the right column of Figure 23. There are several advantages of the Feynmann-Kac approach over the Monte Carlo simulations approach:

(1) Stochastic representations of insurers’ liabilities provide intuitive interpretations of all contributing factors. For example, in (6.8), one can easily identify the accumulated values of benefit outgo and fee income.

(2) Simulation methods can be very time consuming, due to the repeated sampling of stochastic scenarios. In particular, various risk metrics used in practice measure only rare events, and hence their estimators can have

large variances. Analytical or numerical solutions to PDEs are deterministic algorithms, which can be much more efficient.

(3) Monte Carlo simulation allows us only to find a solution at a single point of initial economic conditions, whereas numerical PDE methods require marching through solutions at multiple time points and multiple grid points of economic conditions. These intermediate solutions can be used to produce very efficient algorithms for sensitivity testing of risk measures for insurers' liabilities or sensitivity measures for risk-neutral values as such as the Greeks.

Applications of numerical PDE methods have been studied in Feng [4]and Feng and Huang [5]. This study is an extension of their work to a general setting of the GLWB.

6.3. **Modeling GLWB liabilities: practice versus mathematical formulation.** The guaranteed lifetime withdrawal benefit (GLWB) guarantees a policyholder the ability to withdraw up to a maximum percentage of the guarantee base for the whole life, regardless of the balance in his/her nominal account.

Policy year	Investment return	Account value before withdrawal	Annual withdrawal	Account value after withdrawal	Benefit base
0	-	-	-	10,000.00	10,000.00
1	5%	10,500.00	500.00	10,000.00	10,000.00
2	10%	11,000.00	500.00	10,500.00	10,500.00
3	5%	11,025.00	525.00	10,500.00	10,500.00
4	10%	11,550.00	525.00	11,025.00	11,025.00
5	-20%	8,820.00	551.25	8,268.75	11,025.00
6	-10%	7,441.88	551.25	6,890.63	11,025.00
7	-10%	6,201.56	551.25	5,650.31	11,025.00
8	5%	5,932.83	551.25	5,381.58	11,025.00
9	10%	5,919.74	551.25	5,368.49	11,025.00
10	20%	6,442.18	551.25	5,890.93	11,025.00
11	-5%	5,596.39	551.25	5,045.14	11,025.00
12	-15%	4,288.37	551.25	3,737.12	11,025.00
13	-10%	3,363.40	551.25	2,812.15	11,025.00
14	10%	3,093.37	551.25	2,542.12	11,025.00
15	-15%	2,160.80	551.25	1,609.55	11,025.00
16	5%	1,690.03	551.25	1,138.78	11,025.00
17	-10%	1,024.90	551.25	473.65	11,025.00
18	-5%	449.97	551.25	0.00	11,025.00
19	-	0.00	551.25	0.00	11,025.00
20	-	0.00	551.25	0.00	11,025.00

TABLE 6. GLWB with a step-up option

Let us consider a numerical example of the payoffs from a GLWB rider in Table 6. The second column gives a particular sequence (a sample path) of yearly rates of return from the underlying equity index over the 20-year period. The return rates are chosen for the convenience of illustration in Table 6. The example assumes an initial investment of $\$10,000$, and the guarantee base is set to the same amount at the start. Assume that the policyholder elects the option to withdraw 5% of the guarantee base every year. At the end of each policy year, a rate of investment return is recorded and applied to the policyholder's account from the previous year. For example, the equity index increases by 5% in the first year, and so does the account value. Then 5% of the guarantee base from the previous year, 5% $\times$ $\$10,000 = \$500$, is withdrawn from the account. The same recursive relation for each year goes on for the life time of the policyholder. Note that the policyholder's account receives a boost of financial return in the second year resulting in an account value after withdrawal $\$10,500$, which is higher than the benefit base from the previous period $\$10,000$. Then the "step-up" option takes effect, and the benefit base is automatically reset to the account value. In other words, the step-up option allows the policyholder to lock in the maximum account value to date with the guarantee base. Note that when the policyholder dies, the remaining balance of his/her account would be paid to a designated beneficiary. For example, if the person dies at the end of the sixth year, then the beneficiary is entitled to $\$6890.63$. If the person dies at the end of the 20-th year, then there is no remaining balance to be paid. Note, however, the return of remaining balance is not a benefit payment from the GLWB rider, but rather part of the policyholder's own investment:

To formulate this problem, we introduce the following notation.

- F_t, the market value of a policyholder's subaccount at $t \geq 0$. F_0 is considered to be the initial purchase payment (premium) invested at the start of the contract. For simplicity, we consider the single-premium variable annuity where no additional purchase payment is made into the account.
- G_t, the guarantee base at time $t \geq 0$. The guarantee base may also accrue compound interest at the so-called roll-up rate.
- ρ, the annualized roll-up rate for the guarantee base.
- Y_t, the in-the-moneyness ratio. While there are many ways to define the moneyness, we use the ratio $Y_t = F_t/G_t \in [0, 1]$.
- S_t, the market value of the underlying equity index or fund at t. If more than one fund is involved, this is considered to be the portfolio value of all funds.
- $\mathfrak{B}_t$, the market value of risk-free bonds at time t.
- p, the percentage of the investment account invested in equities. For example, a "conservative" allocation portfolio with 40% stocks and 60% bonds would have $p = 40$%.
- m, the annualized rate at which account-value-based fees are deducted from the separate account. As the fees are used to compensate for costs of covering mortality risks and expenses, they are often referred as the mortality and expenses (M&E) fees. Except for purchase-payments-based withdrawal charges, all contract fees and expenses are typically calculated and accrued on daily basis.
- m_w, the annualized rate of the guarantee-based-rider-charge, allocated to fund the GLWB rider.
- r, the continuously compounding annual risk-free interest rate.
- σ, annualized volatility per annum of the underlying equity index.

- ${}_tp_x$, the survival probability after t years of a policyholder of age x at issue. However, as we shall consider dynamic policyholder behavior later, it would be more appropriate to view ${}_tp_x$ as the proportion of in-force policies in the initial population.
- μ_t^d, the instantaneous death rate at time t.
- μ_t^l, the instantaneous lapse rate at time t.
- h, the free partial withdrawal rate per annum.
- c_t, the account-value-based surrender charge rate at time t.
- V_t, the risk-neutral value of the GLWB.

One should keep in mind that all annualized rates, such as ρ, r, m, are all nominal rates with compounding frequency according to their uses. For example, if the fees are taken on a daily basis, then m is the corresponding annual nominal fee rate compounded daily. Regardless of different frequencies in examples, all rates should be equivalent to their corresponding annual effective rates.

6.3.1. *Outer loop: surplus calculation.* Before we introduce the continuous-time model, let us consider how the model is formulated with a typical discrete-time model. Note that in practice a discrete-time model can be implemented with a spreadsheet or computer algorithm. To make a connection between the discrete-time model (recursive calculation) and the continuous-time model, we shall divide each year into n periods. For example, if the projection is done on a quarterly basis, then we are assuming $n = 4$.

Section 1: Fund values before decrements

Discrete-time model (recursive calculation)

Under the GLWB, the amount of systematic withdrawals per period is typically determined by a pre-specified percentage of a guarantee base. It is typical that the starting value of the guarantee base matches with the initial premium $G_0 = F_0$, and the evolution of the guarantee base over each period is based on either of the following models.

(1) Roll-up option

Under the roll-up option, the guarantee base accrues interest throughout the term of the GLWB rider. Then the guarantee base at the end of $k+1$ periods is determined by a recursive relation from that of the $k-$th period:

$$G_{(k+1)/n} = G_{k/n}\left(1 + \frac{\rho}{n}\right), \qquad \text{for } k = 0, 1, \cdots .$$

Note that this recursive relation implies that

$$G_{k/n} = G_0\left(1 + \frac{\rho}{n}\right)^k .$$

(2) Step-up option

It is also known as the ratchet option. The guarantee base can increase with the policyholder's investment account at the end of each period. However, the guarantee base would never decrease, even if the investment account loses value:

$$G_{(k+1)/n} = \max\left\{G_{k/n}, F_{(k+1)/n}\right\}, \qquad \text{for } k = 0, 1, \cdots .$$

Observe that this recursive relation leads to the representation

$$G_{k/n} = \max_{j=0,1,\cdots,k}\left\{F_{j/n}\right\} .$$

(3) A combination

This option offers guaranteed compound growth on the guarantee base and also allows the guarantee base to “lock in” gains from the policyholder’s designated investment:

$$G_{(k+1)/n} = \max\left\{G_{k/n}\left(1+\frac{\rho}{n}\right), F_{(k+1)/n}\right\}, \qquad \text{for } k = 0, 1, \cdots .$$

Observe that this option also has a representation

$$G_{k/n} = \left(1+\frac{\rho}{n}\right)^{k} \max_{j=0,1,\cdots,k}\left\{\left(1+\frac{\rho}{n}\right)^{-j} F_{j/n}\right\}, \tag{6.2}$$

which can be proved by mathematical induction. For example, in the induction step,

$$\begin{aligned} G_{(k+1)/n} &= \max\left\{\left(1+\frac{\rho}{n}\right)^{k+1} \max_{j=0,1,\cdots,k}\left\{\left(1+\frac{\rho}{n}\right)^{-j} F_{j/n}\right\}, F_{(k+1)/n}\right\} \\ &= \left(1+\frac{\rho}{n}\right)^{k+1} \max\left\{\max_{j=0,1,\cdots,k}\left\{\left(1+\frac{\rho}{n}\right)^{-j} F_{j/n}\right\}, \left(1+\frac{\rho}{n}\right)^{-(k+1)} F_{(k+1)/n}\right\}, \end{aligned}$$

which agrees with (6.2) with k replaced with $k+1$.

There are also other combinations in the market. For example, a common practice is to offer the greater of a step-up option and a roll-up option:

$$G_{k/n} = \max\left\{G_0\left(1+\frac{\rho}{n}\right)^{k}, \max_{j=0,1,\cdots,k}\{F_{j/n}\}\right\}.$$

Although we do not use this combination in the following numerical example, it also be accommodated by all techniques discussed in this report.

Keep in mind, however, that the guarantee base is a nominal account that serves only as a base for determining withdrawal amounts. The actual withdrawals, which are typically a fixed percentage, denoted by h, of the then-current guarantee base, are taken out of the policyholder’s own investment account (not the guarantee base). There are two common types of fees that appear in various versions of the product:

(1) Account-value-based fees

Traditional product designs utilize this type of fee structure, under which M&E fees are a pre-specified fixed percentage of the policyholder’s account value in each period. We shall denote the annualized rate of account-value-based fees by m.

(2) Guarantee-based fees

It has become increasingly common that rider charges, allocated to fund the GLWB, are taken as a pre-specified fixed percentage of the guarantee base in each period. The rationale is to provide a better alignment of fee incomes and (guarantee-based) withdrawal benefits. We shall denote the annualized rate of guarantee based fees by m_w. (Note: in this case, the rider charge rate m_w is not part of the M&E fee rate m.)

In a recursive calculation, we would observe that the incremental change per period for the account value is given by

(6.3)

$$F_{(k+1)/n} - F_{k/n} = p\frac{S_{(k+1)/n} - S_{k/n}}{S_{k/n}} F_{k/n} + (1-p)\frac{\mathfrak{B}_{(k+1)/n} - \mathfrak{B}_{k/n}}{\mathfrak{B}_{k/n}} F_{k/n} - \frac{m_w + h}{n} G_{k/n} - \frac{m}{n} F_{k/n}.$$

Immediately after the policyholder’s account hits zero, the recursive relation is no longer valid as there is no money left to invest. Therefore, we would write for $k = 0, 1, \cdots$

$$F_{(k+1)/n} = \max\left\{\left[p\frac{S_{(k+1)/n}}{S_{k/n}} + (1-p)\frac{\mathfrak{B}_{(k+1)/n}}{\mathfrak{B}_{k/n}}\right] F_{k/n} - \frac{m_w + h}{n} G_{k/n} - \frac{m}{n} F_{k/n}, 0\right\}.$$

Continuous-time model

Although the above formulation works for all model assumptions of equity indices, we shall consider a specific model for the equity index to facilitate continued discussion of continuous-time models. For simplicity, we assume that the equity index is driven by a geometric Brownian motion, also known as the independent lognormal model in the industry. Under the risk-neutral measure,

$$\mathrm{d}S_t = rS_t\mathrm{d}t + \sigma S_t\mathrm{d}B_t.$$

The money market account accrues interest, $\mathfrak{B}(t) = \mathfrak{B}(0)e^{rt}$, or equivalently

$$\mathrm{d}\mathfrak{B}_t = r\mathfrak{B}_t\mathrm{d}t.$$

Keep in mind, however, that the rest of the analysis can be extended to more general models.

If we write the time increment $\Delta t = 1/n$ and $t = k/n$ in (6.3), then

$$F_{t+\Delta t} - F_t = p\frac{S_{t+\Delta t} - S_t}{S_t}F_t + (1-p)\frac{\mathfrak{B}_{t+\Delta t} - \mathfrak{B}_t}{\mathfrak{B}_t}F_t - (m_w + h)G_t\Delta t - mF_t\Delta t.$$

When we shrink the time period Δt to zero, the continuous-time version of the recursive relation in (6.3) becomes the stochastic differential equation

$$\begin{aligned}\mathrm{d}F_t &= p\frac{F_t}{S_t}\mathrm{d}S_t + (1-p)\frac{F_t}{\mathfrak{B}_t}\mathrm{d}\mathfrak{B}_t - (m_w + h)G_t\mathrm{d}t - mF_t\mathrm{d}t \\ &= [(r-m)F_t - (m_w + h)G_t]\mathrm{d}t + p\sigma F_t\mathrm{d}B_t. \end{aligned} \tag{6.4}$$

As the projection period shrinks to zero, we note that the growth of guarantee base in (6.2) is turning into a continuous-time stochastic process

$$G_t = e^{\rho t}\sup_{0\le s\le t}\left\{e^{-\rho t}F_t\right\}.$$

One can prove that the stochastic process can be represented as a solution to the stochastic differential equation

$$\mathrm{d}G_t = G_t\mathrm{d}L_t + \rho G_t\mathrm{d}t,$$

where L_t increases only when the in-the-moneyness ratio $Y_t = 1$. Note that ρ in this stochastic differential equation is the corresponding continously compounding (nominal) rate. Without getting into too much technical detail, we can interpret the stochastic equation as follows. The instantaneous change of the guarantee base consists of two components: (1) the instantaneous rise of the guarantee base to match the account value if the in-the-moneyness tends to exceed one; and (2) the instantaneous growth due to the roll-up interest rate.

Using Itô's formula, one can also show that the in-the-moneyness ratio $Y_t = F_t/G_t$ also satisfies the stochastic differential equation

$$\mathrm{d}Y_t = [(r-m-\rho)Y_t - (m_w + h)]\mathrm{d}t + p\sigma Y_t\mathrm{d}B_t - \mathrm{d}L_t.$$

The interpretation of the dynamics of in-the-moneyness ratio process is also clear: (1) The $\mathrm{d}t$ term represents the instantaneous change in the in-the-moneyness ratio due to changes in account value and changes in guarantee base. Note that the account F_t grows at the rate $r - m$ per dollar per time unit, and the guarantee base G_t grows at the roll-up rate ρ per dollar per time unit. The net effect on the in-the-moneyness ratio $Y_t = F_t/G_t$ is the growth of $r - m - \rho$ per dollar per time unit. In addition, as shown in (6.4), fees and withdrawals are taken out of the account F_t by the rate $(m_w + h)G_t$ per time unit at time t. Therefore, the in-the-money ratio decreases by the fixed rate $m_w + h$ per time unit at time t. (2) The $\mathrm{d}L_t$ term is the outcome of the continuous-time "step-up" feature by which the guarantee base is at least as big as the account value, and hence in-the-moneyness ratio cannot exceed upper limit 1. The process L_t acts as a regulator to take away the extra amount so that Y_t stays at 1 whenever it has the tendency to move above 1.

Section 2: Fund values after decrements

Discrete-time model (recursive calculation)

As we consider only the withdrawals to inforce policies, there are two sources of decrements to be considered:

- ${}_tq_x^d$, the probability that a policyholder at age x dies within t periods.
- ${}_sq_t^l$, the probability that a policy still in force at time t lapses within s periods.
- ${}_sq_t^b$, the probability that a policy still in force at time t lapses within s periods without the consideration of dynamic policyholder behavior
- ${}_tp_x$, the probability that the policy from a policyholder at age x is still in force after t periods.

The values of ${}_tq_x^d$ are typically provided in a life table for integer values x and t. We may use fractional year assumptions to infer their values at non-integer values.

We consider ${}_tq^b$ to be base lapse rates, which are estimates of lapse rates in the absence of dynamic policyholder behavior. As the in-the-moneyness ratio increases and is close to 1, we expect policyholders to feel confident of their own investment and more likely to surrender their contracts. As the in-the-moneyness ratio decreases, policyholders are more likely to rely on guarantee benefits from the GLWB rider and hold on to their policies. Therefore, the dynamic lapse rate at time t is the product of the base lapse rate and the dynamic factor, which is a function f of the in-the-moneyness ratio Y_t,

$$
{}_{\triangle t}q_t^l = {}_{\triangle t}q_t^b \, f(Y_t). \tag{6.5}
$$

Note that survival probabilities are usually calculated from the recursive relation

$$
{}_{(k+1)/n}p_x = {}_{k/n}p_x(1 - {}_{1/n}q_{x+k/n}^d)(1 - {}_{1/n}q_{k/n}^l), \qquad k = 1, 2, \cdots . \tag{6.6}
$$

Readers should be reminded that the survivorship model $\{ {}_tp_x, t \geq 0\}$ is path dependent on equity returns through the in-the-moneyness ratio process $\{Y_t, t \geq 0\}$.

