ELSEVIER



Insurance: Mathematics and Economics

journal homepage: www.elsevier.com/locate/ime



Fast and efficient nested simulation for large variable annuity portfolios: A surrogate modeling approach



X. Sheldon Lin, Shuai Yang*

Department of Statistical Sciences, University of Toronto, 100 St. George Street, Toronto, Ontario, Canada M5S 3G3

ARTICLE INFO

Article history: Received February 2019 Received in revised form October 2019 Accepted 3 January 2020 Available online 15 January 2020

Keywords: Variable annuity portfolio Nested-simulation Surrogate modeling Spline regression Population sampling

ABSTRACT

The nested-simulation is commonly used for calculating the predictive distribution of the total variable annuity (VA) liabilities of large VA portfolios. Due to the large numbers of policies, inner-loops and outer-loops, running the nested-simulation for a large VA portfolio is extremely time consuming and often prohibitive. In this paper, the use of surrogate models is incorporated into the nested-simulation algorithm so that the relationship between the inputs and the outputs of a simulation model is approximated by various statistical models. As a result, the nested-simulation algorithm can be run with much smaller numbers of different inputs. Specifically, a spline regression model is used to reduce the number of outer-loops and a model-assisted finite population estimation studies, our proposed algorithm is able to accurately approximate the predictive distribution of the total VA liability at a significantly reduced running time.

© 2020 Elsevier B.V. All rights reserved.

1. Introduction

Variable annuities (VA) have gained significant market share over the last several decades since its introduction, especially in the North American annuity market. According to LIMRA Secure Retirement Institutes' Fourth Quarter 2017 U.S. Individual Annuity Sales Survey, the total VA sales in the U.S. market were \$95.6 billion in 2017. In addition, Jackson National Life, TIAA and AXA US all sold over \$10 billion VA and the top 20 VA sellers together share 93% of the total sales in 2017. Nowadays many major insurance companies are managing very large variable annuity portfolios.

Typically a variable annuity is embedded with guarantees which provide death benefits or living protection. There are two main categories of guarantees: guaranteed minimum death benefits (GMDBs) and guaranteed minimum living benefits (GMLB). The GMDB guarantees an amount of death benefit to the policy holder upon death during the policy term. Normally the GMDB is embedded in all variable annuities whereas the GMLBs are offered as riders. Two most commonly elected GMLB riders are guaranteed minimum accumulation benefit (GMAB) and guaranteed minimum withdrawal benefit (GMWB). The GMAB rider guarantees an amount of the accumulated value of the policyholder's account at the contract maturity. The GMWB rider guarantees a stream of periodic benefit payments to the policyholder regardless of underlying investment growth. A detailed introduction

* Corresponding author. E-mail address: shuai.yang@mail.utoronto.ca (S. Yang).

https://doi.org/10.1016/j.insmatheco.2020.01.002 0167-6687/© 2020 Elsevier B.V. All rights reserved. of the guarantees can be found in, for example, the Insured Retirement Institute 2017 Fact Book. These embedded guarantees often resemble the characteristics of a series of path-dependent put options with time-varying strikes prices. In most situations, close-form expressions are not available for calculating the VA liabilities resulting from the guarantees. Hence the insurance company normally uses the so-called *nested-simulation algorithm* to calculate the liabilities.

The nested simulation is a two-step simulation algorithm: an outer-loop simulation and an inner-loop simulation, as demonstrated in Fig. 1.1 for variable annuity portfolios valuation. Without loss of generality we denote the valuation time and the future time point by t = 0 and t = 1, respectively. In the first step each policy's account value is projected from t = 0 to t = 1 using many outer-loops where each outer-loop represents a realworld scenario. In the second step a large number of inner-loops are simulated at each outer-loop to calculate the fair value of the VA liability by averaging the present values of the insurer's cash flows. The sum of liabilities over all policies among different outer-loop simulations give the distribution of the total liability at that future time point and that distribution is usually referred to as the *predictive total liability distribution*.

Due to demographic variation and different guarantee designs, a variable annuity portfolio is often constituted by highly non-homogeneous contracts. As a result, running the nested simulation for a large VA portfolio can be time consuming and sometimes prohibitive. For example, assuming a variable annuity portfolio with 100,000 policies and a nested simulation algorithm with 1000 outer-loops and 10,000 inner-loops nested in



Fig. 1.1. The flow chart illustration of the nested simulation.

each outer-loop. If one wishes to find the predictive distribution with a computer that can process 2,000,000 projections per second, then the entire algorithm will run for $100,000 \times 1000 \times 10,000/2,000,000$ seconds which is approximately 6 days. This speed is nowhere close to fulfill the portfolio risk management needs.

There have been much research on developing efficient nested-simulation algorithms for VA valuation. The existing research on single VA policies mainly relies on the least-square Monte Carlo (LSMC) method. This method was first proposed by Longstaff and Schwartz (2001) for pricing American options. The idea is to approximate the continuation value of the American option by regressing the discounted payoffs onto a set of polynomials. The procedure starts from the maturity date and the continuation values are estimated backward through time. As a result of this basis approximation, running the nested inner-loops is avoided. In Bacinello et al. (2011), a framework for valuing VA using the LSMC method was proposed, in which the backward algorithm is applied to the guarantee payoffs over the policy term to calculate the VA liability at the current time. Ha and Bauer (2015) considered stochastic interest rate and volatility models and used the LSMC method to price VA with GMWB riders. Huang and Kwok (2016) incorporated the LSMC method into a stochastic control framework to price and hedge VA with GLWB riders. Hörig et al. (2013) discussed the use of LSMC from a practical prospective. Feng et al. (2016) conducted a study in which the performance of the LSMC approach in the context of VA was discussed. In addition to the valuation of VA, the LSMC approach was also used to calculate various capital requirements in terms of single VA policies. See, for example, Cathcart and Morrison (2009), Bauer et al. (2012), Bauer and Ha (2015), etc.

Despite the broad application of the LSMC approach in the context of VA, two main issues remain. The first issue is the uncertainty of the performance under different circumstances as documented by, for example, Areal et al. (2008). One primary reason for this non-robust performance across different cases is the accumulation of the estimation error throughout the backward procedure. Another issue faced by the LSMC method is the determination of the order of the polynomial basis functions as mentioned in, for example, Feng et al. (2016) and Koursaris (2011). Since the VA guarantee payments are usually multi-dated, path-dependent, and policy-specific, these issues may result in highly inaccurate VA liability estimates. Furthermore, when dealing with large VA portfolios that are made up of a large number of non-homogeneous VA policies, the aforementioned problems may be further amplified. As a result, applying the LSMC approach to the valuation of large VA portfolios is highly challenging.