Continuous-time model

In the continuous-time model, we consider the instantaneous rate of mortality

$$
\mu_{x+t}^d = \frac{\partial({}_tq_x^d)/\partial t}{{}_tp_x}.
$$

We can infer the instantaneous death rates $\{\mu_s^d, s \geq 0\}$ from a life table with fractional age assumptions. For example, the constant force of mortality assumption implies that

$$
\mu_s^d = -\ln(1 - q_x^d), \qquad \text{for all integer } x \text{ and } 0 \leq s \leq 1.
$$

In other words, for integer x and $t \geq 0$,

$$
{}_tq_x^d = 1 - e^{-\int_0^t \mu_{x+s}^d \mathrm{d}s}.
$$

Similarly, we can define the instantaneous lapse rates $\{\mu^l_s, s \geq 0\}$ and the instantaneous base lapse rates $\{\mu^b_s, s \geq 0\}$:

$$\mu^l_t = \left.\frac{\partial({}_sq^l_t)}{\partial s}\right|_{s=0}, \qquad \mu^b_t = \left.\frac{\partial({}_sq^b_t)}{\partial s}\right|_{s=0}.$$

Observe that (6.5) can be rewritten as

$$\frac{{}_{\Delta t}q^l_t}{\Delta t} = \frac{{}_{\Delta t}q^b_t}{\Delta t}\, f(Y_t),$$

which implies by taking the limit $\Delta t \to 0$ that

$$\mu^l_t = \mu^b_t f(Y_t).$$

To see the analogue of (6.6) in the continuous time model, we set $t = x + k/n$ and $\Delta t = 1/n$ in (6.6) and then let $\Delta t \to 0$. It follows that

$$\mathsf{d}_t p_x = -\,{}_tp_x(\mu^l_t + \mu^d_{x+t})\mathsf{d}t.$$

Section 3: Income statement

The last section discussed computing the accumulated surplus/deficiency under each projection of account values. Let us use the following notation to capture the cash flows in each period:

- R_t, the accumulated surplus at time t
- c_t, the surrender charge at time t
- E_t, the expense per time unit at time t, which may include percentage of AV unit maintenance/overhead expense and per policy unit maintenance

Discrete time model (recursive calculation)

Before consideration of hedging, the incremental changes in surplus are determined by the recursive relation (6.1), which can be translated as follows:

$$\begin{aligned}(6.7)\qquad R_{t+\Delta t} - R_t = (1 + r\Delta t)\Big[&{}_tp_x(mF_t + m_wG_t)\Delta t I(F_t > 0)\\ &+ {}_tp_x\,{}_{\Delta t}q^l_{x+t}c_tF_tI(F_t > 0) - {}_tp_xhG_t\Delta tI(F_t \leq 0) - {}_tp_xE_t\Delta tI(F_t > 0)\Big] + rR_t\Delta t,\end{aligned}$$

where I is the indicator function for which $I(A) = 1$ if A is true and $I(A) = 0$ otherwise.

Continuous-time model

Taking Δt to zero, we obtain immediately a pathwise-defined ordinary differential equation,

$$\begin{aligned}\mathsf{d}R_t = \Big[&rR_t - {}_tp_xhG_tI(F_t \leq 0) + {}_tp_x(m_wG_t + mF_t)I(F_t > 0)\\ &+\mu^l_tf(Y_t)\,{}_tp_xc_tF_tI(F_t > 0) - {}_tp_xE_tI(F_t > 0)\Big]\mathsf{d}t,\end{aligned}$$

together with the initial condition $R_0 = 0$, which yields the solution

$$\begin{aligned}R_t = e^{rt}\Bigg[&\int_0^t e^{-rs}\,{}_sp_x(m_wG_s + mF_s)I(F_s > 0)\mathsf{d}s + \int_0^t e^{-rs}\mu^l_sf(Y_s)\,{}_sp_xc_sF_sI(F_s > 0)\mathsf{d}s\\ &-\int_0^t e^{-rs}\,{}_sp_x\,h\,G_sI(F_s \leq 0)\mathsf{d}s - \int_0^t e^{-rs}E_sI(F_s > 0)\mathsf{d}s\Bigg].\end{aligned}$$

6.3.2. *Inner loop: Risk-neutral valuation of GLWB liability.* Let us now consider the risk-neutral value of a GLWB rider. Recall that the GLWB guarantees systematic withdrawals until the policyholder's death regardless of whether or not his/her investment account is depleted. There are four components of cash flows from the insurer's point of view:

(1) Withdrawal benefits

Observe that the GLWB rider incurs cost when the policyholder continues to withdraw after the account is depleted. Hence, the present value of all GLWB benefit payments is given by

$$\sum_{k=n\tau}^{\infty} e^{-r(k+1)/n}\ {}_{k/n}p_x \frac{h}{n} G_{k/n},$$

where the first time when the account is exhausted, called the *ruin time*, is given by

$$\tau := \text{min}\left\{\frac{k}{n} > 0 : F_{k/n} = 0\right\}.$$

In the discrete-time case, there is the possibility that the last withdrawal payment before the account is depleted comes partly from the the policyholder's account and partly from the insurer's general account. We ignore the small payment from the insurer here, because this payment appears in the continuous-time model to be introduced next.

(2) Fee incomes

On the revenue side, the GLWB is funded by the collection of rider charges, which include account-value-based fees and guarantee-based fees. The present value of total rider charges is given by

$$\sum_{k=0}^{n\tau} e^{-r(k+1)/n}\ {}_{k/n}p_x \left(F_{k/n}\frac{m}{n} + G_{k/n}\frac{m_w}{n}\right).$$

(3) Surrender charges

If a policyholder voluntarily surrenders a contract, then the remaining balance of the investment account is subject to a surrender charge, which is determined by the surrender charge rate:

$$\sum_{k=0}^{n\tau} e^{-r(k+1)/n}\ {}_{k/n}p_x\ {}_{1/n}q^{l}_{x+k/n}\ c_{(k+1)/n}F_{(k+1)/n}.$$

(4) Expenses

The present value of all expenses is given by

$$\sum_{k=0}^{n\tau} e^{-r(k+1)/n}\ {}_{k/n}p_x\ \frac{E_{(k+1)/n}}{n}.$$

Note, however, although not written explicitly, the survival rate ${}_{k/n}p_x$ in all four components is path dependent as shown in (6.6).

Recall the definition of a Riemann integral from calculus. If we divide each time unit into n subintervals and set the points $x_0 = 0, x_1 = 1/n, x_2 = 2/n, \cdots$, then the integral is defined to be

$$\int_0^{\infty} f(t)\mathrm{d}t = \lim_{1/n\to 0}\sum_{k=0}^{\infty} f(x_k)(x_{k+1} - x_k) = \lim_{n\to\infty}\sum_{k=0}^{\infty} f\left(\frac{k}{n}\right)\left(\frac{1}{n}\right).$$

It follows immediately that as the projection period shrinks to zero, i.e., $n \to \infty$, the withdrawal benefits can be written as

$$\lim_{n\to\infty}\sum_{k=0}^{\infty} e^{-r(k+1)/n}\ {}_{k/n}p_x\, h\, G_{k/n}\left(\frac{1}{n}\right) I(F_{k/n} > 0) = \int_0^{\infty} e^{-rt}\ {}_tp_x\, h\, G_t I(F_t \leq 0)\mathrm{d}t.$$

The fee incomes can be written as

$$\lim_{n\to\infty}\sum_{k=0}^{\infty} e^{-r(k+1)/n}\ {}_{k/n}p_x \left(F_{k/n}\frac{m}{n}+G_{k/n}\frac{m_w}{n}\right) I(F_{k/n}>0)=\int_0^{\infty} e^{-rt}\mu^d_{x+t}\ {}_tp_x(mF_t+m_wG_t)I(F_t>0)\mathrm{d}t.$$

Similarly, the surrender charges can be written as

$$\lim_{n\to\infty}\sum_{k=0}^{\infty} e^{-r(k+1)/n}\ {}_{k/n}p_x\ \frac{{}_{1/n}q^l_{x+k/n}}{1/n}\ c_{(k+1)/n}F_{(k+1)/n}\frac{1}{n}I(F_{k/n}>0)=\int_0^{\infty} e^{-rt}\mu^l_t f(Y_t)\ {}_tp_x c_t F_t I(F_t>0)\mathrm{d}t,$$

and the expenses can be written as

$$\lim_{n\to\infty}\sum_{k=0}^{\infty} e^{-r(k+1)/n}\ {}_{k/n}p_x\ E_{(k+1)/n}\frac{1}{n}I(F_{k/n}>0)=\int_0^{\infty} e^{-rt}\ {}_tp_x\ E_t I(F_t>0)\mathrm{d}t.$$

Total product liability hedging

If we consider all cash flows from an insurer's perspective, then the risk-neutral value of the insurer's total product liability in the continuous time model is given by

$$\begin{aligned} V_0=\mathbb{E}\bigg[&\int_0^{\infty} e^{-rt}\ {}_tp_x\, h\, G_t I(F_s\le 0)\mathrm{d}t-\int_0^{\infty} e^{-rt}\ {}_tp_x(m_wG_s+mF_t)I(F_t>0)\mathrm{d}t\\ &-\int_0^{\infty} e^{-rt}\mu^l_t f(Y_t)\ {}_tp_x c_t F_t I(F_s>0)\mathrm{d}t-\int_0^{\infty} e^{-rt}\ {}_tp_x E_t I(F_t>0)\mathrm{d}t\bigg].\end{aligned}$$

In this formulation, we included only the cash flows in and out of the insurer's general account (not a separate account). This is not to be confused with the entire product from a policyholder's perspective, which could also be calculated using a very similar approach.

From now on, we shall ignore the expenses for simplicity. Note, however, that the expenses term can be easily incorporated in the following calculations. It is known from no-arbitrage option pricing theory that the risk-neutral value of total liability to the insurer at time t is given by the conditional expectation of future cash flows,

$$\begin{aligned} V_t:=\mathbb{E}\bigg[&\int_t^{\infty} e^{-r(s-t)}\ {}_sp_x\, h\, G_s I(F_s<0)\mathrm{d}s-\int_t^{\infty} e^{-r(s-t)}\ {}_sp_x(m_wG_s+mF_s)I(F_s>0)\mathrm{d}s\\ &-\int_t^{\infty} e^{-r(s-t)}\mu^l_s f(Y_s)\ {}_sp_x c_s F_s I(F_s>0)\mathrm{d}s\bigg|\mathcal{F}_t\bigg],\end{aligned} \tag{6.8}$$

where $\{\mathcal{F}_t, t\ge 0\}$ is the natural filtration of the Brownian motion driving the dynamics of the equity index process. We shall refer to the risk-neutral value of net total product liability as total liability for short. It follows from the strong Markov property of the underlying process $(F_t, G_t,\ {}_tp_x)$ that there exists a smooth function $v(t,x,y,z)$ that

$$V_t=v(t,F_t,G_t,\ {}_tp_x). \tag{6.9}$$

GLWB rider hedging

In practice, we may not consider cash flows that are not directly generated from the GLWB rider to be part of a hedging program, such as M&E fees or expenses and surrender charges. Similar to the earlier model of total liability hedging, the risk-neutral value of the GLWB net liability at time t is given by the conditional expectation

$$V_t:=\mathbb{E}\bigg[\int_t^{\infty} e^{-r(s-t)}\ {}_sp_x\, h\, G_s I(F_s<0)\mathrm{d}s-\int_t^{\infty} e^{-r(s-t)}\ {}_sp_x m_w G_s I(F_s>0)\mathrm{d}s\bigg|\mathcal{F}_t\bigg].$$

Note this formula is actually a special case of (6.8) with $m=0$ and $c_t=0$ for all t.

For brevity, we shall consider only the more general model of total liabilities in the following derivations. However, the PDE method can be easily adapted for the GLWB rider hedging.

6.3.3. *Inner loop: Delta-hedging program.* We now discuss how to construct a hedging portfolio in order to hedge against the GLWB liability. To make a clear presentation, we consider the hedging program in the continuous-time model. Since the hedging is done in theory at every instant of time and the underlying equity process has constant volatility and a constant risk-free interest rate, it is sufficient to conduct delta hedging alone to eliminate equity risk. Recall that delta measures the sensitivity of the GLWB liability to changes in equity index values assuming all other variables remain the same. We have shown that the GLWB liability is given by $V_t = v(t, F_t, G_t, {}_tp_x)$, and hence the delta should be defined as[4]

$$\Delta_t = v_x(t, F_t, G_t, {}_tp_x). \tag{6.10}$$

In practice, the derivative is often approximated by the difference quotient

$$\Delta_t \approx \frac{v(t, F_t, G_t, {}_tp_x) - v(t, F_t - \Delta F, G_t, {}_tp_x)}{\Delta F}. \tag{6.11}$$

For example, an approximation of the delta can be obtained from the percentage change of the GLWB liability when the current account value F_t is shocked by 1%. Since we cannot trade policyholders' accounts, we can buy and sell equity index futures instead. In practice, hedging (rebalancing) is performed at discrete-time intervals generally ranging from daily to quarterly, in accordance with market circumstances and company philosophy.

Dynamics of mixed surplus and hedging portfolio

The idea is to keep track of deltas needed to create a hedging portfolio, and the cost of changes in delta is entirely financed by the accumulated surplus. In other words, the insurer's entire portfolio consists of two components at any point of time: the hedging instruments on the asset side and the accumulation of surplus from the liability side. We denote the value of the entire portfolio by H_t and the value of accumulated surplus/deficiency by R_t. For a better presentation of the results, we write $h_t = (F_t/S_t)v_x(t, F_t, G_t, {}_tp_x)$ for the number of units of equity index futures at time t. Then it is clear that

$$H_t = h_t S_t + R_t.$$

Although this formulation is not usually stated explicitly in practice, we will show next that it is indeed consistent with the recursive relation (6.1), which is used in practitioners' discrete-time models.

First, let us observe how the hedging instruments are financed with the surplus:

$$h_{t-\Delta t} S_t + R_{t-\Delta t}(1 + r\Delta t) + \mathsf{CF}(\Delta t) = h_t S_t + R_t, \tag{6.12}$$

where CF stands for cash flows (fee incomes, surrender charges and benefits):

$$\begin{aligned}\mathsf{CF}(\Delta t) = {}& {}_tp_x(mF_t + m_w G_t)\Delta t I(F_t > 0) + {}_tp_x\, {}_{\Delta t}q^{l}_{x+t} c_t F_t I(F_t > 0) \\ & - {}_tp_x h G_t \Delta t I(F_t \le 0).\end{aligned}$$

[4] In academic literature, Δ_t is often interpreted as the number of shares/units invested in the underlying asset. Therefore, it is common to see a definition such as

$$\Delta_t = \frac{F_t}{S_t} v_x(t, F_t, G_t, {}_tp_x).$$

However, as pointed out by a POG member, here we use a definition more commonly used by practitioners.

The left-hand side of (6.12) is the value of the surplus and hedging assets before rebalancing, and the right-hand side of (6.12) is the value of the portfolio immediately after rebalancing.

We shall use the notation $\Delta X_t = X_{t+\Delta t} - X_t$ for $X = R, h.$ Observe that (6.12) can be rewritten as

$$\Delta R_{t-\Delta t} = \mathsf{CF}(\Delta t) - \Delta h_{t-\Delta} S_t + rR_{t-\Delta t}\Delta t,$$

which is precisely the recursive relation (6.1) where $-\Delta h_{t-\Delta} S_t$ represents the additional cost of buying/selling extra deltas (or changes in asset values). In contrast with the recursive relation in (6.7), the change in surplus now includes the changes due to the hedging portfolio.

We have shown in Appendix (Section A.5) that the mixed portfolio provides a perfect hedge of the GLWB portfolio

$$H_t = V_t - e^{rt} V_0. \tag{6.13}$$

In other words, the value of the mixed portfolio at any point of time differs from the GLWB value only by a risk-free asset that is worth V_0 at time 0, regardless of the fluctuation of the GLWB liability due to changes in equity index.

6.4. **Computational techniques.** While all techniques described in this report can be adapted for the second case, we intend to focus on the implementation of a few representative techniques:

A. Closed-form solution/approximation
B. Crude Monte Carlo
E. Preprocessed inner loops
F. Least-Squares Monte Carlo
H. Numerical partial differential equation (PDE) method

6.4.1. *Numerical PDE method.* As shown in Appendix Section A.4, the function v can be determined by

$$v(t, x, y, z) = zyu(t, x/y), \tag{6.14}$$

where the function u satisfies a relatively simple PDE

$$\begin{aligned} \frac{1}{2} p^2 \sigma^2 s^2 u_{ss} + [(r - m - \rho)s - (m_w + h)]u_s + u_t - [\mu^d_{x+t} + \mu^l_t f(s) + r - \rho]u \\ -(m_w + ms) - \mu^l_t f(s) c_t s = 0, \end{aligned} \tag{6.15}$$

where u_{ss}, u_s, u_t are second derivative with respect to s and first derivatives with respect to s, t respectively. The PDE is subject to the boundary conditions

$$u(t, 1) = u_s(t, 1), \tag{6.16a}$$

$$u(t, 0) = h\overline{a}_{x+t}, \tag{6.16b}$$

$$\lim_{t\to\infty} u(t, s) = 0, \tag{6.16c}$$

where the annuity symbol refers to

$$\overline{a}_{x+t} := \int_0^\infty e^{-(r-\rho)s} \, {}_s p_{x+t} \mathrm{d}s.$$

The numerical algorithm for solving this PDE is given in Appendix B.

It follows from the reduction of dimensions in (6.14) that $v_x(t, x, y, z) = zu_s(t, x/y)$. Hence, we obtain an alternative approach to approximate the delta:

$$\Delta_t \approx {}_t p_x \frac{u(t, F_t/G_t + \Delta s) - u(t, F_t/G_t - \Delta s)}{2\Delta s}. \tag{6.17}$$

6.4.2. *Closed-form solution.* Many people dismiss this approach as impractical for complex products in general, such as variable annuity guaranteed benefits. Analytical solutions are in fact available in more cases than one might think. There have been many recent developments in academia on development analytical methods and approximations on risk-neutral valuations and risk measures of various guaranteed benefits. See, for example, PDE methods for GMAB, GMDB and GMWB in Feng and Volkmer [7], Milevsky and Salisbury [14], Ulm [16], etc. However, they do often rely on advanced computer techniques, which are not usually user friendly.

Here we provide an example of risk-neutral valuation of the GLWB liability and its delta calculation. The closed-form solutions presented here were based on simplifying assumptions of constant force of mortality and constant lapse rate. Techniques are available to allow for approximations under more realistic assumptions. In Section 6.5, we intend only to use closed-form solutions in this special case as benchmarks for testing Monte Carlo and numerical PDE methods.

It is clear from (6.9) and (6.10) that we need to find efficient methods to evaluate both v and v_x, which by reduction of dimensions in turn relies on the functions u and u_s. The derivation of analytical solutions to both u and u_s is provided in Appendix Section A.6.

6.4.3. *Least-Squares Monte Carlo.* We intend to estimate the functional relationship between the deltas of GLWB liability and four determining factors: time t, account value F_t, guarantee base G_t and survivorship ${}_tp_x$. Following the discussion in Section 5.6, we approximate the functional relationship by a linear combination of basis functions.

$$\bar{\Delta}_t = v_x(t, F_t, G_t, {}_tp_x) \approx \sum_{l=1}^{s} \beta_l \phi_l(t, F_t, G_t, {}_tp_x),$$

where ϕ_l's are basis functions: Ideally basis functions should incorporate features of the time-space surface v_x which determines the deltas. However, without any insight about the curve, we follow the standard practice as shown in Koursaris[11] to choose lower term polynomials.

The first and most naive approach of least-squares estimates is to regress the four variables $(t, F_t, G_t, {}_tp_x)$ against the response variable Δ_t. Keep in mind, however, that we do not know exact values of v_x, which should be estimated from inner-loop calculations according to the formula (6.11). The second approach, which requires understanding the structural property of the four-variable function, is to make use of the fact that v is determined by a function u of only two variables in (6.14). Therefore, deltas can be determined by the function u in the following way:

$$\bar{\Delta}_t = v_x(t, F_t, G_t, {}_tp_x) = {}_tp_x u_s(t, Y_t).$$

Then we can approximate the functional relationship

$$u_s(t, Y_t) \approx \sum_{l=1}^{s} \beta_l^* \phi_l^*(t, Y_t),$$

where ϕ_l^*'s are basis functions and β_l^*'s are to be determined by the least-squares method. Again exact values of u_s are not known in advance but can be estimated from the formula (6.17). Note that we have already shown the stochastic representation of $u(t, s)$ in Appendix Section A.7, which can be used for inner-loop simulations.

6.4.4. *Preprocessed inner loops.* Since there are four determining factors in this model, we have to construct a four-dimensional grid for preprocessed scenarios. For each point on the grid, we run an inner-loop calculation to determine risk-neutral valuation of GLWB liability and use difference quotient to approximate delta. The interpolation formula is the four-dimensional extension of the two-dimensional formula (5.10).

6.5. **Numerical comparison of Monte Carlo and PDE methods: inner loop.** We shall carry out the comparative analysis in two steps: (1) the accuracy of Monte Carlo simulations and other methods are tested on inner loops (delta calculation), which gives us an understanding of the compromise on accuracy from each method. (2) The efficiency (time consumption) of competing methods is then tested on a full scale with the Test Case II.

All computations are done on an iMac with 2.7 GHz Intel Core i5 and 8 GB 1600 MHz DDR3. Readers are reminded that all results on time consumption should be interpreted in relative terms, as insurance companies can implement them on much faster computing facilities and on much larger scales.

6.5.1. *Accuracy test.* We first consider the special case with constant force of mortality and lapse rate, under which closed-form solutions are developed in Appendix A.6. The valuation basis is given as follows:[5]

- Constant force of mortality $\lambda = \mu^d_{x+t} = 0.2$
- Constant lapse rate $\mu^l_t = 0$
- Risk-free rate of interest per annum $r = 0.0577$
- Free partial withdrawal rate per annum $h = 0.04$
- GLWB guarantee based rider charge per annum $m_w = 0.01$
- Account-value-based M&E fee rate per annum $m = (0.0036 + 0.0014) \times 4$
- Roll-up rate on guarantee base per annum $\rho = 0.05$

	Total liability	Relative error	Time (secs)
Monte Carlo (N=100,$1/n$=0.01)	-0.111643 (0.000628)	0.228%	15.380
PDE ($\Delta s = 0.01$)	-0.111380	0.008%	0.946
PDE ($\Delta s = 0.001$)	-0.111389	0.00008%	53.433
Analytical solution	-0.111389	-	0.075

TABLE 7. Risk-neutral valuation of the insurer's total liability ($\sigma = 0.05$)

For Monte Carlo simulations, we estimate the GLWB liability by its sample mean with a sample of size 20,

$$\hat{V}_t = \frac{1}{20}\sum_{i=1}^{20}\hat{V}_i,$$

where each V_i is a realization of the discrete time version of (6.8):

$$\hat{V}_i = \frac{1}{N}\sum_{j=1}^{N}\Bigg[\sum_{k=\lfloor n(t\vee\tau)\rfloor}^{nT} e^{-r[(k+1)/n-t]}\;{}_{k/n}p_x^{(j)}\,\frac{h}{n}G_{k/n}^{(j)}$$

$$-\sum_{k=\lfloor n(t\wedge\tau)\rfloor}^{\lfloor n(\tau\wedge T)\rfloor} e^{-r[(k+1)/n-t]}\;{}_{k/n}p_x^{(j)}\left(F_{k/n}^{(j)}\frac{m}{n}+G_{k/n}^{(j)}\frac{m_w}{n}+F_{(k+1)/n}^{(j)}\frac{\left({}_{1/n}q^l_{x+k/n}\right)^{(j)}}{1/n}\,\frac{c_{(k+1)/n}}{n}\right)\Bigg].$$

[5] The interest and fee rates stated here are all nominal convertible quarterly and per annum. When applying Monte Carlo methods, we always use their equivalent nominal rates with conversion period matched to time step. Similarly, when applying PDE methods, we always use their equivalent effective annual rates.

	Total liability	Relative error	Time (secs)
Monte Carlo (N=100,$1/n$=0.01)	-0.093583 (0.004248)	2.51%	17.481
Monte Carlo (N=1,000,$1/n$=0.01)	-0.092076 (0.001474)	0.86%	173.936
Monte Carlo (N=1000,$1/n$=0.001)	-0.091403 (0.001644)	0.12%	1782.660
PDE ($\Delta s = 0.01$)	-0.091271	0.02%	1.487
PDE ($\Delta s = 0.001$)	-0.091289	0.0002%	53.563
Analytical solution	-0.091290	-	0.075

TABLE 8. Risk-neutral valuation of total liability ($\sigma = 0.3$)

The superscript (j) indicates the j-th repetition of simulations. This formula is presented in a general form and will be used for numerical examples in the Test Case II. Note, however, in this simplified example, the lapsation is virtually ignored. The closed-form solution can be modified to include a positive constant lapse rate.

In Table 7, we test three methods for risk-neutral valuation, namely, Monte Carlo, numerical PDE and analytical solution, under a low-volatility assumption that $\sigma = 0.05$. For the Monte Carlo method, sample standard deviations are reported in brackets below their corresponding point estimates. It is clear from Table 7 that both Monte Carlo and PDE methods are quite accurate compared with analytical solutions. Keep in mind that the analytical solution and numerical PDE methods are much more complex than Monte Carlo. In such a case, it does not appear to be worthwhile the extra efforts to apply complex methods. If we move to a high-volatility assumption that $\sigma = 0.3$, then Table 8 shows that results of Monte Carlo simulations converge rather slowly. We can reduce the discrepancy between results from the PDE method and the Monte Carlo method by taking large sample size $N = 1,000$ and small time step $1/n = 0.001$. If one desires high accuracy, then it would take tremendous computation efforts to reduce sampling errors from Monte Carlo methods. This shows that it takes less computational efforts for the PDE method to achieve the convergence of results than it does for the Monte Carlo method. Note that these projections using very small time steps are shown for illustrative purposes. In practice, a more common time step is $1/n = 0.08$ (monthly) or greater. On the other hand, in practice it can take many more scenarios for results to converge and a much higher runtime overall. Readers should consider their own circumstances when assessing the level of run time improvement that they can expect from the PDE or other techniques.

In a second numerical example, we test the accuracy of results on deltas. Assume that $t = 0, F_0 = 0.8, G_0 = 1, {}_0p_x = 1$ and $F_0/S_0 = 1$. Here we present the calculation of $\Delta_0 = v_x(0, 0.8, 1, 1) = u_s(0, 0.8)$, which are estimated from (6.11) and (6.17) respectively.

In Table 9, we compare the accuracy of results from all three methods based on the approximation formula in (6.11). The sample size N and time step $1/n$ are reported for each Monte Carlo calculation. Since the volatility is relatively low, Monte Carlo simulation achieves rather high accuracy. Note, however, the true delta based on the analytical solution

	$v_x(0, 0.8, 1, 1)$	RE	Total liability	RE	Total liability (1% shock)	RE	Time (secs)
Monte Carlo (100,0.02)	-0.127337	0.22%	-0.088196 (0.000078)	0.32%	-0.089215 (0.000078)	0.32%	17.882
Monte Carlo (100,0.01)	-0.127079	0.01%	-0.087851 (0.000071)	0.07%	-0.088867 (0.000072)	0.07%	33.874
PDE ($\Delta s = 0.01$)	-0.126957	0.08%	-0.087902	0.01%	-0.088918	0.01%	0.513
PDE ($\Delta s = 0.001$)	-0.127064	0.00%	-0.087912	0.00%	-0.088928	0.00%	54.125
Analytical	-0.127064	-	-0.087912	-	-0.088928	-	0.583

TABLE 9. Approximation of $v_x(0, 0.8, 1, 1)$ based on formula (6.11) ($\sigma = 0.05$)

$\Delta s = 0.01$	$u_s(0, 0.8)$	RE	$u(0, 0.79)$	RE	$u(0, 0.81)$	RE	Time (secs)
Monte Carlo (100,0.01)	-0.127491	0.08%	-0.086477 (0.000831)	0.18%	-0.089026 (0.000835)	0.17%	41.545
PDE	-0.127619	0.02%	-0.086619	0.01%	-0.089172	0.01%	0.514
Analytical	-0.127588	-	-0.086629	-	-0.089181	-	0.364

$\Delta s = 0.001$	$u_s(0, 0.8)$	RE	$u(0, 0.799)$	RE	$u(0, 0.801)$	RE	Time (secs)
Monte Carlo (100,0.01)	-0.127781	0.15%	-0.088033 (0.000839)	0.28%	-0.088288 (0.000841)	0.28%	41.545
PDE	-0.127588	0.00%	-0.087784	0.00%	-0.088039	0.00%	53.881
Analytical	-0.127588	-	-0.087784	-	-0.088039	-	0.364

TABLE 10. Accuracy of $u_s(0, 0.8)$ based on formula (6.17) ($\sigma = 0.05$)

(A.10) is given in Table 10. If one desires very high accuracy, then the PDE method provides viable solutions as shown in Table 10. We can observe very similar results in Tables 11 and 12 when $\sigma = 0.3$.

We should also point out that delta estimates from the formula (6.17) in Tables 11 and 12 agree with exact values from the analytical formula for $u_s(0, 0.8)$ for at least five decimal places. The deviations of corresponding results in Tables 9 and 10 from the exact results is due to the approximation of the right-hand derivative in (6.11).

For the remainder of the report, we shall always use the approximation formula (6.17) to determine Δ_t with PDE methods ($\Delta s = 0.001$) and treat the results as the “best estimates” of true values for both GLWB liabilities and deltas.

6.5.2. *Efficiency test.* As we outlined earlier, there are two steps of computations to determine the conditional tail expectation risk measure of the greatest present value of the accumulated deficiency:

(1) Outer loops: Generate the cash flows including fee incomes, surrender charges, expenses, GLWB benefits and change in hedging instruments under each scenario of equity returns.
(2) Inner loops: Every step of rebalancing with the hedging instruments requires the computation of deltas. Deltas are approximated by difference quotients, which rely on risk-neutral valuation of the GLWB liability.

	$v_x(0,0.8,1,1)$	RE	Total liability	RE	Total liability (1% shock)	RE	Time (secs)
Monte Carlo (100,0.02)	-0.120425	2.96%	-0.071811 (0.004223)	1.85%	-0.072774 (0.004244)	1.86%	16.436
Monte Carlo (100,0.01)	-0.117323	0.31%	-0.068743 (0.003730)	2.50%	-0.069681 (0.003746)	2.46%	33.323
PDE $(\Delta s = 0.01)$	-0.116813	0.13%	-0.070490	0.023%	-0.071424	0.025%	1.052
PDE $(\Delta s = 0.001)$	-0.116961	0.00%	-0.070506	0.00%	-0.071442	0.00%	58.112
Analytical	-0.116961	-	-0.070506	-	-0.071442	-	0.431

TABLE 11. Approximation of $v_x(0,0.8,1,1)$ based on formula (6.11) $(\sigma = 0.3)$

$\Delta s = 0.01$	$u_s(0,0.8)$	RE	$u(0,0.79)$	RE	$u(0,0.81)$	RE	Time (secs)
Monte Carlo (100,0.01)	-0.119013	1.24%	-0.070797 (0.005168)	2.13%	-0.073177 (0.005227)	2.10%	40.868
PDE	-0.117563	0.0025 %	-0.069307	0.023%	-0.071658	0.023%	0.514
Analytical	-0.117558	-	-0.0693232	-	-0.071675	-	0.431

$\Delta s = 0.001$	$u_s(0,0.8)$	RE	$u(0,0.799)$	RE	$u(0,0.801)$	RE	Time
Monte Carlo (100,0.01)	-0.118381	0.70%	-0.072321 (0.005178)	2.75%	-0.072565 (0.005289)	2.75%	42.164
PDE	-0.117558	0.00%	-0.070389	0.00%	-0.070624	0.00%	53.882
Analytical	-0.117558	-	-0.070389	-	-0.070624	-	0.431

TABLE 12. Accuracy of $u_s(0,0.8)$ based on formula (6.17) $(\sigma = 0.3)$

In this numerical example, model parameters for the AG-43 CTE calculation and risk-neutral valuation of the GLWB liability are provided in Table 13. The majority of product features and model parameters in this example are taken from an actual example illustrated through a sample spreadsheet donated by an actuarial software vendor for this SOA research study. Keep in mind that instantaneous rates (interest, withdrawal, roll-up, fee etc.) are used in the continuous-time model for PDEs, while the Monte Carlo methods are based on the discrete-time model, which utilizes periodic rates (annual, quarterly, monthly, daily etc.). For consistency, we shall always use equivalent rates for different frequencies. For example, the roll-up rate is assumed to be 5% effective per annum. If we run projections on a monthly basis, $n = 12$, then ρ in discrete-time formulas such as (6.2) should be interpreted as the nominal rate compounded monthly $\rho = \rho^{(n)} = n[(1 + 5\%)^{1/n} - 1] = 4.889\%$. The corresponding instantaneous rate for the PDE method would be $\rho = \ln(1 + 5\%) = 4.879\%$. The acquisition cost is not actually used in this calculation, because the valuation date is four years after issue and the acquisition cost has already occurred prior to the start of projection. The survivorship is assumed to follow Table 14 with a fractional age assumption (constant force of mortality). We also include dynamic policyholder

Parameters	Assumptions (equivalent annual rate)
Age at issue	61
Age on valuation date	65
Initial account value on valuation date	5,850,000
GLWB ratchet	Yes
GLWB roll-up	5%
Annual free partial withdrawal percentage	4%
GLWB initial in-the-moneyness ratio as of valuation date (guarantee base/ account value)	100%
Management fees	0.56%
Kick back rate (percentage of management fees distributed to insurer)	40%
GLWB base charge	1%
Mortality and expense charge	1.40%
Number of policies at the start of projection	117.18
Acquisition cost (percentage of initial premium)	3%
Per policy unit maintenance overhead expense	21.25
Percentage of AV unit maintenance overhead expense	0.064%
Inflation rate applied to per policy expense	3%
Investment income earned rate on cash flows	5.76%
Investment income earned rate on accumulated surplus	5.76%
Discount rate on accumulated surplus or guaranteed benefit cash flows	3.76%

TABLE 13. Assumptions on GLWB product features

behavior in this model, where base lapse rates are stated in Table 15 and dynamic lapse rates are determined by

$$\text{dynamic lapse rate} = \text{base lapse rate} \times \text{dynamic factor}$$

and the dynamic factor function[6]

$$\text{dynamic factor} = (3-2x)x^2, \qquad x = \frac{\text{then-current account value}}{\text{then-current guarantee base}}.$$

[6] This function is suggested by one of the POG members, Mark Evans. Note, however, that none of the following analysis depends on this particular function. One may replace this function by some other function that reflects a company's own experience. It has also been suggested that the in-the-moneyness ratio be defined as $x = \max\left\{\frac{\text{then-current account value}}{\text{present value of future withdrawals}}, 1\right\}$. In such a case, the dynamic factor becomes a function of account value only, and the PDE method in this report can be adapted to address this type of dynamic lapse rates as well.

x	q_x	x	q_x	x	q_x	x	q_x	x	q_x
65	0.018191	76	0.050813	87	0.143012	98	0.344977	109	0.543602
66	0.020259	77	0.056327	88	0.156969	99	0.363757	110	0.547664
67	0.022398	78	0.062629	89	0.172199	100	0.382606	111	0.549540
68	0.024581	79	0.069595	90	0.188517	101	0.401942	112	0.550000
69	0.026869	80	0.077114	91	0.205742	102	0.422569	113	0.550000
70	0.029363	81	0.085075	92	0.223978	103	0.445282	114	0.550000
71	0.032169	82	0.093273	93	0.243533	104	0.469115	115	1
72	0.035268	83	0.101578	94	0.264171	105	0.491923		
73	0.038558	84	0.11025	95	0.285199	106	0.511560		
74	0.042106	85	0.119764	96	0.305931	107	0.526441		
75	0.046121	86	0.130583	97	0.325849	108	0.536732		

TABLE 14. GL 34 GMDB male mortality table

Policy year	Base lapse rates	Surrender charge rate
1	0.8%	7.0%
2	2.0%	6.0%
3	2.0%	5.0%
4	2.0%	4.0%
5	3.0%	3.0%
6	4.0%	2.0%
7	5.0%	1.0%
8	10.0%	0.0%
9	6.0%	0.0%
≥ 10	2.0%	0.0%

TABLE 15. Assumptions on lapsation

The most time-consuming component of the nested simulations is the computation of deltas. Let us first compare the delta calculation using Monte Carlo (MC) simulations and PDE methods. This is in essence a repetition of the work in Section 6.5.1. Nonetheless, this example is more general as the force of mortality is extracted from the life table, and dynamic lapse rates are considered. The focus of this section is to compare time consumptions.

Consider the risk-neutral valuation of GLWB liability in five years:

$$t = 5, F_t = 5,755,800, G_t = 7,674,000, S_t = 1,268, \ {}_tp_x = 0.72.$$

The equity index is always projected for $T = 50$ years at which ${}_Tp_x$ is zero for practical purposes. Monte Carlo simulations are done in the same way as in Section 6.5.1 with the inclusion of surrender charges and dynamic lapse factors. In Tables 16 and 18, we report the results on deltas for both Monte Carlo simulations with different numbers of repetitions, $N = 100, 1,000, 10,000$, and those from the PDE method with grid sizes $\Delta t = 0.01$ and $\Delta s = 0.001$. The first column represents the values of v_x in (6.10). Some practitioners may prefer using the approximation of delta by

$\Delta'_t \approx v(t, F_t, G_t,\ {}_tp_x) - v(t, F_t - \Delta F, G_t,\ {}_tp_x)$, which is the difference between the third and fourth columns. Hence we also present these results in the second column of Table 16. Note that in the second column the estimation errors are artificially inflated because of the large size of the liability value. Sample standard deviations are quoted in brackets under each estimate from a Monte Carlo simulation. The deltas obtained from the PDE approach are determined by (6.17).