Research on efficient nested-simulation for large VA portfolios are comparatively more recent. Gan (2013) proposed a clustering technique based on the *k*-means algorithm to select the socalled representative policies. Gan and Lin (2015) proposed a functional data approach where the liability for each policy in the portfolio is assumed as a linear function of the selected representative policies, The coefficients in the linear function are fitted using a universal Kriging approach. Hejazi and Jackson (2016) proposed a neural network framework to estimate the functional dependence between the liability and the policy attributes. Recently Gan and Valdez (2018) proposed a regression approach with a GB2 model to capture the skewness of the liability distribution.

A common issue in the aforementioned research on efficient nested-simulation for VA portfolios is the lack of theoretical justification on the estimator of the quantities of interest for a VA portfolio. In other words, it is not clear whether the existing methods can perform uniformly well over different portfolios. Furthermore, all of the research to date focused mainly on compressing the number of policies. When there are multiple underlying assets being invested by the policyholders, the existing methods might become inefficient again due to the large number of inner and outer-loop projections.

To address these issues, in this paper we propose an efficient nested simulation algorithm in which both the number of policies and the number of inner and outer-loop projections are reduced, as demonstrated in Fig. 1.2. The main contributions of our paper are two-fold. First, we propose the use of a model-assisted population sampling framework to reduce the number of policies. This method is fundamentally different from the clustering method and Latin hypercube method currently used in other papers. Second, we propose to use a spline regression together with scenario clustering to reduce the number of inner and outer-loop projections. Specifically, we assign each policy in the portfolio an inclusion probability and use the Cube sampling algorithm (Deville and Tillé, 2004) to select a set of representative policies; and a k-means clustering approach to select a set of so called representative outer-loops. More importantly, the theoretical justification behind the selection methods are provided. This ensures that the algorithm would work well in general with different VA portfolios and economic scenario generators. Since several statistical models are built into the proposed simulation algorithm, we name it the surrogate model assisted nested-simulation algorithm.

The rest of the paper is organized as the follows. In Section 2 we focus the discussion on a single VA policy and introduce a spline regression approach that enables the nested simulation algorithm to run with fewer numbers of inner-loops and



Fig. 1.2. The flow chart illustration of the proposed nested simulation.

outer-loops. In Section 3 we review the model-assisted finite population estimation framework which is used to reduce the number of policies. We construct a synthetic portfolio with 100,000 VA policies in Section 4 that realistically reflects a real VA portfolio, the predictive distribution of its total liability is computed using both the full simulation algorithm and the proposed algorithm. Results are compared in terms of the estimation accuracy and the running time. We show that the relative errors for various risk metrics of the predictive total VA liability distribution are very small meanwhile the running time is significantly shortened.

2. Fast and efficient nested simulation for single VA policies: A spline regression approach

As mentioned in Introduction, most of the existing methods for efficient nested simulation for single VAs are built upon the LSMC method. To calculate the VA liability using the LSMC method, the accumulated present values of the guarantee payments from all of the projected scenarios are regressed to a set of polynomial basis functions. As a result, for valuation purposes each projected scenario contains two parts where the first part is generated using the real-world economic scenario generator (ESG) and the second part is generated using the risk-neutral ESG, as demonstrated in Bauer et al. (2012). Essentially, under this method the VA liability at the future time of interest along each real-world projection is estimated using only one risk-neutral scenario. Therefore, one needs to run the LSMC algorithm with a large number of scenario projections (outer-loops) in order to obtain an accurate estimate of the VA liability. In addition to the estimation error that might arise from each liability calculation, a large number of outer-loops may still be time consuming to execute, especially when the method is applied to a portfolio of many non-homogeneous policies.

Here, we propose an alternative approach to estimating the VA liabilities in which both the outer and inner-loop simulation are run independently but the outer-loop is run only for a few selected ones. Because the VA liabilities are calculated from multiple inner-loops, they contain less estimation errors. Our proposed method contains two steps: first, a set of representative outer-loops is selected and the VA liabilities at those outer-loops are calculated through multiple inner-loops; second, these VA liabilities are used to estimate the true VA liabilities at all the outer-loops (including both the selected and non-selected ones). The first step is achieved by a scenario clustering approach. The second step is done by fitting a functional relationship between a policy's account values and its liabilities over different outer-loops at the future time of interest. A similar approach without scenario clustering was used in Hörig et al. (2013), in which a polynomial basis was used to fit the functional relationship. Because each polynomial basis function is defined globally, all of the observed data points are fitted using the same set of basis functions. As a result, a smooth curve fitted by the polynomial basis is not flexible enough to capture the dependency over the entire range. To improve upon this issue, in this paper we choose to use a spline regression model with a set of B-spline basis functions (see Section 2.2) to approximate the underlying relationship. Different than the polynomial basis, the B-spline basis are piece-wise polynomials jointed at a sequence of knots. Because of this structure the spline regression model is very flexible to accommodate any functional relationships and it can produce accurate liability estimates at all outer-loops for all types of embedded guarantees. This makes our method particularly efficient at the portfolio level. Furthermore, the close approximation allows for accurate estimation of the derivatives of the fitted curve, which can then be used to estimate the partial dollar Deltas of the VA liabilities with respect to different underlying assets, see Lin and Yang (2020).

2.1. Motivation

As mentioned before, there are two steps in the nestedsimulation algorithm. In the first step a policy's account value is projected from t = 0 to t = 1 using multiple outerloop. Then in the second step multiple inner-loops are simulated at each projected account value from t = 1 to the contract maturity in order to find the liability. The total running time of the algorithm is therefore proportional to the product of the numbers of inner-loops and outer-loops. Furthermore, in practice the VA mangers normally run the nested-simulation with different sets of assumptions in order to study the sensitivity of the total VA liability with respect to different parameters. If every time the nested-simulation is run with large numbers of innerloops and outer-loops, then calculating the liability for a single policy would be time consuming, not to mention doing that for a large VA portfolio. Motivated by this we intend to reduce the computing time by running much fewer numbers of inner-loops and outer-loops. To illustrate we use the following two generic but common VA policies in this section to demonstrate our methodology:

• VA1: Male, age 45, 20 years of maturity, GMDB + GMAB rider where the guaranteed death benefit base rolls up at 3% per year and the accumulation benefit base rolls up at 1% per year.



Fig. 2.1. Predictive liabilities with different numbers of inner-loops.

• VA2: Female, age 65, 15 years of maturity, GMDB + GMWB rider where guaranteed death benefit base is of ratcheting type and annual withdrawal rate equals 1/15.

We run two sets of nested-simulations for the two policies to calculate the predictive distributions of their VA liabilities in one year (t = 1). The two simulation algorithms are 1000/1000 (1000 outer-loops with 1000 inner-loops) algorithm and 1000/10,000 algorithm. Both the inner-loops and the outer-loops are generated by the regime-switching lognormal model (see Hardy (2001)) with different sets of parameters. The details of the simulation are provided in Appendix A. The results under these two settings are shown in Fig. 2.1. Each plot in Fig. 2.1 contains 1000 data points corresponding to 1000 outer-loops. The values on the horizontal axis represent the projected account values, and the values on the vertical axis are the estimated liabilities calculated from different numbers of inner-loops.