There are two reasons why results from Monte Carlo simulations tend to differ from those from PDE methods:

(1) Continuous-time versus discrete-time models: The PDE method is based on the assumptions of continuous cash flows, whereas MC simulations are performed on a discrete-time basis.

(2) Sampling errors from MC methods: As a statistical procedure, MC methods inevitably introduce sampling errors that can be reduced only with very large sample sizes.

	$v_x(t, F_t, G_t,\ {}_tp_x)$	$(1) - (2)$	Total liability (1) $(F_t = 5,755,800)$	Total liability (2) (1% shock)	Time (secs)
Monte Carlo $(100, 0.01)$	-0.197038	$11,340$	$-57,163$ (79,768)	$-68,503$ (80,056)	14.536
Monte Carlo $(1,000, 0.01)$	-0.193887	$11,161$	$-42,407$ (21,165)	$-53,568$ (21,169)	140.519
MC (Antithetic) $(1,000, 0.001)$	-0.189232	$10,892$	$-29,361$ (13,834)	$-40,253$ (13,981)	$1,688.864$
MC (Antithetic) $(1,000, 0.0001)$	-0.188277	$10,721$	$-28,730$ (11,886)	$39,451$ (11,952)	$16,388.812$
PDE $(\Delta s = 0.001)$	-0.187765	$10,808$	-27,688	-38,496	306.882

TABLE 16. Delta calculation based on formula (6.11)($t = 5$ and $\sigma = 0.3$)

$\Delta_s = 0.001$	${}_tp_x u_s(t, F_t/G_t)$	$u(t, F_t/G_t - \Delta_s)$	$u(t, F_t/G_t + \Delta_s)$	Time (secs)
MC (Antithetic) (1,000,0.01)	-0.197212	-0.005704 (0.001822)	-0.006099 (0.001825)	195.80
MC (Antithetic) (1,000,0.001)	-0.191669	-0.003701 (0.001380)	-0.004085 (0.001388)	1,933.755
MC (Antithetic) (1,000,0.0001)	-0.189705	-0.003593 (0.001629)	-0.003972 (0.001631)	19,284.629
PDE $(\Delta s = 0.001)$	-0.189218	-0.003426	-0.003804	306.882

TABLE 17. Delta calculation based on formula (6.17) ($t = 5$ and $\sigma = 0.3$)

(Accuracy) In Tables 16 and 18, delta values at $t = 5$ and $t = 10$ are estimated from the approximation formula (6.11). In contrast, delta values at $t = 5$ and $t = 10$ are estimated from the approximation formula (6.17) in Tables 17 and 19.

Table 16 shows the case with a relatively high volatility $\sigma = 0.3$, under which we observe a gradual convergence of results from MC methods to those from the PDE method as the time step $1/n$ decreases. To reduce sampling errors, we implement the method of antithetic variates, which employs the sampling strategy that, for every sample path of cash flows based on randomly generated random variables $\{x_1, x_2, \cdots, x_{nT}\}$, we produce another path determined by opposite values $\{-x_1, -x_2, \cdots, -x_{nT}\}$. Note that in this case there are only 500 randomly generated sample paths even though it is stated that $N = 1,000$. The simulation results for deltas are relatively close to the value from the PDE. We should point out that standard deviations from Monte Carlo simulations with $N = 100$ and $N = 1,000$ are so high that the confidence intervals for the GLWB liability value would be rather wide. For example, the approximate 95% confidence interval for the GLWB liability using the MC with $N = 1,000$ and $1/n = 0.0001$ would be $[-52,026.56, -5,433.44]$ based on asymptotic normality. Delta estimates appear to be close to that from the PDE method.

	$v_x(t, F_t, G_t,\ {}_tp_x)$	$(1) - (2)$	Total liability (1) $(F_t = 5,755,800)$	Total liability (2) (1% shock)	Time (secs)
MC $(100, 0.01)$	-0.322120	$18,540$	$-442,592$ (38,443)	$-461,132$ (38,458)	15.629
MC $(1,000, 0.01)$	-0.323383	$18,613$	$-453,226$ (11,534)	$-471,839$ (11,522)	154.885
MC $(10,000, 0.01)$	-0.322815	$18,581$	$-452,679$ (2,198)	$-471,260$ (2,201)	$1,610.244$
PDE $(\Delta s = 0.001)$	-0.322656	$18,572$	-451,532	-470104	308.566

TABLE 18. Delta calculation based on formula (6.11) ($t = 5$ and $\sigma = 0.1$)

$\Delta s = 0.001$	${}_tp_x u_s(t, F_t/G_t)$	$u(t, F_t/G_t - \Delta s)$	$u(t, F_t/G_t + \Delta s)$	Time (secs)
Monte Carlo (1,000,0.01)	-0.324662	-0.058078 (0.001098)	-0.058727 (0.001097)	179.982
PDE	-0.324676	-0.058514	-0.059164	308.566

TABLE 19. Delta calculation based on formula (6.17) ($t = 5$ and $\sigma = 0.1$)

In the case of $\sigma = 0.1$, pilot experiments show that discrete-time versus continuous-time models do not cause a very significant difference. Hence we focus on the demonstration of the fact that sampling errors of Monte Carlo simulations tend to decrease as sample sizes increase in Tables 18 and 19. As with the previous case, we also observe that computations with the formula (6.17) achieves a higher level of accuracy than those with the formula (6.11).

To illustrate the consistency of results, we run another example of delta calculation at a later date:

$$t = 10, F_t = 5,755,800, G_t = 7,674,000,\ {}_tp_x = 0.72.$$

Tables 20 and 22 present the deltas determined by the formula (6.11) under assumptions of high volatility and low volatility, whereas Tables (21) and (23) show the corresponding results from the formula (6.17). It turns out that risk neutral valuations of total liabilities improve for $t = 10$ as opposed to corresponding results for $t = 5$.

	$v_x(t, F_t, G_t, {}_tp_x)$	$(1)-(2)$	Total liability (1)	Total liability (2)	Time
			$(F_t = 5,755,800)$	(1% shock)	(secs)
Monte Carlo	-0.171132	10,847	-337,945	-348,792	14.436
(100,0.01)			(58,346)	(58,510)	
Monte Carlo	-0.171456	9,868	-329,079	-338,947	145.323
(1,000,0.01)			(21,404)	(21,519)	
Monte Carlo	-0.167909	9665	-317,798	-327,463	1,406.892
(1,000,0.001)			(23,382)	(23,471)	
PDE	-0.167224	9626	-317,048	-326,674	308.279
(Δs=0.001)					

TABLE 20. Delta calculation based on formula (6.11) ($t = 10$ and $\sigma = 0.3$)

$\Delta s = 0.001$	${}_tp_x u_s(t, F_t/G_t)$	$u(t, F_t/G_t - \Delta s)$	$u(t, F_t/G_t + \Delta s)$	Time (secs)
Monte Carlo	-0.171769	-0.042030	-0.042373	160.307
(1,000,0.01)		(0.002641)	(0.002641)	
Monte Carlo	-0.169929	-0.041055	-0.041395	1,590.024
(1,000,0.001)		(0.002659)	(0.002661)	
PDE	-0.168798	-0.041146	-0.041483	308.279

TABLE 21. Delta calculation based on formula (6.17) ($t = 10$ and $\sigma = 0.3$)

	$v_x(t, F_t, G_t, {}_tp_x)$	$(1)-(2)$	Total liability (1)	Total liability (2)	Time
			$(F_t = 5,755,800)$	(1% shock)	(secs)
Monte Carlo	-0.221057	12,724	-587,171	-599,895	13.984
(100,0.01)			(19,018)	(19,002)	
Monte Carlo	-0.221802	12,766	-599,713	-612,479	137.291
(1,000,0.01)			(6,084)	(6,059)	
Monte Carlo	-0.221619	12,757	-599,992	-612,749	1,384.635
(10,000,0.01)			(2,860)	(2,853)	
PDE	-0.221343	12,740	-599,297	-612,037	308.566
(Δs=0.001)					

TABLE 22. Delta calculation at $t = 10$ and $\sigma = 0.1$ based on formula (6.11)

(Efficiency) Based on the results from Tables 16 and 18, it is attempting to say that Monte Carlo estimates with $N = 100$ are modestly accurate but highly efficient, and there does not appear to any advantage for the PDE method. However, one should keep in mind that the delta calculation is required for every point of time for rebalancing. Here is a rough estimate of how much time is needed to calculate the CTE risk measures with the Monte Carlo simulations with $N = 1,000$ scenarios, ignoring additional time consumption for generating outer-loop scenarios.

(Monte Carlo $N = 100$) 14 (secs) × 50 (times of rebalancing each year) × 50 (years) × 1000 (outer loops) = 7×10^7 (secs) ≈ 3.82 (months).

$\Delta s = 0.001$	${}_tp_x u_s(t, F_t/G_t)$	$u(t, F_t/G_t - \Delta s)$	$u(t, F_t/G_t + \Delta s)$	Time (secs)
Monte Carlo (1,000,0.01)	-0.223095	-0.077975 (0.000817)	-0.078422 (0.000817)	160.333
PDE	-0.222938	-0.077871	-0.078317	308.566

TABLE 23. Delta calculation based on formula (6.17) ($t = 10$ and $\sigma = 0.1$)

(Monte Carlo $N = 1000$) 140 (secs) $\times$ 50 (times of rebalancing each year) $\times$ 50 (years) $\times$ 1000 (outer loops) = 7×10^8 (secs) ≈ 11.25 (years).

In contrast, the PDE method does not require inner-loop simulations. The CTE risk measure calculation with the PDE method **requires only about five minutes** plus additional time for generating outer-loop scenarios. Although estimates of run times appear high for this personal computer, this is not to suggest that it is impractical to use crude Monte Carlo. One could perhaps use faster computers with parallel computing to solve this problem. These estimates are merely used here to make a comparison between relative time consumptions of the two methods.

The drastic reduction of run time for the PDE method is owing to the fact that the algorithm marches through all time-space grid points, moving backwards from the terminal time $T = 50$ to time 0. A by-product of such an algorithm is that the risk-neutral values at all grid points are produced all at once, which can be viewed as a table of risk-neutral values for all combinations of $(t, F_t, G_t, {}_tp_x)$. The deltas can be easily estimated from risk-neutral values at neighboring grid points for all periods. In essence, this approach is very similar to method E (preprocessed inner loops).

We employ a biweekly dynamic hedging portfolio in order to better illustrate the visual effect of outcomes from hedging, i.e., $n = 100$. To demonstrate the effect of hedging, we use the same rate of return $r = 0.0576$ per annum and assume that the expected rate of return for the equity index $\mu = r = 0.0577$ per annum in the outer-loop calculations.

Due to the tremendous improvement on time consumption, we carry out the rest of the calculations with the PDE method. Here we compare the dynamics of the surplus with and without the delta hedging program. In Figure 24, we present the evolution of present values of the accumulated surplus/deficiency under 12 different scenarios. The red line represents the surplus before implementing the hedging program, whereas the blue line corresponds to the surplus after applying dynamic hedging on a biweekly basis. Note that the hedging instruments are not included in the values for the blue line. It is clear from the figure that the surplus with hedging always approaches $-V_0 = -2.9138 \times 10^4$. It is predicted by (6.13) that $e^{-rt}(H_t - V_t) = -V_0$. When $t \to T$, it is easy to see that $V_t \to 0$ as ${}_Tp_x \to 0$. Therefore, the present value of accumulated surplus/deficiency approaches $-V_0$. The fact that we set the parameters in such a way that $V_0 = 2.9138 \times 10^4$ indicates that the risk-neutral value of GLWB benefits exceeds that of fee incomes and surrender charges.

We compare the distributions of present values of terminal surplus with and without the hedging program in Figure 25. The histogram on the left-hand side represents the surplus at the end of $T = 50$ years without a hedging program, whereas the histogram on the right-hand side shows the distribution of the terminal surplus with the effect of dynamic hedging. In both cases, the height of each bar in the histograms shows the frequency of occurrences in each category corresponding to the width of each bar. It is clear that the surplus is widely spread over both the negative and positive sides without a hedging problem. The implementation of the hedging program is so effective that the terminal surplus is nearly concentrated around the predicted cost of the GLWB value at the inception.

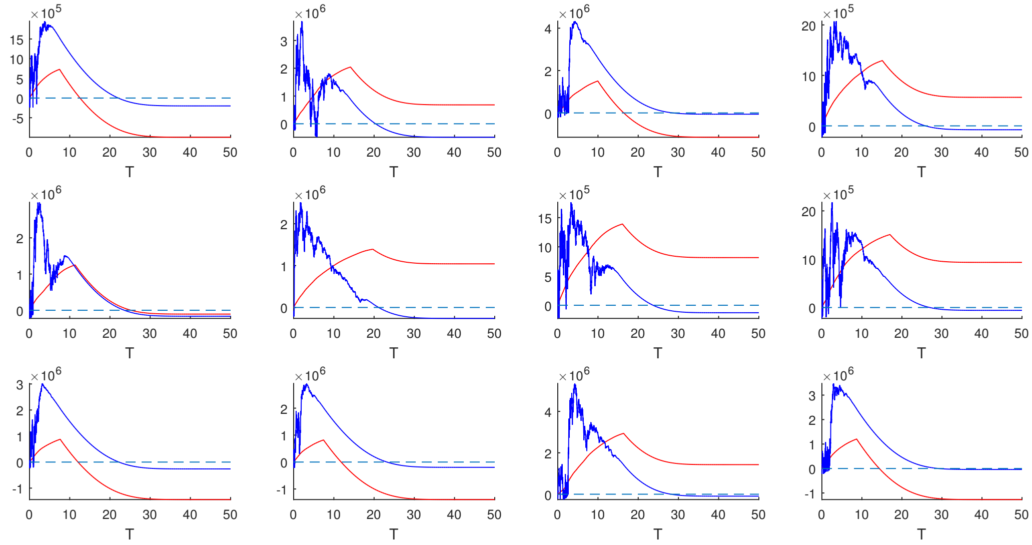

FIGURE 24. Sample paths of the insurer's surplus over time with and without a hedging program

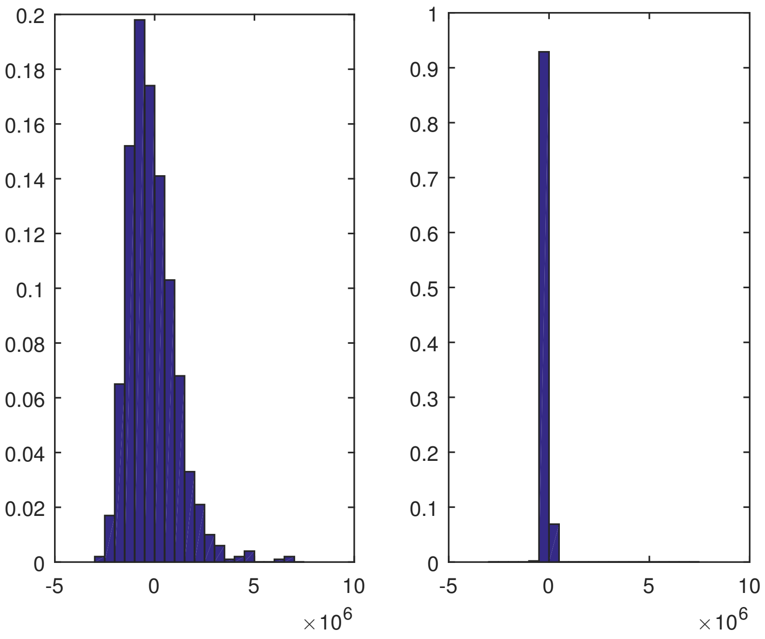

FIGURE 25. Histograms of the insurer's terminal surplus with and without a hedging program

6.5.3. *Additional example of GLWB rider hedging.* To show that the PDE method has similar efficiency in other cases as in the above example of total product liability hedging, here we present an additional example of delta calculations for the GLWB rider hedging. The difference between two types of liabilities is discussed in Section 6.3.2. In this case, we consider a common design in practice for the GLWB rider where roll-up stops when withdrawals start. Deltas are evaluated using two Monte Carlo methods. The first one is based on the four dimensional formulation similar to (6.11),[7]

$$\Delta_t \approx \frac{v(t, F_t + \Delta F, G_t, \ {}_tp_x) - v(t, F_t - \Delta F, G_t, \ {}_tp_x)}{2\Delta F}, \tag{6.18}$$

and the second one is based on the two dimensional formulation similar to (6.17),

$$\Delta_t \approx {}_tp_x \frac{u(t, s + \Delta s) - u(t, s - \Delta s)}{2\Delta s}. \tag{6.19}$$

For better comparison, we assume that withdrawals start at time 0 and the account value and guarantee base all reach the same levels as in the previous example, for instance,

$$t = 5, F_t = 5,755,800, G_t = 7,674,000, S_t = 1,268, \ {}_tp_x = 0.72.$$

We consider the risk-free interest rate to be $r = 0.02$ and the rate of withdrawals per annum per dollar of the policyholder's account value to be $h = 0.04$. The equity model remains the same as the previous example with the volatility parameter $\sigma = 0.1$. Tables 24 and 25 show the risk-neutral values of GLWB rider liability and estimates of delta values. For each of the Monte Carlo methods, the pair in brackets show the number of repetitions and the size of time step. The numbers in brackets underneath rider liabilities and delta values show corresponding sample standard deviations. Observe that the convergence of results for Monte Carlo methods and those from the PDE method is consistent with that in Tables 18 and 19.

Similarly, we also test the methods at a later time point:

$$t = 10, F_t = 5,755,800, G_t = 7,674,000, S_t = 1,268, \ {}_tp_x = 0.72.$$

Under the same assumption of volatility parameter $\sigma = 0.1$, we show the results for rider liabilities and delta values in Tables 26 and 27. We observe a similar pattern as in Tables 22 and 23. As pointed out in the previous section, the PDE method does not necessarily save time if the comparison with Monte Carlo methods is done only for each estimate of deltas. Note, however, that outer-loops invoke inner-loop delta evaluations at multiple time nodes and multiple levels of risk factors. The PDE algorithm produces all estimates of risk-neutral values at every point on the time-space grid, whereas Monte Carlo methods would require repetitive calculations at every single point in time and risk factor.

	Δ_t	Δ'_t	Rider liability (-1% shock)	Rider liability (1% shock)	Time (secs)
MC (100,0.01)	-0.169603 (0.003909)	-976,200	-147,850 (5,232)	-167,374 (5,459)	13.458
MC (1000,0.01)	-0.169102 (0.001174)	-973,317	-147,181 (1,552))	-166,947 (1,642)	131.092
MC (1,000,0.001)	-0.169015 (0.001115)	-972,815	-148,304 (1,491)	-167,760 (1,544)	1323.448
PDE (ds=0.001)	-0.168755	-971,318	-148,210	-167,636	283.949

TABLE 24. Delta calculation based on formula (6.18) ($t = 5$ and $\sigma = 0.1$)

[7] Some practitioners may prefer using the approximation of delta by $\Delta'_t \approx v(t, F_t + \Delta F, G_t, \ {}_tp_x) - v(t, F_t - \Delta F, G_t, \ {}_tp_x)$. Hence we also present these results in Tables 24 through 27.

$\Delta s = 0.001$	Δ_t	Δ_t'	$u(t, s - \Delta s)$	$u(t, s + \Delta s)$	Time (secs)
Monte Carlo (1,000,0.01)	-0.168650 (0.001235)	-970,714	-0.028285 (0.000256)	-0.028753 (0.000254)	128.756
PDE	-0.168742	-971,244	-0.028369	-0.028838	283.949

TABLE 25. Delta calculation based on formula (6.19) ($t = 5$ and $\sigma = 0.1$)

6.6. **Numerical comparison of all techniques: outer loop.**

6.6.1. *Least-Squares Monte Carlo.* In the first approach, we set up an equidistant time-space grid (t_i, x_j, y_k, z_l) as follows:

$$\begin{aligned}
t_i &= (i-1)\Delta t, \qquad \Delta t = 10,\ i = 1, \cdots, 6,\\
x_j &= 1,462,500 + (j-1)\Delta x, \qquad \Delta x = 1,462,500,\ j = 1, \cdots, 8,\\
y_j &= \max(x_j, 5,850,000) + (j-1)\Delta y, \qquad \Delta y = 1,462,500,\\
z_l &= (l-1)\Delta z, \qquad \Delta z = 0.2,\ l = 1, \cdots, 6.
\end{aligned}$$

We select the following basis functions:

$$1, t, t^2, x, x^2, y, y^2, z, z^2, tx, ty, tz, xy, xz, yz.$$

Note that these lower terms polynomials are selected for their simplicity, in a similar manner to the example in Koursaris [11].

Applying a least-squares method and removing terms with negligible coefficients, we obtain the following function:

	Δ_t	Δ_t'	Rider liability (-1% shock)	Rider liability (1% shock)	Time (secs)
MC (100,0.01)	-0.095237 (0.002424)	-548,167	-260,899 (5,416)	-271,862 (5,403)	12.915
MC (1,000,0.01)	-0.094250 (0.000899)	-542,482	-260,101 (1,405)	-270,954 (1,446)	124.307
MC (1,000,0.001)	-0.094196 (0.000698)	-542,175	-259,349 (1,618)	-270,193 (1,637)	1,230.840
PDE (ds=0.001)	-0.094071	-541,457	-259,719	-270,548	283.949

TABLE 26. Delta calculation based on formula (6.18) ($t = 10$ and $\sigma = 0.1$)

$\Delta s = 0.001$	Δ_t	Δ_t'	$u(t, s - \Delta s)$	$u(t, s + \Delta s)$	Time (secs)
Monte Carlo (1,000,0.01)	-0.094303 (0.000942)	-542,791	-0.047862 (0.000235)	-0.048124 (0.000236)	114.317
PDE	-0.094055	-541,457	-0.047870	-0.048131	283.949

TABLE 27. Delta calculation based on formula (6.19) ($t = 10$ and $\sigma = 0.1$)

$v_x(t,x,y,z) = -2.28293723535360 \times 10^{-5} t^2 + 5.76548315616552 \times 10^{-8} x - 3.25860281780238 \times 10^{-15} x^2 - 6.53438204339024 \times 10^{-8} y + 2.08191833331689 \times 10^{-15} y^2 - 1.35618136701221 \times 10^{-9} tx + 1.64788946873263 \times 10^{-9} ty + 1.20219406643236 \times 10^{-15} xy + 5.49811600487470 \times 10^{-8} xz - 6.10036014233364 \times 10^{-8} yz.$

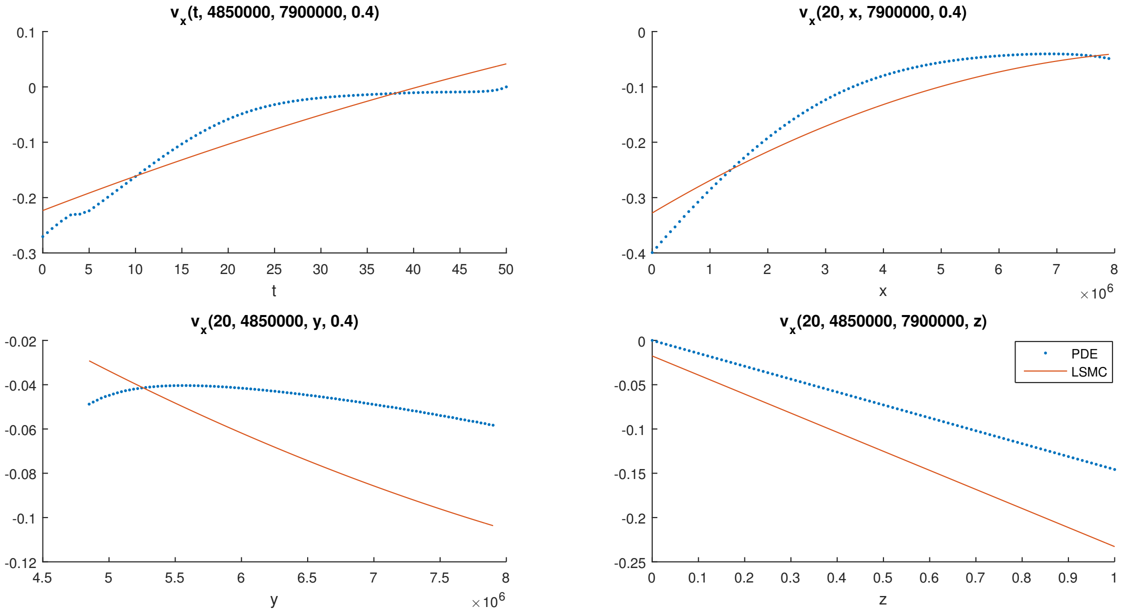

FIGURE 26. Curve fitting in LSMC

We demonstrated in Section 6.5.1 that the PDE method produces relatively accurate results, which are used as benchmarks in this example. Figure 26 shows that the polynomial approximation does better in t and F_t dimensions than it does in G_t and ${}_tp_x$ dimensions. Although the polynomial does capture the general pattern of the time-space surface, the results are generally not very accurate. It is shown in Table 28 that many results are quite far from those found using the PDE method. Note that errors from LSMC can be affected by errors from inner-loop calculations. Comparing the results from Monte Carlos simulations and the LSMC on a few off-grid points, we observe that errors from inner-loop simulations are not actually big contributors to the errors from the LSMC method.

In the second approach, we shall demonstrate the effect of LSMC in two dimensions. Note that here we take advantage of the reduction of dimensions shown in (6.14). We have to admit that this would be an unusual approach through which to apply the LSMC method. Nevertheless, we can use this approach to demonstrate significant improvement due to employing the analytical structure of the underlying stochastic model as opposed to brute force Monte Carlo simulations. In this case, we choose the following basis functions to approximate $u_s(t,s)$:

$$1, t, t^2, t^3, s, s^2, s^3, ts, ts^2, t^2s$$

Applying the method of least squares and ignoring terms with negligible coefficients, we obtain the following approximating function:

$u_s(t,s) = -1.260962 + 0.012849t + 0.000054t^2 - 0.0000016t^3 + 2.334049s - 3.711706s^2 + 2.393581s^3 + 0.048018ts - 0.0003019t^2s - 0.043594ts^2.$

(t,x,y,z)	PDE	Monte Carlo	RE	LSMC	RE
(6.68, 6103304, 6203304, 0.8720)	-0.175280	-0.178346	1.08%	-0.074610	-57%
(9.34,5304180,6304180,0.6797)	-0.188576	-0.178346	0.62%	-0.110181	41.57%
(15.34, 6974643, 6974643, 0.3492)	-0.051869	-0.051839	0.11%	-0.041223	-20.53%
(17.55, 598529, 6974643, 0.2968)	-0.248494	-0.248275	0.41%	-0.252185	1.50%
(19.07,5201927,6304180,0.6550)	-0.073334	-0.051839	0.058%	-0.087315	-19.06%
(21.93, 2745589, 6816817,0.2760)	-0.062374	-0.062183	0.31%	-0.111818	79.27%
(26.39, 3175785, 7102702, 0.1359)	-0.014242	-0.014192	0.55%	-0.037565	163.73%
(28.27, 4588345, 6384283, 0.1626)	-0.008606	-0.014192	0.23%	-0.020196	134.69%
(38.62, 2781940, 7102702, 0.0601)	-0.001991	-0.001991	0.19%	0.055272	-2878.18%
(39.76,1198307, 7792180, 0.1189)	-0.016106	-0.016106	0.28%	0.053988	-435.20%

TABLE 28. LSMC with four variables and relative errors

We use an equidistant grid of (t_i, s_j) given by

$$t_i = (i-1)\Delta t, \qquad \Delta t = 1, i = 1, \cdots, 51,$$

$$s_j = (j-1)\Delta s, \qquad \Delta s = 0.1, j = 1, \cdots, 11.$$

(t,s)	PDE	Monte Carlo	RE	LSMC	RE
(0,1)	-0.128920	-0.159537	23.75%	-0.245038	90.06%
(1,0.9)	-0.358906	-0.355374	0.98%	-0.401345	11.82%
(2,0.8)	-0.436890	-0.431742	1.18%	-0.497740	13.93%
(3,0.7)	-0.498723	-0.495406	0.67%	-0.551028	10.49%
(4,0.6)	-0.587266	-0.581283	1.02%	-0.578014	1.58%
(5,0.5)	-0.659056	-0.651280	1.18%	-0.595504	9.64%
(6,0.4)	-0.720308	-0.715813	0.62%	-0.620304	13.89%
(7,0.3)	-0.783270	-0.777856	0.69%	-0.669220	14.56%
(8,0.2)	-0.848309	-0.843980	0.51%	-0.759058	10.52%
(9,0.1)	-0.918761	-0.907156	1.26%	-0.906622	1.32%
(10,0)	-0.999415	-1.044595	4.52%	-1.128720	12.94%

TABLE 29. Accuracy of $u_s(t, s)$ on the grid

A clear advantage of applying the LSMC to the two-dimensional functional relationship is that both t and s have bounded domains. Comparing Tables 28 and 29, we observe significant improvement on the accuracy of results. However, as we shall see later, these results are still not accurate enough to produce a highly effective hedging program. Table 29 shows the comparison of accuracy on grid points (t_i, s_j), whereas Table 30 reports the accuracy of results on points off the grid.

6.6.2. *Preprocessed inner loops.* We use the same grid as in the previous section and estimates of V_t from Monte Carlo simulations. Then the risk-neutral values are used to find deltas according to (6.11) and (6.17). As with the LSMC method, we can also implement the method of preprocessed inner loops with both the four-variable functional relationship v_x and the two-variable functional relationship u_s.

In Table 31, we apply the method to approximate the actual function $u_x(t,x,y,z)$. Since there is no interpolation for grid points, we shall test accuracy on only a few off-the-grid points. The interpolation is based on the four dimensional analogue of (5.10), and we use the built-in function interpn in Matlab to implement the procedure.

Similarly, we apply the method to approximate the function $u_s(t,s)$ in Table 32. Compared with Table 31, we observe that the interpolation turns to be much more accurate for the functional relationship with two variables than that with four variables. It is in fact a common phenomenon in numerical analysis, known as the "curse of dimensionality." As the number of dimensions increases, the volume of the space increases so much that sampling points become scarce. To

(t,s)	PDE	Monte Carlo	RE	LSMC	RE
(4.88,0.971)	-0.233102	-0.239505	2.75%	-0.219808	5.70%
(6.35,0.959)	-0.241904	-0.243275	0.56%	-0.214860	11.17%
(13.93,0.957)	-0.160784	-0.159947	0.52%	-0.115662	28.06%
(17.34,0.916)	-0.126440	-0.125833	0.48%	-0.122065	3.46%
(21.62,0.800)	-0.089758	-0.089423	0.42%	-0.142661	58.94%
(24.74,0.792)	-0.068372	-0.068551	0.26%	-0.107248	56.86%
(35.29,0.485)	-0.037405	-0.037354	0.13%	-0.002076	94.45%
(36.67,0.422)	-0.036653	-0.036583	0.19%	-0.006615	81.95%
(41.88,0.158)	-0.110142	-0.104925	4.73%	-0.274434	149.16%
(48.24,0.142)	-0.018514	-0.018683	0.90%	-0.247936	1239.16 %

TABLE 30. Accuracy of $u_s(t,s)$ off the grid

(t,x,y,z)	PDE	Monte Carlo	RE	Preprocessed	RE
(6.68, 6103304, 6203304, 0.8720)	-0.175280	-0.178346	1.08%	-0.128398	26.75%
(9.34,5304180,6304180,0.6797)	-0.188576	-0.178346	0.62%	-0.082481	56.26%
(15.34, 6974643, 6974643, 0.3492)	-0.051869	-0.051839	0.11%	-0.057946	11.71%
(17.55, 598529, 6974643, 0.2968)	-0.248494	-0.248275	0.41%	-0.003968	98.40%
(19.07,5201927,6304180,0.6550)	-0.073334	-0.051839	0.058%	-0.060960	16.87%
(21.93, 2745589, 6816817,0.2760)	-0.062374	-0.062183	0.31%	-0.013473	78.40%
(26.39, 3175785, 7102702, 0.1359)	-0.014242	-0.014192	0.55%	-0.006281	55.90%
(28.27, 4588345, 6384283, 0.1626)	-0.008606	-0.014192	0.23%	-0.009668	12.34%
(38.62, 2781940, 7102702, 0.0601)	-0.001991	-0.001991	0.19%	-0.001172	41.09%
(39.76,1198307, 7792180, 0.1189)	-0.016106	-0.016106	0.28%	-0.000928	94.24%

TABLE 31. Preprocessed with four variables and relative errors

achieve the same level of accuracy in four dimensions as that in two dimensions, one has to enormously increase the number of grid points.

(t,s)	PDE	Monte Carlo	RE	Preprocessed	RE
(4.88,0.971)	-0.233102	-0.239505	2.75%	-0.227417	2.44%
(6.35,0.959)	-0.241904	-0.243275	0.56%	-0.233378	3.52%
(13.93,0.957)	-0.160784	-0.159947	0.52%	-0.160633	0.09%
(17.34,0.916)	-0.126440	-0.125833	0.48%	-0.127042	0.47%
(21.62,0.800)	-0.089758	-0.89423	0.42%	-0.0895989	0.18%
(24.74,0.792)	-0.068372	-0.068551	0.26%	-0.068434	0.09%
(35.29,0.485)	-0.037405	-0.037354	0.13%	-0.037711	0.82%
(36.67,0.422)	-0.036653	-0.036583	0.19%	-0.037003	0.95%
(41.88,0.158)	-0.110142	-0.104925	4.73%	-0.131462	19.35%
(48.24,0.142)	-0.018514	-0.018683	0.90%	-0.046432	150.79 %

TABLE 32. Accuracy of $u_s(t,s)$ off the grid

(t,s)	PDE	Monte Carlo	RE
(48,0.1)	-0.128013	-0.083933	34.43%
(48,0.2)	-0.019462	-0.019669	1.06%
(49,0.1)	-0.013582	-0.013738	1.15%
(49,0.2)	-0.013533	-0.013692	1.17%

TABLE 33. Errors from neighboring points of (48.24,0.142)

There is an “outlier” in Table 32 for the approximate delta value at $(48.24, 0.142)$, which appears to have an error of 150.79%. We look up the neighboring grid points and the corresponding delta values from inner loops in Table 33. At first glance, it might be surprising that off-grid points by linear interpolation would be much less accurate than the neighboring four grid points. Owing to the efficiency of the PDE approach, we can actually present a plot of the surface in Figure 27. A careful examination of the delta values would reveal that the surface of $u_s(t,s)$ has more curvature in this particular neighborhood than other places. If points for interpolation are drawn from regions with deep curvature, then the results would be quite inaccurate unless one uses a very dense grid.

We are now ready to present the end results of the AG-43 stochastic scenario amount calculation. Recall that cash flows in outer loops are calculated from the GLWB model with parameters listed in Tables 13–14. The model incorporates dynamic policyholder behavior and all expenses. In Table 34, we present CTE stochastic scenario amounts under various volatility assumptions. The formulas for computing deltas in each section of the table are identified in the first column. It is clear that the method of preprocessed inner loops produces results close to those from the PDE methods in the case where $\sigma = 0.3$, and the inner loops are replaced by approximations based on two variable functional relationship v. In contrast, the results for preprocessed inner loops are not accurate at all when $\sigma = 0.3$, and inner loops are replaced by approximations based on the four variable functional relationship u. In general, all methods produce reasonably close results under the low-volatility assumption $\sigma = 0.1$.