Two observations are shown from Fig. 2.1: (i) the dependency between the predicted account value and the predicted liability is not linear and the pattern varies across policies and (ii) the scatterplot becomes smoother with a larger number of innerloops, which is well-expected because running more inner-loops reduces the estimation error. There exists a trade-off between the estimation accuracy and the running time. According to our study, for a single VA policy, the 1000/1000 and 1000/10,000 nested-simulation algorithm takes around 15 s and 150 s to run, respectively. Hence, there is a need to have an efficient algorithm that runs on a small number of inner-loops and still provides accurate values of the liabilities, which can be achieved by a spline regression approach introduced in the next subsection.

2.2. Spline regression: an overview

According to the above numerical illustration, it is reasonable to assume a functional relationship between the predicted account value and the true liability. We propose to use a spline regression model to approximate this relationship. The spline regression was first proposed by Friedman and Stuetzle (1981) as a generalization of the classic linear modeling framework. A spline is a continuous function defined by piecewise polynomials. One advantage of the spline regression modeling is its ability in capturing the dependency between the predictors and the response variables when a linear relationship fails to hold. O'Sullivan (1986) was one of the first works that advocates the use of the splines due to their desired properties. In the VA context the spline model is fitted using the predicted account values and the simulated liabilities. From a practical prospective, the simulated liabilities do not need to be highly accurate in order to fit the spline model which means they can be calculated using fewer number of inner-loops. As a result of this the simulation algorithm can be run at a faster speed.

In the following we briefly review the definition of the B-spline functions (see De Boor (1978)) which are the building blocks of splines. We adapt the notation in Claeskens et al. (2009), and let $\kappa = \{a = \kappa_0 < \kappa_1 < \cdots < \kappa_K < \kappa_{K+1} = b\}$ be a sequence of nondecreasing knots. In addition, define *p* knots at the left boundary $\kappa_{-p} = \kappa_{-p+1} = \cdots = \kappa_{-1} = \kappa_0$ and another *p* knots at the right boundary $\kappa_{K+1} = \kappa_{K+2} = \cdots = \kappa_{K+p+1}$. The B-splines are piecewise polynomials defined recursively on interval [*a*, *b*], with the first-order B-splines being indicator functions:

$$N_{j,1}(x) = \begin{cases} 1 & \text{if } \kappa_j \le x < \kappa_{j+1}, \\ 0 & \text{otherwise,} \end{cases}$$

where $N_{j,1}(x) = 0$ if $\kappa_j = \kappa_{j+1}$. The higher order B-splines are defined as:

$$N_{j,k}(x) = \omega_{j,k}(x)N_{j,k-1}(x) + (1 - \omega_{j+1,k}(x))N_{j+1,k-1}(x),$$

where $j = -p, \dots, K$ and
$$\omega_{j,k}(x) = \begin{cases} \frac{x - \kappa_j}{\kappa_{j+k-1} - \kappa_j} & \text{if } \kappa_j \neq \kappa_{j+k-1}, \\ 0 & \text{otherwise}, \end{cases}$$

with the convention 0/0 = 0. One special property on the derivatives of the spline functions which is useful in later discussions is stated below.

Lemma 2.1 (Derivatives of Splines). Consider a spline function of order p + 1, $\sum_{j=-p}^{K} \beta_j N_{j,p+1}(x)$, then its qth derivative where $q \le p$, $(\sum_{j=-p}^{K} \beta_j N_{j,p+1}(x))^{(q)}$, is another spline function of order p + 1 - q and

$$\left(\sum_{j=-p}^{K}\beta_{j}N_{j,p+1}(x)\right)^{(q)} = \sum_{j=-p+q}^{K}\beta_{j}^{(q)}N_{j,p+1-q}(x),$$

where $\beta_i^{(q)}$ is obtained recursively by

$$eta_{j}^{(q)} = (p+1-q) rac{eta_{j}^{(q-1)} - eta_{j-1}^{(q-1)}}{\kappa_{j+p+1-q} - \kappa_{j}}.$$

It follows from the above identity that the relation between $\beta^{(q)} = (\beta^{(q)}_{-p+q}, \dots, \beta^{(q)}_{K})^t$ and $\beta = (\beta_{-p}, \dots, \beta_{K})^t$ may be written in matrix form $\beta^{(q)} = \Delta_q \beta$ where $\Delta_q = M_q M_{q-1} \cdots M_1$ with M_l given in Box I.

Consider now a univariate spline regression constructed by B-splines with the aforementioned knots sequence κ . Let Y_i denote the response variable and x_i denotes the predictor. The spline regression with (p+1)st order B-spline basis functions can be written as:

$$Y_i = \sum_{j=-p}^{K} \beta_j N_{j,p+1}(x_i) + \epsilon_i, \qquad (2.1)$$

where ϵ_i 's are assumed i.i.d. with mean 0 and variance σ^2 . The parameter β_j 's may be fitted using the *penalized least squares estimation* approach such that the following objective function is minimized:

$$\sum_{i=1}^{n} \left(Y_i - \sum_{j=-p}^{K} \beta_j N_{j,p+1}(x_i) \right)^2 + \lambda \int_a^b \left((\sum_{j=-p}^{K} \beta_j N_{j,p+1}(x))^{(q)} \right)^2 dx.$$
(2.2)

Here, the first term is the objective function of a usual least square optimization and the second term is a penalty term which prevents the fitted curve from over-fitting. The condition $q \le p$ is normally imposed to ensure the existence of the second term. The tuning parameter λ controls the degree of penalization. Under the extreme cases when $\lambda \to \infty$ the fitted function becomes to a (q - 1)th order polynomial whereas $\lambda = 0$ results in an unpenalized estimate which is normally referred to as the *regression spline estimator*. In practice the parameters λ is usually determined by cross validation (CV) or generalized cross validation (GCV).