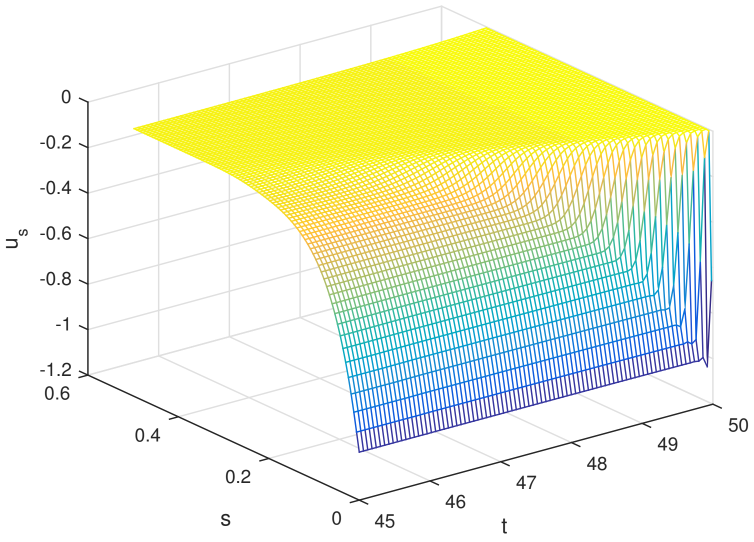

FIGURE 27. Surface of $u_s(t,s)$

$\sigma = 0.3$	excluding CDHS	LSMC	PDE	Preprocessed
70% CTE (6.17)	-3,725,724 (1,135,173)	-2,204,752 (741,975)	-1,471,402 (495,620)	-1,319,600 (432,776)
time (secs)	67.89	2692.27	544.135	4632.42
$\sigma = 0.3$	excluding CDHS	LSMC	PDE	Preprocessed
70% CTE (6.11)	-3,666,006 (1,107,968)	-11,685,363 (41,661,646)	-1,551,701 (484,667)	-3,818,887,705 (6,188,486,393)
time (secs)	68.482	2973.548	473.845	3938.779
$\sigma = 0.1$	excluding CDHS	LSMC	PDE	Preprocessed
70% CTE (6.11)	-40,255 (169,037)	461,599 (31,096)	665,981 (67,409)	581,240 (187,028)
time (secs)	64.854	2,871.289	489.255	3,921.446

TABLE 34. Comparison of CTE (best efforts) with various techniques

Another intuitive way for us to check the accuracy of delta calculations is through looking at histograms of terminal surpluses with hedging programs. If deltas are calculated correctly, then the mixed portfolio of hedging and surplus should become more or less a risk-free asset according to the no-arbitrage theory. The effectiveness of a hedging program in this model largely relies on the accuracy of delta calculation. Figure 25 has confirmed such a prediction in the earlier example without the consideration of expenses. In Figure 28, we can tell that the hedging program developed from the PDE method is highly effective as terminal balances of the mixed portfolio are concentrated around values close to zero. The LSMC method fails to produce any visible effect of hedging against financial risks, as the variation of portfolio values is as big as the surplus prior to hedging. This lack of hedging effect is likely due to the fact that deltas are far from accurate, as shown in Table 28. The method of processed inner loops produces even worse results, as terminal portfolio values vary so much that the scale of the plot can no longer match those of plots for LSMC and PDE methods. Figure 29 shows the effect of hedging programs under the low-volatility assumption $\sigma = 0.1$. Although the preprocessed inner loops do not appear to have produced accurate deltas, the distribution of the terminal surplus is on the same scale as the rest of distributions. In Figure 30, we implement similar procedures except that the LSMC and the method of processed inner loops are applied to approximate the two-variable functional relationship, which significantly improves their accuracy. The comparison between Figure 28 and 30 clearly shows that the accuracy of delta calculation is critical to the success of a hedging program.

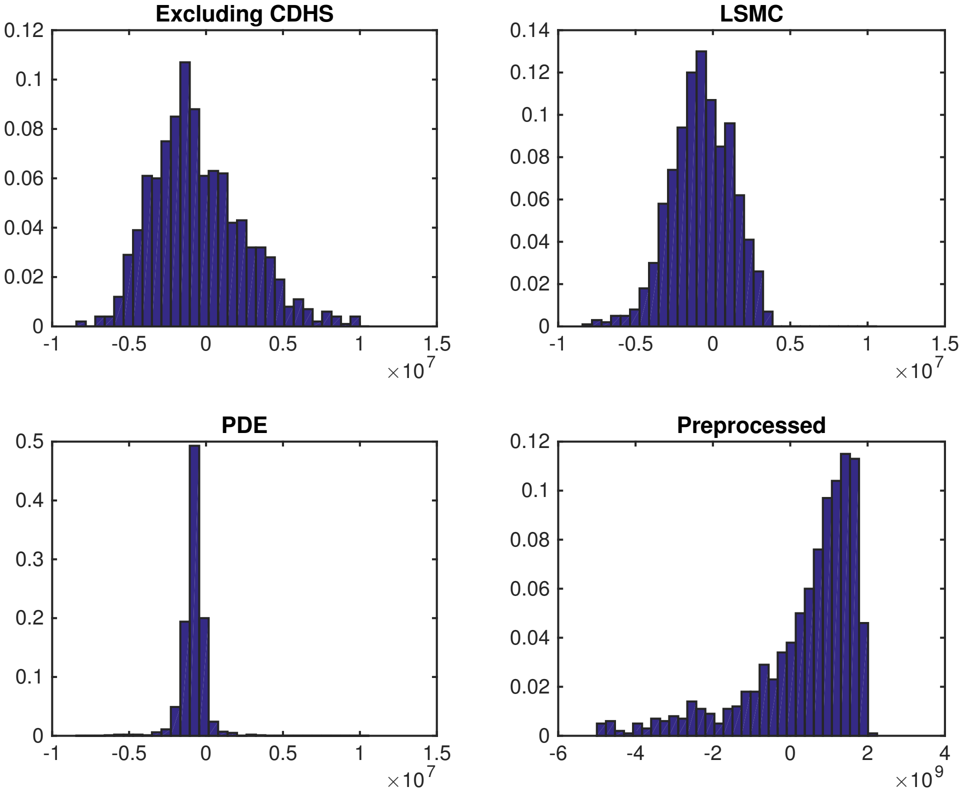

FIGURE 28. Terminal surplus before/after hedging with delta calculations based on formula (6.11) ($\sigma = 0.3$)

6.7. **Conclusions and future work.** Even though the example of the AG-43 reserving exercise for a GLWB rider in Test Case II is only based on many simplifying assumptions, it is significantly more complex than the computation of risk measures for GMAB liabilities in Test Case I in at least two aspects:

(1) Test case I involves only one period for each of the inner loops and outer loops, much like examples in existing academic literature on nested simulations. Test Case II makes projections in outer loops of multiple periods, each of which invokes an inner-loop calculation.

(2) There is only one risk driver for inner loops and for outer loops in Test Case I, for which all methods work reasonably well, albeit with different degrees of accuracy and efficiency. The risk-neutral valuation of GLWB in

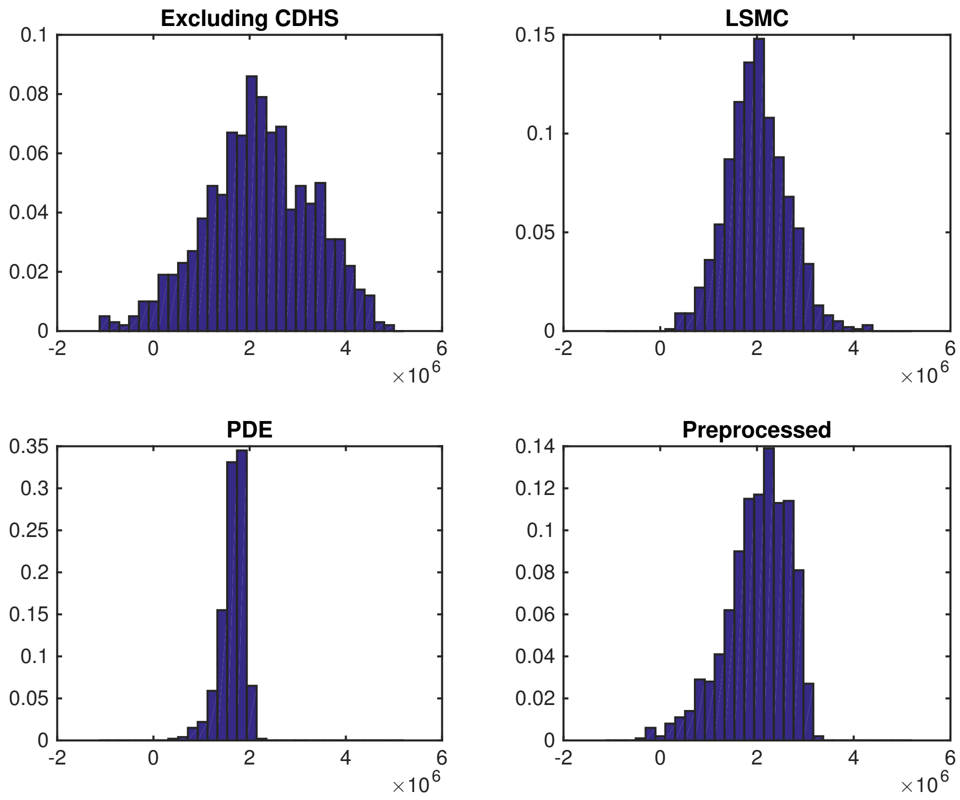

FIGURE 29. Terminal surplus before/after hedging with delta calculations based on formula (6.11) $(\sigma = 0.1)$

Test Case II has four dependent variables, a high-dimension problem for which run time increases so much that even a simple convergence test becomes difficult. For Test Case II, we do not directly test accuracy for outer-loop statistics as done with Test Case I. Nonetheless, a comparison for different methods is performed for inner-loop calculations.

Based on numerical experiments in this section, we draw the following conclusions:

(1) Analytical and numerical PDE methods are in general the most efficient and accurate approaches given a small computation budget. This observation is largely based on the accuracy test and efficiency test on inner-loop calculations as well as the visual effect of hedging programs.

(2) Least-Squares Monte Carlo methods are significant improvements of crude Monte Carlo methods. A collection of lower term polynomials for basis functions work sufficiently well for low-dimension problems, but one may not be able to claim the same for high-dimension problems.

(3) The method of preprocessed inner loops is easiest to implement. Similar to the LSMC, it can be quite efficient in low-dimension problems but appears much less so in high dimensions. The advantage of its efficiency is the use of a small number of grid points. If the spacial structure of the functional relationship in inner-loop calculations exhibits sharp turns, then the interpolation would be inaccurate without using very dense grids, which then diminishes its efficiency.

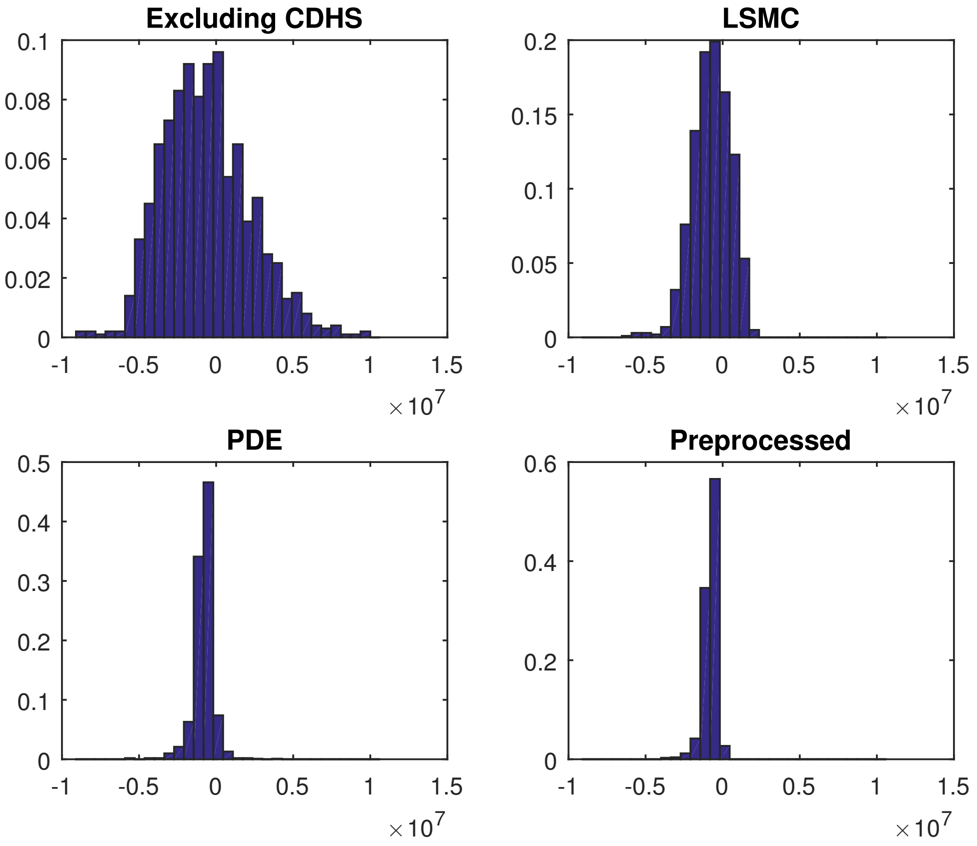

FIGURE 30. Terminal surplus before/after hedging with delta calculations based on formula (6.17) ($\sigma = 0.3$)

A scientific study is not complete without a note of caution on the limitation of its research findings. We briefly comment on constraints of each of the methods discussed in this section as well as potential future work to address these problems:

(1) PDE approaches often require much more stochastic analysis. Although this is not a disadvantage per se, it is harder to explain and to understand until stochastic analysis becomes a common language for communication among actuaries.

(2) Both LSMC and preprocessed methods in this section are for the most part based on naive sampling of the mapping between risk factors and items calculated by inner loops. For example, the equidistant grid may not be a good choice for the surface with sharp turns and corners. It might be more efficient to sample more points where large changes are detected and to sample fewer points where inner-loop items are relatively stable. Future work is necessary to improve their accuracy by introducing elements of experiment design.

All mathematical formulation and methods in this report can be further extended to work with models of multiple equity indices and combined benefits, although their comparative advantages may be different in high dimensions. One could also include stochastic interest rate and stochastic volatility models. Keep in mind that most well-known interest rate and volatility models developed in the literature are continuous time models, which work to the advantage of PDE methods.

Appendix A. Technical Details

A.1. **Optimal budget allocation.** In Test Case I, all required elements for the optimal budget allocation are available in closed form. Observe that the time-t liability L in (5.1) can be written as a function of the separate account value F_t, i.e., $L = \Pi(F_t)$ for some function Π. To be precise, given that $F_t = x$, the liability is determined by

$$\Pi(x) := e^{-r(T-t)} G\Phi(-d_2(x)) - F_t\Phi(-d_1(x)).$$

Keep in mind that in a nested simulation, we can not observe $\Pi(x)$ directly for a given scenario $\{F_t = x\}$, because $\Pi(x)$ is estimated by a sample mean of discounted values of GMAB benefit payments,

$$\hat{L}(x) := \frac{1}{N}\sum_{k=1}^{N} e^{-r(T-t)}(G - x\mathfrak{L}^k)_+,$$

where $\{\mathfrak{L}^1, \cdots, \mathfrak{L}^N\}$ are mutually independent with common lognormal distribution with location parameter $(r - \sigma_1^2/2)(T-t)$ and scale parameter $\sigma_1\sqrt{T-t}$. Hence, given a outer loop scenario $\{F_t = x\}$, we decompose each point estimator of the inner loop as

$$\hat{L}(x) = \Pi(x) + \bar{Z}^N(x),$$

where the pricing error is defined by

$$\bar{Z}^N(x) := \frac{1}{N}\sum_{k=1}^{N} e^{-r(T-t)}(G - x\mathfrak{L}^k)_+ - \Pi(x),$$

For any $x > 0$, by the strong Law of Large Numbers, we obtain the fact that the pricing error vanishes as the sample size of inner loops goes to infinity:

$$\bar{Z}^N(x) \longrightarrow 0, \qquad \text{almost surely.}$$

It shows that $\hat{L}$ is a consistent estimator of L.

Variance of pricing error

Conditioning on that $F_t = x$, the pricing error $\bar{Z}^N(x)$ has mean 0 and variance $h(x)/N$, where

$$\begin{aligned} h(x) := {}& G^2 e^{-2r(T-t)}\Phi(-d_2(x)) - 2Gxe^{-r(T-t)}\Phi(-d_1(x)) \\ & + x^2 e^{\sigma_1^2(T-t)}\Phi(-d_3(x)) - \left[Ge^{-r(T-t)}\Phi(-d_2(x)) - x\Phi(-d_1(x))\right]^2 \end{aligned} \tag{A.1}$$

and

$$d_3(x) := \frac{\ln(x/G) + (r + 3\sigma_1^2/2)(T-t)}{\sigma_1\sqrt{T-t}}.$$

It can be shown that

$$\begin{aligned} h'(x) = {}& -2Ge^{-r(T-t)}\Phi(-d_1(x)) + 2xe^{\sigma_1^2(T-t)}\Phi(-d_3(x)) \\ & + 2\left[Ge^{-r(T-t)}\Phi(-d_2(x)) - x\Phi(-d_1(x))\right]\Phi(-d_1(x)). \end{aligned}$$

Joint density of liability and pricing error

Let $g_N(y, z)$ be the joint density of $(L, \tilde{Z}_N)$ where $\tilde{Z}_N = \sqrt{N}\bar{Z}_N$ and $L = \Pi(F_t)$. Let q be the lognormal density

$$q(y) := \frac{1}{y\sqrt{2\pi t}\sigma_0} \exp\left\{-\frac{[\ln(y/F_0) - (\mu - \sigma_0^2/2)t]^2}{2\sigma_0^2 t}\right\},$$

and hence

$$q'(y) = -\frac{3\sigma_0^2 t/2 + \ln(y/F_0) - \mu t}{y^2\sqrt{2\pi t}\sigma_0^3 t}\,\mathrm{exp}\left\{-\frac{[\ln(y/F_0) - (\mu - \sigma_0^2/2)t]^2}{2\sigma_0^2 t}\right\}.$$

Conditional variance of pricing error

It follows that

$$\begin{aligned}\Theta(u) &= \frac{1}{2}\int_{\mathbb{R}} z^2 g_N(u,z)\mathrm{d}z\\ &= \frac{1}{2}q(\Pi^{-1}(u))h(\Pi^{-1}(u))(\Pi^{-1})'(y).\end{aligned}$$

Recall that $l_p = \Pi(f_p)$. Then,

$$\begin{aligned}\theta_p &:= -\Theta'(l_p)\\ &= \frac{q'(f_p)h(f_p) + q(f_p)h'(f_p)}{2\Pi'(f_p)^2} - \frac{q(f_p)h(f_p)\Pi''(f_p)}{2\Pi'(f_p)^3}.\end{aligned}$$

Recall the delta and gamma of the Black-Scholes put option are given by

$$\Pi'(x) = -\Phi(-d_1(x)), \qquad \Pi''(x) = \frac{\phi(d_1(x))}{\sigma_1 x\sqrt{T-t}},$$

where ϕ is the standard normal density.

Optimal allocation

Let γ_1 be the computation cost of each inner loop, γ_0 be the cost of each outer loop and Γ be the total computation budget. The goal of optimal allocation is to find the optimal n and m such that the mean-squared error, $\mathbb{E}[(\hat{L}_{(\lceil Mp\rceil)} - L)^2]$, is minimized, given the budget constraint

$$n(m\gamma_1 + \gamma_0) = \Gamma.$$

When $\gamma_0 \approx 0$ and Γ is very large, then the optimal n and m can be determined by

$$m^* \approx \left(\frac{2\theta_p^2\Gamma}{p(1-p)\gamma_1}\right)^{1/3}$$

and

$$n^* \approx \left(\frac{p(1-p)}{2\theta_p^2\gamma_1^2}\right)^{1/3}\Gamma^{2/3}.$$

A.2. **Sequential allocation.** Here is an explanation for the optimization strategy (5.9) in the sequential allocation algorithm. Observe that

$$\begin{aligned}\mathbb{P}(\hat{L}_k' < V|\hat{L}_k > V) &= \mathbb{P}(\hat{L}_k + \hat{Z} < V|\hat{L}_k > V)\\ &= \mathbb{P}(\hat{Z} - L(\omega_k) < -m_k(\hat{L}_k - V) - (L(\omega_k) - V)|\hat{L}_k > V),\end{aligned}$$

where $\hat{Z}$ is an estimator of the liability, i.e., $\hat{Z} = e^{-r(T-t)}(G - x\mathfrak{L}^{k+1})_+$. It is assumed that m_k is far larger than 1. Hence the authors use the approximation $-m_k(\hat{L}_k - V) - (L(\omega_k) - V) \approx -m_k(\hat{L}_k - V)$. Then

$$\begin{aligned}\mathbb{P}(\hat{L}_k' < V|\hat{L}_k > V) &\approx \mathbb{P}(\hat{Z} - L(\omega_k) < -m_k(\hat{L}_k - V)|\hat{L}_k > V)\\ &= \mathbb{P}(\hat{Z} - L(\omega_k) < -m_k|\hat{L}_k - V||\hat{L}_k > V).\end{aligned}$$

Similarly, one can approximate

$$\begin{aligned}\mathbb{P}(\hat{L}_k' > V|\hat{L}_k < V) &\approx \mathbb{P}(\hat{Z} - L(\omega_k) > -m_k(\hat{L}_k - V)|\hat{L}_k < V) \\ &= \mathbb{P}(\hat{Z} - L(\omega_k) > m_k|\hat{L}_k - V||\hat{L}_k > V).\end{aligned}$$

For any random variable with mean μ and variance σ^2 and any number $a > 0$, the one-sided Chebyshev inequality says that

$$\mathbb{P}(X \geq \mu + a) \leq \frac{\sigma^2}{\sigma^2 + a^2}, \qquad \mathbb{P}(X \leq \mu - a) \leq \frac{\sigma^2}{\sigma^2 + a^2}.$$

It is obvious that $\hat{Z}$ has the mean 0 and the variance given by $\sigma_k = h(F(\omega_k))$ where h is given by (A.1). Therefore,

$$\begin{aligned}\mathbb{P}(\hat{L}_k' < V|\hat{L}_k > V) &\approx \mathbb{P}(\hat{Z} - L(\omega_k) < -m_k|\hat{L}_k - V||\hat{L}_k > V) \\ &\leq \frac{\sigma_k^2}{\sigma_k^2 + m_k^2|\hat{L}_k - V|^2} = \left(1 + \frac{m_k^2}{\sigma_k^2}|\hat{L}_k - V|\right)^{-1}.\end{aligned}$$

A similar result is true for the other case $\mathbb{P}(\hat{L}_k' > V|\hat{L}_k < V)$. Therefore, in order to maximize the chance of a sign change, we want to minimize

$$\frac{m_k^2}{\sigma_k^2}|\hat{L}_k - V|,$$

which explains the search algorithm in (5.9). □

A.3. **LSMC with basis selection.** We now sketch the idea of the Hankel matrix approximation for identifying the complex weights $\{w_m, m = 1, \cdots, M\}$ and the complex nodes $\{\gamma_m, m = 1, \cdots, M\}$. Consider the $(N+1) \times (N+1)$ Hankel matrix H defined as follows:

$$\mathbf{H} = \begin{bmatrix} h_0 & h_1 & \cdots & h_{N-1} & h_N \\ h_1 & h_2 & \cdots & h_N & h_{N+1} \\ \vdots & & & & \vdots \\ h_{N-1} & h_N & \cdots & h_{2N-2} & h_{2N-1} \\ h_N & h_{N+1} & \cdots & h_{2N-1} & h_{2N} \end{bmatrix}.$$

For the practical purpose of this application, we shall consider only the case where the Hankel matrix is real-valued. Then we can solve for the eigenvalue problem

$$\mathbf{Hu} = \sigma\mathbf{u},$$

where σ is a real and non-negative eigenvalue, and $\mathbf{u} = (u_0, \cdots, u_N)$ is the corresponding eigenvector. By the definition of Hankel matrix, it is easy to show that $\{h_m, m = 0, \cdots, 2N\}$ satisfies the following recursive relation:

$$\sum_{n=0}^{N} h_{k+n}u_n = \sigma u_k, \qquad k = 0, \cdots, N.$$

If we extend the eigenvector **u** to a periodic sequence of period $L(L > N)$ and where $u_k = 0$ for $N < k < L$, then we can define an inhomogeneous recurrence relation

(A.2) $$\sum_{n=0}^{N} x_{k+n}u_n = \sigma u_k, \qquad k \geq 0,$$

given the initial conditions $x_k = h_k$ for $k = 0, 1, \cdots, N-1$. The solution to the recurrence relation is unique and can be solved by

$$x_{N+k} = -\sum_{n=0}^{N-1} \frac{u_n}{u_N}x_{k+n} + \sigma\frac{u_k}{u_N}, \qquad k \geq 0,$$

provided that $u_N \neq 0$. It is well known that the solution to (A.2) can be written as the sum of a general solution to the corresponding homogeneous recurrence relation and a particular solution, denoted by $\{x_k^{(p)}, k \geq 0\}$:

$$x_k = \sum_{n=1}^{N} w_n \gamma_n^k + x_k^{(p)}, \qquad k \geq 0,$$

where $\{\gamma_1, \cdots, \gamma_N\}$ is the set of N roots to the eigen-polynomial $P_{\mathsf{u}}(z) = \sum_{k=1}^{N} u_k z^k$. A particular solution is given by

$$x_k^{(p)} = \frac{\sigma}{L} \sum_{l=0}^{L-1} \alpha^{lk} \frac{P_{\mathsf{u}}(\alpha^{-l})}{P_{\mathsf{u}}(\alpha^{l})}, \qquad \alpha := \mathsf{exp}\left\{\frac{2\pi i}{L}\right\},$$

because for all $k = 0, 1, \cdots, N$,

$$\sum_{n=0}^{N} x_{k+n}^{(p)} u_n = \frac{\sigma}{L} \sum_{l=0}^{L-1} \alpha^{lk} \frac{P_{\mathsf{u}}(\alpha^{-l})}{P_{\mathsf{u}}(\alpha^{l})} \sum_{n=0}^{N} u_n \alpha^{nl} = \frac{\sigma}{L} \sum_{l=0}^{L-1} \alpha^{lk} P_{\mathsf{u}}(\alpha^{-l}) = \frac{\sigma}{L} \sum_{n=0}^{N} \left(\sum_{l=0}^{L-1} \alpha^{(n-k)l} \right) u_k = \sigma u_k.$$

Note that $|x_k^{(p)}| \leq \sigma$ for all $k \geq 0$. Therefore, it follows immediately that

$$\left| h_k - \sum_{n=1}^{N} w_n \gamma_n^k \right| \leq \sigma, \qquad k = 0, 1, \cdots, 2N.$$

To find the approximation, we rank all the eigenvalues of **H** in a decreasing order:

$$\sigma_0 \geq \sigma_1 \geq \cdots \geq \sigma_N.$$

If we choose an eigenvalue σ_M $(0 \leq M \leq N)$ smaller than the level of error tolerance ϵ in (5.13), then we can obtain the expected approximation with absolute error of at most σ. Beylkin and Monzon made the observation that only first M weights $\{w_1, \cdots, w_M\}$ are larger than σ_M, and hence they made the claim that $\sum_{m=1}^{M} w_m \gamma_m^k$ has the " nearly optimal" representation of the Hankel matrix. Observe that one can obtain the unknown weights $(w_1, \cdots, w_N)$ by finding the unique solution to the Vandermonde system:

$$h_k - \sigma x_k^{(p)} = \sum_{n=1}^{N} w_n \gamma_n^k, \qquad 0 \leq k < N.$$

The equation is also valid for $N \leq k \leq 2N$. Thus, the authors recommended using the least-squares solution $(\rho_1, \cdots, \rho_N)$ to the overdetermined problem:

$$h_k = \sum_{n=1}^{N} \rho_n \gamma_n^k, \qquad 0 \leq k \leq 2N.$$

A.4. **Derivation of PDE.** Define the process

$$\begin{aligned} X_t := \mathbb{E}\Big[& \int_0^\infty e^{-rs} \, {}_s p_x \, h \, G_s I(F_s < 0) \mathsf{d}s - \int_0^\infty e_s^{-rs} p_x (m_w G_s + m F_s) I(F_s > 0) \mathsf{d}s \\ & - \int_0^\infty e^{-rs} \mu_s^l f(Y_s) \, {}_s p_x c_s F_s I(F_s > 0) \mathsf{d}s \Big| \mathcal{F}_t \Big] \\ = \; & e^{-rt} V_t + \int_0^t e^{-rs} \, {}_s p_x \, h \, G_s I(F_s < 0) \mathsf{d}s - \int_0^t e^{-rs} \, {}_s p_x (m_w G_s + m F_s) I(F_s > 0) \mathsf{d}s \\ & - \int_0^t e^{-rs} \mu_s^l f(Y_s) \, {}_s p_x c_s F_s I(F_s > 0) \mathsf{d}s, \end{aligned}$$

Therefore,

$$\mathsf{d}V_t = e^{rt} \mathsf{d}X_t + \Big[rV_t - \, {}_t p_x h G_t I(F_t < 0) + \, {}_t p_x (m_w G_t + m F_t) I(F_t > 0) + \mu_t^l f(Y_t) \, {}_t p_x c_t F_t I(F_t > 0) \Big] \mathsf{d}t.$$

(A.3)

Recall that by the strong Markov property of the underlying process, we know that there must exist a smooth function $v(t,x,y,z)$ such that

$$V_t = v(t, F_t, G_t, \ {}_tp_x).$$

Applying Itô's formula, we obtain

$$\begin{aligned}
\text{(A.4)} \qquad & \mathrm{d}v(t, F_t, G_t, \ {}_tp_x) = v_t(t, F_t, G_t, \ {}_tp_x)\mathrm{d}t + v_x(t, F_t, G_t, \ {}_tp_x)\mathrm{d}F_t \\
& +v_y(t, F_t, G_t, \ {}_tp_x)\mathrm{d}G_t + v_z(t, F_t, G_t, \ {}_tp_x)\mathrm{d}_tp_x + \frac{1}{2}v_{xx}(t, F_t, G_t, \ {}_tp_x)\mathrm{d}\langle F\rangle_t \\
= \ & v_t(t, F_t, G_t, \ {}_tp_x)\mathrm{d}t + v_x(t, F_t, G_t, \ {}_tp_x)[(r-m)F_t\mathrm{d}t + p\sigma F_t\mathrm{d}B_t - (m_w+h)G_t\mathrm{d}t] \\
& +v_y(t, F_t, G_t, \ {}_tp_x)\mathrm{d}G_t + v_z(t, F_t, G_{t,t}p_x)\mathrm{d}_tp_x + \frac{1}{2}p^2\sigma^2 v_{xx}(t, F_t, G_t, \ {}_tp_x)F_t^2\mathrm{d}t \\
= \ & \Bigg[v_t(t, F_t, G_t, \ {}_tp_x) + v_x(t, F_t, G_t, \ {}_tp_x)[(r-m)F_t - (m_w+h)G_t] \\
& +\frac{1}{2}v_{xx}(t, F_t, G_t, \ {}_tp_x)p^2\sigma^2F_t^2 - v_z(t, F_t, G_t, \ {}_tp_x)[\mu^d_{x+t} + f(Y_t)\mu^l_t]\,{}_tp_x \\
& +v_y(t, F_t, G_t, \ {}_tp_x)\rho G_t\Bigg]\mathrm{d}t + v_y(t, F_t, G_t, \ {}_tp_x)G_t\mathrm{d}L_t + p\sigma F_t v_x(t, F_t, G_t, \ {}_tp_x)\mathrm{d}B_t,
\end{aligned}$$

where

$$\mathrm{d}G_t = G_t\mathrm{d}L_t + \rho G_t\mathrm{d}t.$$

Comparing (A.3) and (A.4), we observe immediately that $\mathrm{d}t$ terms should be equal and the term $v_y(t, F_t, G_t, \ {}_tp_x)G_t\mathrm{d}L_t$ must be zero. We obtain the PDE for $t > 0$ and $x, y > 0, 0 < z < 1$:

$$\begin{aligned}
\text{(A.5)} \qquad & v_t(t,x,y,z) + v_x(t,x,y,z)[(r-m)x - (m_w+h)y] + \frac{1}{2}v_{xx}(t,x,y,z)p^2\sigma^2x^2 \\
& -v_z(t,x,y,z)[\mu^d_{x+t} + f(x/y)\mu^l_t]z + v_y(t,x,y,z)\rho y \\
& = rv(t,x,y,z) + z(m_wy + mx) + \mu^l_t f(x/y)c_txz,
\end{aligned}$$

subject to

$$\text{(A.6)} \qquad v_y(t,x,y,z)|_{x=y} = 0.$$

Note that if $F_t = 0$ then $F_s \leq 0$ for all $s \geq t$. Hence, given that $F_t = 0$, we must have

$$V_t = \ {}_tp_x \int_t^\infty e^{-r(s-t)} \ {}_{s-t}p_{x+t}hG_te^{\rho(s-t)}\mathrm{d}s = \ {}_tp_xhG_t\int_0^\infty e^{-(r-\rho)s} \ {}_sp_{x+t}\mathrm{d}s = \ {}_tp_xhG_t\,\overline{a}_{x+t}.$$

Keep in mind that the life annuity $\overline{a}_{x+t}$ is calculated with the force of interest $r - \rho$. The boundary condition for the original PDE is

$$\text{(A.7)} \qquad v(t, 0, y, z) = hyz\overline{a}_{x+t}.$$

Since $\lim_{t\to\infty} \ {}_tp_x = 0$, it is also clear that

$$\lim_{t\to\infty} v(t,x,y,z) = 0.$$

We can reduce the dimension of the PDE by defining a function $u(t,s)$ such that

$$\text{(A.8)} \qquad v(t,x,y,z) = zyu(t, x/y).$$

It follows immediately that the PDE reduces to (6.15). Observe that

$$v_y(t,x,y,z)|_{x=y} = zu(t,s)|_{s=1} + zyu_s(t,s)\frac{\partial s}{\partial y}\bigg|_{s=1} = zu(t,1) - zu_s(t,1).$$

In view of (A.6), we obtain the corresponding boundary condition for u (6.16a). Similarly, we obtain the boundary condition (6.16b) and the terminal condition (6.16c) from the corresponding conditions for v.

A.5. **Dynamics of mixed surplus and hedging portfolio.** The self-financing condition (6.12) implies that

$$\Delta h_t S_t + \Delta R_t - rR_{t-\Delta t}\Delta t - \mathsf{CF}(\Delta t) = 0.$$

We add and subtract the term $S_{t-\Delta t}\Delta h_t$ and obtain

$$\Delta h_t S_t - S_{t-\Delta t}\Delta h_t + S_{t-\Delta t}\Delta h_t + \Delta R_t - rR_{t-\Delta t}\Delta t - \mathsf{CF}(\Delta t) = 0,$$

or equivalently,

$$\Delta S_t \Delta h_t + S_{t-\Delta t}\Delta h_t + \Delta R_t - rR_{t-\Delta t}\Delta t - \mathsf{CF}(\Delta t) = 0.$$

Taking Δt to zero, we find the continuous-time version of the self-financing condition

$$\mathsf{d}h_t \mathsf{d}S_t + S_t \mathsf{d}h_t + \mathsf{d}R_t = rR_t \mathsf{d}t + \mathsf{CF}(\mathsf{d}t),$$

where

$$\mathsf{CF}(\mathsf{d}t) = \Big[{}_tp_x(m_w G_t + mF_t)I(F_t > 0) + \mu^l_t f(Y_t)\, {}_tp_x\, c_t F_t I(F_t > 0) - \,{}_tp_x h G_t I(F_t \leq 0)\Big]\,\mathsf{d}t.$$

Finally, we are ready to compute the dynamics of the portfolio:

$$\begin{aligned}
\mathsf{d}H_t &= \mathsf{d}(h_t S_t + R_t)\\
&= h_t \mathsf{d}S_t + S_t \mathsf{d}h_t + \mathsf{d}S_t \mathsf{d}h_t + \mathsf{d}R_t\\
&= h_t \mathsf{d}S_t + rR_t \mathsf{d}t + \mathsf{CF}(\mathsf{d}t)\\
&= h_t \mathsf{d}S_t + r(H_t - h_t S_t)\mathsf{d}t + \mathsf{CF}(\mathsf{d}t). \qquad \text{(A.9)}
\end{aligned}$$

Recall that v satisfies the PDE (A.5), then

$$\begin{aligned}
\mathsf{d}v(t, F_t, G_t, {}_tp_x) &= rv(t, F_t, G_t, {}_tp_x)\mathsf{d}t + \mathsf{CF}(\mathsf{d}t) + p\sigma F_t v_x(t, F_t, G_t, {}_tp_x)\mathsf{d}B_t\\
&= r[v(t, F_t, G_t, {}_tp_x) - v_x(t, F_t, G_t, {}_tp_x)F_t]\mathsf{d}t + \mathsf{CF}(\mathsf{d}t)\\
&\quad + \frac{F_t}{S_t} v_x(t, F_t, G_t, {}_tp_x)\mathsf{d}S_t.
\end{aligned}$$

Note that the term ${}_tp_x h G_t I(F_t \leq 0)$ included in $\mathsf{CF}(\mathsf{d}t)$ comes from the boundary condition (A.7). Recall that we consider

$$\Delta_t = \frac{F_t}{S_t} v_x(t, F_t, G_t, {}_tp_x).$$

It follows that

$$\mathsf{d}v(t, F_t, G_t, {}_tp_x) = r[v(t, F_t, G_t, {}_tp_x) - \Delta_t S_t]\mathsf{d}t + \mathsf{CF}(\mathsf{d}t) + \Delta_t \mathsf{d}S_t.$$

Note that the GLWB liability $V_t = v(t, F_t, G_t, {}_tp_x)$ and $h_t = \Delta_t$. Subtracting (A.10) from (A.9) yields

$$\mathsf{d}(V_t - H_t) = r(V_t - H_t)\mathsf{d}t,$$

which implies that for some constant C and all $t \geq 0$

$$e^{-rt}(V_t - H_t) = C.$$

Since $H_0 = 0$, then it shows that $C = V_0$.

$$H_t = V_t - e^{rt} V_0.$$

A.6. **Derivation of analytical solution.** Consider the case where $p = 1$, $\mu^d_{x+t} = \lambda$ and $\mu^l_t = 0$. Then ${}_tp_x = e^{-\lambda t}$ for all $x > 0$ and (assuming $\lambda + r - \rho > 0$)

$$\overline{a}_{x+t} = \frac{1}{r - \rho + \lambda}.$$

The the PDE (6.15) reduces to the ODE

$$\frac{1}{2}\sigma^2 s^2 u_{ss} + [(r - m - \rho)s - (m_w + h)]u_s - [\lambda + r - \rho]u - m_w - ms = 0,$$

subject to the boundary conditions

$$u(1) = u_s(1),$$
$$u(0) = \frac{h}{\lambda + r - \rho}.$$

This reduces to a special case where the future lifetime is exponentially distributed and the lapse rate is a constant that can be combined with the force of mortality. We can find analytical solution to the no-arbitrage value of the GLWB in this case.