The solution to (2.2) is called the *penalized spline estimator*. For the ease of notation hereafter the subscription p + 1 will be dropped when we refer to the (p + 1)st order of B-splines. The subscription will be used when referring other orders. Let $N(x) = (N_{-p,p+1}(x), \ldots, N_{K,p+1}(x))$ be a vector containing all B-splines of order p + 1 at x and $N = (N(x_1)^t, N(x_2)^t, \ldots, N(x_n)^t)$ be an $n \times (K + p + 1)$ matrix, and $\beta = (\beta_{-p}, \ldots, \beta_K)^t$ be the column vector of coefficient. The penalized spline estimator can be solved in close-form by rewriting the penalty term as $\lambda \beta^t \Delta_q^t R \Delta_q \beta$ with Δ_q given in Section 2.2 and matrix R contains the inner-products of (p + 1 - q)th order B-spline functions: $R_{ij} = \int_a^b N_{j,p+1-q}(x)N_{i,p+1-q}(x)dx$. With the notation the optimization can be rewritten as

$$\beta^* = \operatorname{argmin}_{\beta} \|Y - N\beta\|^2 + \lambda \beta^t \Delta_q^t R \Delta_q \beta.$$

Let $D_q = \Delta_q^t R \Delta_q$. The penalized spline estimator is given by

$$\hat{f}(x) = N(x)(N^t N + \lambda D_q)^{-1} N^t Y.$$
(2.3)

Hence, the penalized spline estimator is a spline of order (p + 1) with coefficients given by $(N^t N + \lambda D_q)^{-1}N^t Y$. In a special case when $\lambda = 0$, the estimator is called the *regression spline estimator*. It follows from Lemma 2.1 that the *i*th derivative of (2.3) is a (p + 1 - i)th order spline and it can be written as

$$\hat{f}^{(i)}(x) = N_{p+1-i}(x)\Delta_i (N^t N + \lambda D_q)^{-1} N^t Y.$$
(2.4)

We remark that the method for training set selection relies heavily on the asymptotic properties of the above estimators (2.3) and (2.4), which will be provided in the next subsection.

2.3. Training set selection for spline regression models

The objective in this subsection is to reduce the number of outer-loops. Consider a nested-simulation algorithm with M outer-loops. For each policy one observes M predicted account values where each is obtained through a particular outer-loop. At each account value a predicted liability is computed from simulating multiple inner-loops. Due to the use of the spline model, the running time maybe further reduced by simulating a smaller number of inner-loops. However, it may still be timeconsuming to simulate inner-loops for all outer-loops. Hence, we propose an approach such that the predicted liabilities are simulated only at some selected account values, and then use the resulting pairs of the selected account values and liabilities to fit a spline model to estimate the liabilities at all outer-loops. Ideally the model fitted by the selected subset should be close to the model fitted by the entire observations so that the estimated liabilities have higher accuracy.

In the following we introduce a clustering based method to select the predicted account values. For the ease of notation, let $x = (x_1, \ldots, x_M)$ be the observed values of the predictor, which is generated from a uni-variate distribution of *X*, say Q(X). Let $Y = (Y_1, \ldots, Y_M)$ be the corresponding responses. The relationship *x* and *Y* is modeled by a spline model:

$$Y_{i} = f(x_{i}) + \epsilon_{i} = \sum_{j=-p}^{K} \beta_{j} N_{j,p+1}(x_{i}) + \epsilon_{i}, \ i = 1, \dots, M,$$
(2.5)

Box I.



where ϵ_i 's are i.i.d. with zero means. In the VA context, x and Y represent the predictive account values and the predictive liabilities of a policy at different outer-loops, respectively, and M represents the number of outer-loops. In order to further reduce the running time, we select a subset of the predictive account values, denoted by $x^* = (x_1^*, \ldots, x_m^*)$, where $m \ll M$, so that only at those predictive account values will the predictive liabilities be calculated. As a result, the spline model will be fitted by a smaller amount of data. The objective is to select a set of predictive account values such that the fitted model produces close approximations to the predictive liabilities at all predictive account values.

Let $\hat{f}(\cdot)$ denote the fitted model. We define the following objective function to be minimized:

$$\sum_{i=1}^{M} \mathbb{E}\left((\hat{f}(x_i) - f(x_i))^2 \right).$$
(2.6)

In the next theorem we give an upper bound of (2.6) which depends on the selected predictors x^* . This bound will be used for setting the selection criterion.

Theorem 2.1. Let $\{C_1, \ldots, C_m\}$ be a partition of x such that $x_j^* \in C_j$ for $j = 1, \ldots, m$ and each C_j contains M_j elements, $\sum_{j=1}^m M_j = M$. When $p \ge 3$, the following statement holds:

$$\sum_{i=1}^{M} \mathbb{E}\left((\hat{f}(x_i) - f(x_i))^2\right) \le 3\left(M \max_j MSE(\hat{f}(x_j^*)) + \max_j MSE(\hat{f}'(x_j^*))\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2 + \sum_{j=1}^{m} \sum_{i=1}^{M_j} O((x_i - x_j^*)^4)\right).$$

Proof. Without loss of generality, consider an $x_i \in C_j$ such that $x_i \ge x_j^*$. Applying the Taylor's theorem to both $\hat{f}(x_i)$ and $f(x_i)$, one writes:

$$f(x_i) = f(x_j^*) + f'(x_j^*)(x_i - x_j^*) + \frac{f''(\zeta_i)}{2}(x_i - x_j^*)^2,$$

for some $\zeta_i \in (x_j^*, x_i);$
 $\hat{f}(x_i) = \hat{f}(x_j^*) + \hat{f}'(x_j^*)(x_i - x_j^*) + \frac{\hat{f}''(\eta_i)}{2}(x_i - x_j^*)^2,$

for some
$$\eta_i \in (x_i^*, x_i)$$
.

Thus,

$$\sum_{i=1}^{M} \mathbb{E}\left((\hat{f}(x_i) - f(x_i))^2 \right) = \sum_{j=1}^{m} \sum_{i=1}^{M_j} \mathbb{E}\left((\hat{f}(x_i) - f(x_i))^2 \right)$$
$$= \sum_{j=1}^{m} \sum_{i=1}^{M_j} \mathbb{E}\left((\hat{f}(x_j^*) + \hat{f}'(x_j^*)(x_i - x_j^*) + \frac{\hat{f}''(\eta_i)}{2}(x_i - x_j^*)^2 - f(x_j^*) - f'(x_j^*)(x_i - x_j^*) - \frac{f''(\zeta_i)}{2}(x_i - x_j^*)^2 \right)$$

$$\leq 3 \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} \left(\mathbb{E} \left((\hat{f}(x_{j}^{*}) - f(x_{j}^{*}))^{2} \right) + (x_{i} - x_{j}^{*})^{2} \mathbb{E} \left((\hat{f}'(x_{j}^{*}) - f'(x_{j}^{*}))^{2} \right) \right) \\ + \frac{(x_{i} - x_{j}^{*})^{4}}{4} \mathbb{E} \left((\hat{f}''(\eta_{i}) - f''(\zeta_{i}))^{2} \right) \right) \\ = 3 \left(\sum_{j=1}^{m} \sum_{i=1}^{M_{j}} MSE(\hat{f}(x_{j}^{*})) + \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} (x_{i} - x_{j}^{*})^{2} MSE(\hat{f}'(x_{j}^{*})) \right) \\ + \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} O((x_{i} - x_{j}^{*})^{4}) \right) \\ \leq 3 \left(M \max_{j} MSE(\hat{f}(x_{j}^{*})) + \max_{j} MSE(\hat{f}'(x_{j}^{*})) \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} (x_{i} - x_{j}^{*})^{2} \\ + \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} O((x_{i} - x_{j}^{*})^{4}) \right).$$

$$(2.7)$$

Here, the third line follows from $(u_1 + \dots + u_n)^2 \le n(u_1^2 + \dots + u_n^2)$ where $u_j \in \mathbb{R}$ for $j = 1, \dots, n$. The fourth line follows from the fact that both $f''(\cdot)$ and $\hat{f}''(\cdot)$ are bounded from above when $p \ge 3$. \Box

The first and the second term of (2.7) involve the meansquared-errors of the spline estimator and its first derivative of the selected predictor values. The third term of (2.7) is a fourthorder term which converges to zero faster than the second term when the partition becomes finer and finer. Hence, we will ignore the third term and propose a method for selecting x^* that controls the first two terms. In order to develop a proper selection method, it is necessary to study the properties of the mean-squared-error terms. In the next we will derive the asymptotic bias and variance for the derivatives of a penalized spline estimator. The asymptotic properties of the penalized spline estimator have been studied by several authors. Zhou et al. (1998) first studied the asymptotic properties of the splines estimators and derived their confidence interval. In Zhou and Wolfe (2000) the asymptotic properties of the derivatives were studied. More recently Claeskens et al. (2009) extended the results in Zhou et al. (1998) and derived the asymptotic properties of the penalized splines estimators. We rely on their results and derive the asymptotic properties of the derivatives of the penalized spline estimators. We first state several standard and necessary assumptions followed by the main theorem. To be consistent with the existing literature, we will use *n* to denote the number of data points in the following theorem and corollary.

Assumption 2.1. Let δ denote the maximum distance between adjacent knots, i.e. $\delta = \max_{0 \le j \le K} (\kappa_{j+1} - \kappa_j)$, there exists a constant $\Pi > 0$ such that $\delta = O(K^{-1})$ and $\delta/\min_{0 \le j \le K} (\kappa_{j+1} - \kappa_j) \le \Pi$.

Assumption 2.2. Assume the design density Q is continuously differentiable and $\sup_{x \in [a,b]} |Q_n(x) - Q(x)| = o(K^{-1})$ where Q_n is the empirical distribution of x_1, \ldots, x_n .

Assumption 2.3. The number of knots is of smaller order than the total number of design points, i.e. K = o(n).

Theorem 2.2. Define $K_q = (K + p + 1 - q)(\lambda \tilde{c}_1)^{1/2q} n^{-1/2q}$ where $\tilde{c}_1 = c_1(1 + o(1))$ in which c_1 is a constant and depends only on the design density Q (see Lemma A3 in Claeskens et al. (2009)). Assume $f(x) \in C^{p+1}$ and denote $f^{(i)}(x)$ the ith derivative of f(x) for i = 1, 2, ..., q-1. Under Assumptions 2.1–2.3, the following results hold:

If
$$K_q < 1$$
, then

$$\mathbb{E}\left(\hat{f}^{(i)}(x)\right) - f^{(i)}(x) = O(\lambda n^{-1} \delta^{-q-i});$$
Var $\left(\hat{f}^{(i)}(x)\right) = O(n^{-1} \delta^{-2i-1}) + O(\lambda n^{-2} \delta^{-2q-2i-1});$
If $K_q \ge 1$,¹ then

$$\mathbb{E}\left(\hat{f}^{(i)}(x)\right) - f^{(i)}(x) = O((\lambda/n)^{1/2} \delta^{-q-i});$$
Var $\left(\hat{f}^{(i)}(x)\right) = O((\lambda n)^{-1/2} \delta^{-2i-1}) + O((\lambda n)^{-1} \delta^{-2q-2i-1}).$

We delegate the proof to Appendix C. From the above results, the derivatives of a penalized spline estimator have lower convergence rate than the original estimator. This is consistent to the findings in nonparametric regression literature (for example Zhou and Wolfe, 2000). The constant K_q can be thought as an alternative measurement of the number of knots, and the results show that the convergence rates for both bias and variance are faster when the number of knots is relatively small. Theorem 2.2 gives the order of the pointwise expectation and variance of the penalized splines estimator, with those results the following lemma on mean-squared-errors follows immediately.

Corollary 2.1. As a result of Theorem 2.2, the asymptotic mean-squared-errors of $f^{(i)}(x)$ are as follows:

If
$$K_q < 1$$
, then

$$MSE(f^{(i)}(x)) = O(\lambda^2 n^{-2} \delta^{-2q-2i}) + O(n^{-1} \delta^{-2i-1}) + O(\lambda n^{-2} \delta^{-2q-2i-1});$$

If $K_q \geq 1$, then

$$MSE(f^{(i)}(x)) = O((\lambda/n)\delta^{-2q-2i}) + O((\lambda/n)^{-1/2}\delta^{-2i-1}) + O((\lambda n)^{-1}\delta^{-2q-2i-1}).$$

Corollary 2.1 states that asymptotically the mean-squareerrors of the derivatives of the penalized spline estimator are proportional to the number of observations, maximum mesh size of the knots sequence and the smoothing parameter. In other words, they do not dependent on the value of predictor. For a concrete example, consider a case where there are *m* out of *M* data points used to fit a spline regression. Assume that $K_q < 1$ (a sparse knot sequence relative to the observed data), p = 3 (cubic splines), q = 2 (penalizing the second derivative), the number of knots $K = O(m^{1/9})$, see Stone (1982), and the smoothing parameter $\lambda = O(m^{-1})$. Then there exist some constants C_1 and C_2 such that an upper bound of the sum of the squared errors is approximately:

$$3\left(M\max_{j} \text{MSE}(\hat{f}(x_{j}^{*})) + \max_{j} \text{MSE}(\hat{f}'(x_{j}^{*})) \sum_{j=1}^{m} \sum_{i=1}^{M_{j}} (x_{i} - x_{j}^{*})^{2}\right)$$

¹ When $K_q \ge 1$, the assumption $f(x) \in C^{p+1}$ can be relaxed to $f(x) \in W^q[a, b]$, the Sobolev space of order q, i.e. $W^q[a, b] = \{f : f \in C^{q-1}, \int_a^b (f^{(q)}(x))^2 dx < \infty\}$.

$$= C_1 M m^{-8/9} + C_2 m^{-6/9} \sum_{j=1}^m \sum_{i=1}^{M_j} (x_i - x_j^*)^2.$$

It is easy to see, based on the above example, that asymptotically the upper bound can always be written in a form such that x^* only appears in $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$. Based on this we propose a strategy to select x^* such that $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$ is minimized. We remark that our proposed method is not the optimal strategy in the sense of minimizing the objective function. However, our proposed method is a simple approach which can be implemented easily among different situations.