If we let $w = \frac{\sigma^2}{4(m_w+h)} s = ks$, the ODE of w becomes

$$2w^2 u_{ww} + \left[\frac{4(r - m - \rho)}{\sigma^2} w - 1\right] u_w - \frac{4(\lambda + r - \rho)}{\sigma^2} u - \frac{4mw}{k\sigma^2} - \frac{4m_w}{\sigma^2} = 0,$$

with boundary conditions

$$u(k) = k u_w(k),$$
$$u(0) = \frac{h}{\lambda + r - \rho}.$$

Let $r^* := \frac{4(\lambda+r-\rho)}{\sigma^2}$, $\nu = \frac{2(r-m-\rho)-\sigma^2}{\sigma^2}$ and $\mu = \sqrt{2r^* + \nu^2}$. The ODE is simplified to

$$2w^2 u_{ww} + [2(\nu + 1)w - 1]u_w - r^* u - \frac{4mw}{k\sigma^2} - \frac{4m_w}{\sigma^2} = 0.$$

The differential equation has the following general solution:

$$\begin{aligned} u(w) = {} & C_1 w^{(1-\nu)/2} \,\text{exp}\left(-\frac{1}{4w}\right) M_{(\nu-1)/2,\mu/2}(\frac{1}{2w}) \\ & + C_2 w^{(1-\nu)/2} \,\text{exp}\left(-\frac{1}{4w}\right) W_{(\nu-1)/2,\mu/2}(\frac{1}{2w}) - \frac{mw}{k(\lambda + m)} + \frac{\frac{(h+m_w)m}{\lambda+m} - m_w}{r - \rho + \lambda}, \end{aligned}$$

where M is the Whittaker-M function and W is the Whittaker-W function. The derivations of these fundamental solutions can be found in Feng and Volkmer [8].

Let us rewrite the two fundamental solutions

$$f_{(\nu-1)/2}(w) = w^{(1-\nu)/2} \,\text{exp}\left(-\frac{1}{4w}\right) M_{(\nu-1)/2,\mu/2}(\frac{1}{2w}),$$

and

$$g_{(\nu-1)/2}(w) = w^{(1-\nu)/2} \,\text{exp}\left(-\frac{1}{4w}\right) W_{(\nu-1)/2,\mu/2}(\frac{1}{2w}).$$

It is easy to show that

$$f'_\kappa(y) = -(\tfrac{1}{2} + \mu + \kappa) f_{\kappa+1}(y), \qquad g'_\kappa(y) = g_{\kappa+1}(y),$$
$$\lim_{y\to\infty} f_\kappa(y) = 0, \qquad \lim_{y\to 0+} g_\kappa(y) = 0.$$

Then we obtain

$$C_1 = \frac{(h+m_w)\lambda\Gamma(1+(\mu-\nu)/2)}{(\lambda+m)(r-\rho+\lambda)\Gamma(1+\mu)} 2^{-(\nu-1)/2},$$
$$C_2 = \frac{C_1[f_{(\nu-1)/2}(k) + k(\mu/2+\nu/2) f_{(\nu+1)/2}(k)] + \frac{\frac{(h+m_w)m}{\lambda+m} - m_w}{r-\rho+\lambda}}{k g_{(\nu+1)/2}(k) - g_{(\nu-1)/2}(k)},$$

Finally, we get the solution of the ODE

$$u(w) = C_1 f_{(\nu-1)/2}(w) + C_2 g_{(\nu-1)/2}(w) - \frac{mw}{k(\lambda+m)} + \frac{\frac{(h+m_w)m}{\lambda+m} - m_w}{r-\rho+\lambda}.$$

Here we can also its derivative with respect to s:

$$u_s = k u_w = k\left[C_2 g_{(\nu+1)/2}(w) - \frac{C_1(\nu+\mu)}{2} f_{(\nu+1)/2}(w)\right] - \frac{m}{\lambda+m}. \tag{A.10}$$

A.7. **Stochastic representation of $u(t,s)$.** Observe that the bivariate process $\{(F_t, G_t), t \geq 0\}$ is a strong Markov process. Recall that the process G has the following stochastic representation:

$$G_t = e^{\rho t} \max\left\{G_0, \sup_{0\le u\le t}\{e^{-\rho u} F_u\}\right\} = e^{\rho t} G_0 \max\left\{1, \sup_{0\le u\le t}\{e^{-\rho u}\tilde{F}_u\}\right\},$$

where $\tilde{F}_0 = F_0/G_0$ and

$$\mathrm{d}\tilde{F}_t = [(r-m)\tilde{F}_t - (m_w+h)G_t]\mathrm{d}t + p\sigma\tilde{F}_t \mathrm{d}B_t.$$

Observe that for $s \geq t \geq 0$

$$G_s = e^{\rho s}\max\left\{e^{-\rho t}G_t, \sup_{t\le u\le s}\{e^{-\rho u}F_u\}\right\}$$
$$= e^{\rho(s-t)}\max\left\{G_t, e^{\rho t}\sup_{t\le u\le s}\{e^{-\rho u}F_u\}\right\}.$$

Then it follows from the strong Markov property that

$$\mathbb{E}[{}_s p_x G_s|\mathcal{F}_t] = {}_t p_x G_t \mathbb{E}^{(Y_t,1)}\left[{}_{t-s}p_{x+t} e^{\rho(s-t)}\max\left\{1, \sup_{0\le u\le s-t}\{e^{-\rho u}\tilde{F}_u\}\right\}\right]$$
$$= {}_t p_x G_t \mathbb{E}^{(Y_t,1)}\left[{}_{t-s}p_{x+t} G_{t-s}\right],$$

where $\mathbb{E}^{(x,y)}$ is the expectation under the probability measure for which $F_0 = x$ and $G_0 = y$. Similarly,

$$\mathbb{E}[{}_s p_x F_s|\mathcal{F}_t] = {}_t p_x F_t\, \mathbb{E}^{(Y_t,1)}\left[{}_{t-s}p_{x+t}\tilde{F}_{s-t}\middle|\mathcal{F}_t\right] = G_t\, \mathbb{E}^{(Y_t,1)}\left[{}_{t-s}p_{x+t}\tilde{F}_{s-t}\right].$$

Then

$$\begin{aligned}
V_t :=\ & {}_tp_x\mathbb{E}\Bigg[\int_t^\infty e^{-r(s-t)}\ {}_{s-t}p_{x+t}\, h\, G_s I(F_s<0)\mathsf{d}s - \int_t^\infty e^{-r(s-t)}\ {}_{s-t}p_{x+t}(m_w G_s + mF_s)I(F_s>0)\mathsf{d}s \\
& - \int_t^\infty e^{-r(s-t)}\mu_s^l f(Y_s)\ {}_{s-t}p_{x+t}c_sF_sI(F_s>0)\mathsf{d}s\Bigg|\mathcal{F}_t\Bigg] \\
=\ & {}_tp_xG_t\mathbb{E}^{(Y_t,1)}\Bigg[\int_0^\infty e^{-rv}\ {}_vp_{x+t}\, h\, G_vI(\tilde{F}_v<0)\mathsf{d}v - \int_0^\infty e^{-rv}\ {}_vp_{x+t}(m_wG_v + mY_t\tilde{F}_v)I(\tilde{F}_v>0)\mathsf{d}v \\
& - \int_0^\infty e^{-rv}\mu_{t+v}^l f(Y_v)\ {}_vp_{x+t}c_{t+v}\tilde{F}_vI(\tilde{F}_v>0)\mathsf{d}v\Bigg].
\end{aligned}$$

Comparing the above expression and (A.8), we obtain the following representation:

$$\begin{aligned}
u(t,s) = \mathbb{E}^{(s,1)}\Bigg[&\int_0^\infty e^{-rv}\ {}_vp_{x+t}\, h\, G_vI(\tilde{F}_v<0)\mathsf{d}v - \int_0^\infty e^{-rv}\ {}_vp_{x+t}(m_wG_v + m\tilde{F}_v)I(\tilde{F}_v>0)\mathsf{d}v \\
& - \int_0^\infty e^{-rv}\mu_{t+v}^l f(Y_v)\ {}_vp_{x+t}c_{t+v}\tilde{F}_vI(\tilde{F}_v>0)\mathsf{d}v\Bigg].
\end{aligned}$$

Appendix B. Numerical Algorithms

For numerical purposes, we truncate the state space of t to $(0,T)$ for enough big $T>0$. Consider the following conditions:

$$\begin{aligned}
u(T,s) &= 0, \\
u(t,0) &= h\bar{a}_{x-t}, \\
u_s(t,1) &= u(t,1).
\end{aligned}$$

For ease of presentation, we define $c_1(s), c_2(s), c_3(s) and c_4(s)$ as follows:

$$\begin{aligned}
c_1(s) &= \frac{1}{2}p^2\sigma^2s^2, \\
c_2(s) &= (r-m-\rho)s - (m_w+h), \\
c_3(t,s) &= -[\mu_{x+t}^d + \mu_t^l f(s) + r - \rho], \\
c_4(t,s) &= -(m_w+ms) - \mu_t^l f(s)c_ts.
\end{aligned}$$

We then introduce $w(t,s) = u(T-t,s)$ defined on $(0,T)\times[0,1]$, which satisfies the forward PDE

$$w_t = c_1(s)w_{ss} + c_2(s)w_s + c_3(T-t,s)w + c_4(T-t,s) \tag{B.1}$$

subject to the boundary conditions:

$$w(0,s) = 0, \tag{B.2}$$

$$w(t,0) = h\bar{a}_{x+T-t}, \tag{B.3}$$

$$w_s(t,1) = w(t,1). \tag{B.4}$$

We set up the grids (t_i,s_j) as follows and denote the solution $w(t_i,s_j)$ on the grid points by w_j^i, where

$$\begin{aligned}
t_i &= (i-1)\triangle t, \quad \text{for } i = 1,\cdots,N_t+1, \ \ N_t := T/\triangle t, \\
s_j &= j\triangle s, \qquad \text{for } j = 0,\cdots,N_s, \ \ N_s := 1/\triangle s.
\end{aligned}$$

Discretization of (B.1) gives

$$\begin{aligned}\frac{w_j^{i+1}-w_j^i}{\triangle t} =&c_1(j\triangle s)\frac{w_{j+1}^{i+1}+w_{j-1}^{i+1}-2w_j^{i+1}}{(\triangle s)^2}+c_2(j\triangle s)\frac{w_{j+1}^{i+1}-w_{j-1}^{i+1}}{2\triangle s}\\ &+c_3(T-(i-1)\triangle t, j\triangle s)w_j^{i+1}+c_4(T-(i-1)\triangle t, j\triangle s).\end{aligned}$$

For brevity, we use the notation

$$\alpha=\frac{1}{\triangle t},\ \beta_j=\frac{c_1(j\triangle s)}{(\triangle s)^2},\ \gamma_j=\frac{c_2(j\triangle s)}{2\triangle s},$$

$$c_{3,j}^i=c_3(T-(i-1)\triangle t, j\triangle s),\ c_{4,j}^i=c_4(T-(i-1)\triangle t, j\triangle s).$$

For $j=1,\cdots,N_s$, the recursive relation can be rewritten as

(B.5) $$-(\beta_j-\gamma_j)w_{j-1}^{i+1}+(\alpha+2\beta_j-c_{3,j}^i)w_j^{i+1}-(\beta_j+\gamma_j)w_{j+1}^{i+1}=\alpha w_j^i+c_{4,j}^i,$$

The Neumann condition (B.4) yields

$$w_{N_s+1}^{i+1}=w_{N_s-1}^{i+1}+2\triangle s w_{N_s}^{i+1}.$$

For $j=N_s$, we obtain

$$-2\beta_{N_s}w_{N_s-1}^{i+1}+(\alpha+2\beta_{N_s}-c_{3,N_s}^i-(\beta_{N_s}+\gamma_{N_s})2\triangle s)w_{N_s}^{i+1}=\alpha w_{N_s}^i+c_{4,N_s}^i.$$

Combining boundary condition (B.3) and recursive relation (B) for $j=1$ gives

$$(\alpha+2\beta_1-c_{3,1}^i)w_1^{i+1}-(\beta_1+\gamma_1)w_2^{i+1}=\alpha w_1^i+c_{4,1}^i+(\beta_1-\gamma_1)w_0^{i+1}.$$

We set the vector

$$\mathbf{w}^i=(w_1^i,\cdots,w_{N_s}^i)^\top.$$

Then (B) translates to

$$\mathbf{B}^i\mathbf{w}^{i+1}=\alpha\mathbf{w}^i+\mathbf{c}_4^i+((\beta_1-\gamma_1)w_0^{i+1},0,\cdots,0)^\top,$$

where $\mathbf{c}_4^i=(c_{4,1}^i,\cdots,c_{4,N_s}^i)^\top$. The N_s by N_s matrix B^i is given by

$$\mathbf{B}^i=\begin{pmatrix}\alpha+2\beta_1-c_{3,1}^i & -(\beta_1+\gamma_1) & 0 & \cdots 0 & 0 & 0\\ -(\beta_2-\gamma_2) & \alpha+2\beta_2-c_{3,2}^i & -(\beta_2+\gamma_2) & \cdots 0 & 0 & 0\\ \vdots & \vdots & \vdots & \vdots\ \vdots & \vdots & \vdots\\ 0 & 0 & 0 & \cdots 0 & -2\beta_{N_s} & \alpha+2\beta_{N_s}-c_{3,N_s}^i-(\beta_{N_s}+\gamma_{N_s})2\triangle s\end{pmatrix}.$$

On the j-th row $(j=2,\cdots,N_s-1)$, the tridiagonal elements are

$$-(\beta_j-\gamma_j),\ (\alpha+2\beta_j-c_{3,j}^i),\ -(\beta_j+\gamma_j).$$

We can determine **w** by

$$\mathbf{w}^{i+1}=(\mathbf{B}^i)^{-1}\left[\alpha\mathbf{w}^i+\mathbf{c}_4^i+((\beta_1-\gamma_1)w_0^{i+1},0,\cdots,0)^\top\right],$$

marching alternatively in the i direction, starting from $i=1$, where we have the initial condition

$$\mathbf{w}^1=0.$$

References

[1] Beylkin, G. and Monzon, L. (2005). On approximation of functions by exponential sums. *Applied and Computational Harmonic Analysis*, 19(1): 17–48.

[2] Broadie, M., Du, Y., and Moallemi, C. C. (2015). Risk estimation via regression. *Operations Research* (5): 1077–1097.

[3] Gorski, L. M., Brown and R. A. B. (2005). *Recommended Approach for Setting Regulatory Risk-Based Capital Requirements for Variable Annuities and Similar Products*. American Academy of Actuaries, Boston, MA.

[4] Feng, R. (2014). A comparative study of risk measures for guaranteed minimum maturity benefits by a PDE method. *North American Actuarial Journal*, 18(4), 455–466

[5] Feng, R. and Huang, H. (2016) Statutory financial reporting for variable annuity guaranteed death benefits: Market practice, mathematical modeling and computation, *Insurance: Mathematics and Economics* 67: 54–64.

[6] Feng, R. and Jing, X. (2016) Analytic valuation and hedging of variable annuity guaranteed lifetime withdrawal benefits, Preprint.

[7] Feng, R. and Volkmer, H. (2012). Analytical calculation of risk measures for variable annuity guaranteed benefits. *Insurance: Mathematics and Economics* 51(3): 636–648.

[8] Feng, R. and Volkmer, H. V. (2016). An identity of hitting times with application to the valuation of guaranteed minimum withdrawal benefit. *Mathematics and Financial Economics* 10(2): 127–149.

[9] Gordy, M. B. and Juneja, S. (2010). Nested simulation in portfolio risk measurement. *Management Science* 56(10): 1833–1848.

[10] Hardy, M. R. (2003) *Investment Guarantees: Modeling and Risk Management for Equity-Linked Life Insurance*. John Wiley & Sons

[11] Koursaris, A. (2011) A least squares Monte Carlo approach to liability proxy modelling and capital calculation. Barrie & Hibbert, technical report.

[12] Liu, M. and Staum, J. (2010) Stochastic kriging for efficient nested simulation of expected shortfall. *Journal of Risk* 12(3): 3–27.

[13] Longstaff, F. A. and Schwartz, E. S. (2001). Valuing American options by simulation: A simple least-squares approach. *Review of Financial Studies*, 14: 113–147.

[14] Milevsky, M. A. and Salisbury, T. S. (2006). Financial valuation of guaranteed minimum withdrawal benefits. *Insurance: Mathematics and Economics*, 38(1): 21–38.

[15] Stentoft, L. (2004). Convergence of the Least Squares Monte Carlo approach to American option valuation, *Management Science* 50(9): 1193–1203.

[16] Ulm, E. R. (2008). Analytic solution for return of premium and rollup guaranteed minimum death benefit options under some simple mortality laws. *ASTIN Bulletin* 38(2): 543–563.

Nested Stochastic Modeling for Insurance Companies

SPONSOR Financial Reporting Section

Modeling Section

AUTHORS

LEAD INVESTIGATOR:

Runhuan Feng, PhD, FSA, CERA
University of Illinois at Urbana-Champaign

ASSISTANTS:

Zhenyu Cui, PhD
Steven Institute of Technology

Peng Li, MS
Central University of Finance and
Economics

Caveat and Disclaimer

USE THIS FOR EXPERIENCE STUDIES AND DIR:

This study is published by the Society of Actuaries (SOA) and contains information from a variety of sources. It may or may not reflect the experience of any individual company. The study is for informational purposes only and should not be construed as professional or financial advice. The SOA does not recommend or endorse any particular use of the information provided in this study. The SOA makes no warranty, express or implied, or representation whatsoever and assumes no liability in connection with the use or misuse of this study.

USE THIS FOR PRACTICE RESEARCH:

The opinions expressed and conclusions reached by the authors are their own and do not represent any official position or opinion of the Society of Actuaries or its members. The Society of Actuaries makes no representation or warranty to the accuracy of the information.