In order to implement the proposed method, we need to find a vector $x^* = (x_1^*, \ldots, x_m^*)$ which minimizes $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$. This is a NP-hard problem which has been documented by, for example, Aloise et al. (2009). The well established *k*-means clustering algorithm can be used to serve for the purpose. The *k*-means clustering algorithm is an unsupervised learning algorithm which was introduced by MacQueen et al. (1967). For a given vector $\mathbf{x} = (x_1, \ldots, x_M)$, the algorithm finds a partition whose objective is minimizing the *within-cluster sum of squares* (WCSS), $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - z_j)^2$ where z_j represents the average of all x_i in the *j*th cluster. It is worth noticing that a cluster center is usually not an element of that cluster. Hence, as an additional step to running the *k*-means algorithm with the predictors we will select the point that is the closest to its cluster center from each cluster.

2.4. Selection of representative outer-loops

In practice running the *k*-means algorithm with the predictive account values for different policies may result in different sets of training data. This is caused by two main issues: the first issue is that the *k*-means algorithm starts at a set of randomly initialized cluster centers and it stops at a local optimum. Hence the results vary from time to time; the second issue is that the clustering depends on the policy's predictive account values which are different among different policies.

Having different training data for different policies is inefficient in terms of computation because it means the nestedsimulation algorithm runs for different sets of outer-loops for different policies. To conquer the above issue we decide to select the training set based on the underlying asset's returns from different outer-loops. We will refer to this set as a set of *representative outer-loops*. In order to better control the variability at the tail parts the maximum and minimum returns are also included in the representative set. By doing this one gets the same training set for all of the polices in the portfolio. Generally speaking the selection based on real-world returns is similar to the selection based on the predicted account values since a higher return normally yields a higher predicted account value.

Fig. 2.2 demonstrates the performance of our proposed method. The left two panels show the selected data points and the fitted spline curves. The scattered crosses are those liabilities computed using only 1000 inner-loop simulations. The red circles are the data points selected by running the *k*-means algorithm with the return vector. We select 200 outer-loops out of 1000 so in the clustering algorithm we set m = 200. The black curve is the fitted spline curve fitted by the selected data. In the numerical study we use ten B-spline basis functions with equidistant knots to fit the spline model.

The predictive VA liabilities obtained from the 1000/10,000 simulation algorithm are used to benchmark the performance of our proposed method. The results are compared in the right panels of Fig. 2.2. Overall the proposed method provides good approximations at all predicted account values including the two





extreme regions. The average absolute relative error of the approximated liabilities is 1.12% for the first VA policy and 0.79% for the second VA policy whereas the running time for each policy is reduced by approximately 50 times.

We remark that there exists a trade-off between the numbers of outer and inner-loops in the reduced simulation and the standard error of the VA liability estimates. This trade-off can also be seen from Theorem 2.2. In practice, the allocation of the computing resources between the numbers of outer and innerloops is normally constrained by a simulation budget. A thorough review of some mathematical frameworks on the allocation of simulation budget can be found in, for example, Feng et al. (2016). Here we do not elaborate further into that direction and simply treat the simulation budget as given for illustration purpose.

3. Selection of representative policies: A finite population estimation approach

To obtain the predictive distribution of the total VA liability at a future time point (t = 1), we need to calculate the total liability at each of the outer-loops. When the portfolio contains a large number of policies (100K+) the calculation can be extremely time consuming even with small numbers of inner and outerloops. In order to estimate the total liability at a significantly less running time we adapt the model-assisted population estimation framework to select a small subset of the policies, which is called *representative policies*, and the total liability is estimated only by the liabilities of the selected representative policies.

3.1. Model-assisted estimation: an overview

We start with several terminologies. Consider a population of size N with population units labeled by $U = \{1, 2, ..., N\}$ and let L_i , i = 1, 2, ..., N denote the value associated with the population units. In the VA context, N is the total number of policies and L_i is the quantity of interest of policy i such as the predictive liability. A random sample S that is sampled without replacement is an element in the power set of U. By assigning a probability to each element in the power set of Uone generates a probability distribution P for all possible samples, called sampling distribution. In this paper we consider only the fixed size sampling design and we use n to denote the sample size. Again, in the VA context the sample size n represents the number of representative policies. A sampling distribution can be uniquely determined by assigning an *inclusion probability* to each population unit. Define $\mathbb{1}_k$, the inclusion indicator random variable, that indicates the inclusion of the unit k in a random sample S. The *first-order inclusion probability* of unit k is defined as $\pi_k = \mathbb{E}_P(\mathbb{1}_k) = P(k \in S)$. For a fixed size sampling design the constrain $\sum_{k=1}^N \pi_k = \sum_{k=1}^N \mathbb{E}_P(\mathbb{1}_k) = n$ must be satisfied by all inclusion probabilities.

A typical objective in finite population estimation is estimating the population total $T = \sum_{k=1}^{N} L_k$, which is the total VA liability, by a linear estimator $\hat{T} = \sum_{k\in S} \omega_{kS} L_k$ where ω_{kS} represents the weight given to the sampled unit k in sample S. In order to find the estimate of the population total one needs to specify the weights and a sampling design, both of which are normally derived from a set of criteria given by the sampler. One of the most commonly imposed criterion is design unbiasedness such that $\mathbb{E}_P(\hat{T}) = T$. The design unbiasedness criterion implies a relationship between the weight ω_{kS} and the inclusion probability π_k which is given by the following theorem.

Theorem 3.1 (*Godambe*, 1955). Denote $C_k = \mathbb{E}_P(\omega_{kS} \mathbb{1}_k) = \pi_k \mathbb{E}_P(\omega_{kS} | \mathbb{1}_k = 1)$. The estimator $\hat{T} = \sum_{k \in S} \omega_{kS} L_k$ is design-unbiased to $T = \sum_{k=1}^N L_k$ if and only if $C_k = 1$.

Hence, under the design unbiasedness criterion the expected weights for the population units are the reciprocal of their respective inclusion probabilities, i.e. $\mathbb{E}_P(\omega_{kS}|\mathbb{1}_k = 1) = 1/\pi_k$, or $\pi_k = 1/\mathbb{E}_P(\omega_{kS}|\mathbb{1}_k = 1)$, k = 1, ..., N.

The above classic framework focuses on the properties of a population estimator with respect to the sampling distribution. The *model-assisted population estimation* framework (see, for example, Särndal et al. (2003)) extends the classic framework by utilizing the auxiliary information of the population units. Under this framework a model is assumed between the value associated with the population unit (response variable) and their auxiliary information (predictors). This assumed model is referred to as the *superpopulation model*. In most cases the superpopulation model takes a linear form. Let L_k denote the response variable and \mathbf{x}_k denote the predictors. The superpopulation model is:

$$\xi: L_k = \mathbf{x}'_k \mathbf{b} + e_k, \, k = 1, \dots, N,$$

where the residuals e_k 's are assumed independently distributed with zero means and standard deviation σ_k . In the VA context the predictors \mathbf{x}_k may represent the attribute variables of the *k*th policy such as age, gender, term of maturity, guarantee type, etc., and the linear superpopulation model can be viewed as a first-order approximation to the relation between the quantity of interest and the attribute variables. Different from the classic framework, the model-assisted estimation aims at a valid inference under both the sampling distribution and the superpopulation model. A commonly used measurement for the validity of an estimator under the model-assisted framework is the *anticipated meansquared error* (Isaki and Fuller, 1982). We use \mathbb{E}_{ξ} and var $_{\xi}$ to denote the mean and variance with respect to the superpopulation model, the anticipated mean-squared error is defined as follows.

Definition 3.1. The anticipated mean-squared error (anticipated MSE) of an estimator \hat{T} is $\mathbb{E}_{P}\mathbb{E}_{\xi}(\hat{T} - T)^{2}$.

The anticipated MSE takes into account the estimation errors from both the random sampling and the model mis-specification. Nedyalkova and Tillé (2008) derived an expression of the anticipated MSE that contains the constant C_k defined in Theorem 3.1.

Theorem 3.2 (Result 1 of Nedyalkova and Tillé (2008)). Under the superpopulation linear model (3.3) the anticipated mean-squared error of the linear estimator $\hat{T} = \sum_{k \in S} \omega_{kS} L_k$ can be expressed as

$$\mathbb{E}_{P}\mathbb{E}_{\xi}(\hat{T}-T)^{2} = \sum_{k\in U} \sigma_{k}^{2}C_{k}^{2}\frac{1-\pi_{k}}{\pi_{k}} + \sum_{k\in U} \sigma_{k}^{2}\pi_{k}var_{P}(\omega_{kS}|\mathbb{1}_{k}=1)$$

$$+\sum_{k\in U} \sigma_k^2 (C_k - 1)^2 + var_P \left(\sum_{k\in S} w_{kS} \mathbf{x}'_k \mathbf{b} \right) + \left(\sum_{k\in U} C_k \mathbf{x}'_k \mathbf{b} - \sum_{k\in U} \mathbf{x}'_k \mathbf{b} \right)^2.$$
(3.1)

All the terms in (3.1) are non-negative. Theorem 3.1 states that, if \hat{T} is design-unbiased, then $C_k = 1$ and $\mathbb{E}_P(\omega_{kS}|\mathbb{1}_k = 1) = 1/\pi_k$ for all $k \in U$. By setting $\omega_{kS} = 1/\pi_k$ for all samples containing the *k*th population unit, one has $\operatorname{var}_P(\omega_{kS}|\mathbb{1}_k = 1) = 0$. The anticipated MSE (3.1) now becomes:

$$\mathbb{E}_P \mathbb{E}_{\xi} (\hat{T} - T)^2 = \sum_{k \in U} \sigma_k^2 \frac{1 - \pi_k}{\pi_k} + \operatorname{var}_P \left(\sum_{k \in S} \mathbf{x}'_k \mathbf{b} / \pi_k \right).$$

The term $\operatorname{var}_{P}\left(\sum_{k\in S} \boldsymbol{x}'_{k}\boldsymbol{b}/\pi_{k}\right)$ is zero if $\sum_{k\in S} \boldsymbol{x}'_{k}\boldsymbol{b}/\pi_{k}$ is independent of the random sampling. This can be achieved by requiring the random sample to be balanced, as defined below.

Definition 3.2 (*Deville and Tillé, 2004*). A sample is said to be balanced on predictors if the Horvitz–Thompson estimator of the predictors satisfies

$$\sum_{k\in S}\frac{\mathbf{x}_k}{\pi_k}=\sum_{k\in U}\mathbf{x}_k.$$

Clearly if a sample is balanced then $\operatorname{var}_P\left(\sum_{k\in S} \mathbf{x}'_k \mathbf{b}/\pi_k\right) = \operatorname{var}_P\left(\sum_{k\in U} \mathbf{x}'_k \mathbf{b}\right) = 0$. As a result, the anticipated MSE is further reduced to:

$$\mathbb{E}_P \mathbb{E}_{\xi} (\hat{T} - T)^2 = \sum_{k \in U} \sigma_k^2 \frac{1 - \pi_k}{\pi_k}.$$

Minimizing the above expression implies choosing $\pi_k = \min(1, \alpha \sigma_k/N)$ for some normalizing constant α . For fixed size sampling design we have $\pi_k = n\sigma_k / \sum_{k \in U} \sigma_k$ since $\sum_{k \in U} \pi_k = n$.

3.2. Balanced sampling: the cube method

The cube sampling method proposed by Deville and Tillé (2004) is an algorithm that uses a first-order inclusion probability to obtain a sample that is nearly balanced. The cube sampling algorithm has two phases: flight phase and landing phase. Assume a population with N units and each unit is equipped with r attributes. The flight phase translates the given inclusion probabilities to a vector of at least (N - r) zeros or ones. The landing phase then converts the non-zero/one units to either zero or one giving an approximately balanced random sample. The balanced condition $\sum_{k \in S} \mathbf{x}_k / \pi_k = \sum_{k \in U} \mathbf{x}_k$ can be written in the matrix form:

$$AS = A\pi, \tag{3.2}$$

where $A = (\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_N)$ a $r \times N$ matrix and $\mathbf{a}_k = \mathbf{x}_k/\pi_k$ for $k = 1, \dots, N$; $\mathbf{S} = (\mathbb{1}_1, \mathbb{1}_2, \dots, \mathbb{1}_N)^t$ a random column vector where $\mathbb{1}_k = 1$ or 0 indicating the inclusion of the *k*th unit. Eq. (3.2) implies that all balanced samples form a subspace of \mathbb{R}^N with dimension N - r. Hence \mathbf{S} can be written as $\pi + \mathbf{u}$ where \mathbf{u} is in the kernel of matrix A, i.e. $A\mathbf{u} = \mathbf{0}$. The flight phase uses this fact and moves the given inclusion probabilities π randomly inside the kernel space of A until it reaches to a point that is close to a vertex of the N dimensional hypercube.

There are three steps in each iteration in the flight phase. For a given inclusion probability vector π , set $\pi(1) = \pi$, at iteration i = 1, 2, ..., l:

.

- Step 1: Randomly generate a vector $\boldsymbol{u}(i)$ in the kernel of matrix $A = (\boldsymbol{a}_1, \boldsymbol{a}_2, \dots, \boldsymbol{a}_N)$. Set $\boldsymbol{u}_k(i) = 0$ if $\pi_k(i) = 0$ or 1.
- Step 2: Compute $\lambda_1^*(i)$ and $\lambda_2^*(i)$, the largest values among $\lambda_1(i)$ and $\lambda_2(i)$ such that:

 $0 \le \boldsymbol{\pi}(i) + \lambda_1(i)\boldsymbol{u}(i) \le 1;$ $0 < \boldsymbol{\pi}(i) - \lambda_2(i)\boldsymbol{u}(i) < 1.$

• Step 3: Compute $\pi(i + 1)$ as follows:

$$\pi(i+1) = \pi(i) + \lambda_1^*(i)\boldsymbol{u}(i) \text{ with probability } \frac{\lambda_2^*(i)}{\lambda_1^*(i) + \lambda_2^*(i)};$$

$$\pi(i+1) = \pi(i) - \lambda_2^*(i)\boldsymbol{u}(i) \text{ with probability } \frac{\lambda_1^*(i)}{\lambda_1^*(i) + \lambda_2^*(i)}.$$

The above three steps iterate until $\pi(i)$ stops changing. Deville and Tillé (2004) show the above algorithm runs in the order of $O(N \times r^2)$. Hence it is very efficient given that r is not too large. They also show that at the end of the flight phase the number of non-integer elements in $\pi(I)$ is less than the number of balancing variables r. In the landing phase, the non-integer elements resulted from the flight phase are converted into either zero or one by linear programming. The resulting vector with only zeros and ones gives a balanced sample.

3.3. Selection of representative policies

We will use the cube sampling algorithm in Section 3.2 to select a set of representative policies. We assume a linear surrogate model as a first-order approximation between the policies' liabilities and their attributes at each outer-loop s:

$$L_{k,1}(s) = \mathbf{x}'_{k,0}\mathbf{b}(s) + e_k(s), \tag{3.3}$$

where $L_{k,1}(s)$ is the predicted liability estimated from the spline model of policy k at the sth outer-loop simulation where k = 1, 2, ..., N and s = 1, 2, ..., M, the vector $\mathbf{x}_{k,0}$ is the attribute vector of policy k, i.e. age, gender, guarantee type, term of maturity, account value, etc., at the valuation time t = 0. The dimension of this vector depends on the complexity of the products that an insurance company offers. The term $e_k(s)$ represents the discrepancy between the first-order approximation and the true liability and they are assumed to be independently distributed with $\mathbb{E}_{\xi}(e_k(s)) = 0$ and $\operatorname{var}_{\xi}(e_k(s)) = \sigma_k(s)^2$.

Model (3.3) is scenario specific as for each population unit its error standard deviation $\sigma_k(s)$ varies over different outerloops. In order to achieve the highest efficiency we look for a selection procedure that is scenario independent, hence a same set of representative policies can be used to estimate the total VA liability across all outer-loops. We propose a two-stage sampling procedure in order to get a single set of representative policies for all selected outer-loops. The main objective of the first stage is to fit a deterministic function to the error standard deviation $\sigma_k(s) =$ $h_s(x_{k,0}^*)$ where $x_{k,0}^*$ is some predictor variable that accounts the most for the heteroscedasticity of the residual variation. This fitting procedure is done through residual diagnostic in which the residuals from the fitted linear model are plotted against different predictors. Then set $\pi_k(s) = nh_s(x_{k,0}^*) / \sum_{k \in U} h_s(x_{k,0}^*)$ in the second stage to select a set of representative policies. Furthermore in order to achieve the scenario independence we assume that function $h_s(\cdot)$ takes a separable form

$$h_s(x_{k,0}^*) = \gamma(s)g(x_{k,0}^*), \tag{3.4}$$

so that the scenario dependency of the error standard deviation is captured by $\gamma(s)$. As a result, the inclusion probabilities are

$$\pi_k(s) = n\gamma(s)g(x_{k,0}^*) / \sum_{k \in U} \gamma(s)g(x_{k,0}^*) = ng(x_{k,0}^*) / \sum_{k \in U} g(x_{k,0}^*),$$

k = 1,..., N,

which depends only on the information at time t = 0. In principle the function $h_s(x_{k,0}^*)$ should be fitted by running the nested simulation for the entire portfolio. In practice, however, it is extremely time consuming yet unnecessary to do so. One may use a sample of the population to diagnose the residuals as the sample has similar distributional properties to the population for all predictors. We use a balanced sampling algorithm with $\pi_k =$ n_1/N to select a sample, calling this the first stage sample, where n_1 is the budgeted first stage sample size which can be different from the budgeted number of representative policies. Using equal inclusion probabilities the resulting sampling distribution of each predictor variable would mimic its corresponding distribution for the population (see Tillé (2006)). This implies that the residual distribution obtained from such a balanced sample would be similar to those generated by the population. In the second stage another balanced sampling is run using the inclusion probabilities given by the first stage results. The resulting set of policies is a set of representative policies whose liabilities are used to estimate the total liability of the VA portfolio across all outer-loops.

We remark that if the heteroscedasticity of residuals is significantly related to more than one predictor variables then one can either divide the entire dataset into subgroups based on the levels of certain predictors or follows the idea discussed above to fit a function that is a product of two parts: a part that captures the scenario dependency and another part that involves only the predictor variables. This will lead to a set of inclusion probabilities that are scenario independent.

4. A simulation study

In this section we illustrate the performance of the proposed algorithm using a synthetic VA portfolio that realistically reflects a real VA portfolio. We calculate the predictive distribution of the total VA liability in one year (t = 1) using both the full nested-simulation and the proposed algorithm which we call the fast nested-simulation algorithm. We also test the proposed method with another synthetic VA portfolio given in Appendix D which is similar to the portfolio used in other papers of this topic, for example, Gan and Lin (2015) and Hejazi and Jackson (2016).

4.1. Synthetic VA portfolio

We construct a synthetic VA portfolio containing 100,000 policies where the guarantee(s) of each contract fall in one of the three categories: GMDB only, GMDB+GMWB and GMDB+GMAB. Most existing research assumes uniform distribution for the attribute variables. However this assumption is hardly reflected in reality especially for variables such as account value and guarantee type. According to the SOA and LIMRA 2015 Variable Annuity Guaranteed Living Benefits Utilization Study on GMWB and GMAB policies over 13 major companies, the election rates of the GMWB and GMAB riders are different over different age groups. For the GMWB rider the election rates are higher for senior groups while those for the GMAB riders are higher among younger age groups. The total election rate for the GMWB rider is around 60% for people aged between 61 and 80. On the other hand for the GMAB rider almost half of the policies with this rider are between age 45 to 60. In terms of the account value distribution, based on the same study, only 10% of the contracts value at \$250,000 or more, 40% are in between 10,000 to 50,000 and the remaining 50% lies in 50,000 to 250,000. We use the above information to construct a synthetic VA portfolio as follows.



Fig. 4.1. Adjusted residuals and inclusion probabilities.

Table 4.1							
Synthetic VA	portfolio	attribute	variables	and	their	distributions.	

Attribute	Value	Distribution
Gender	Male, Female	Uniform
Policyholder's age	45 to 85	Uniform
Maturity	10 to 25 years	
Guarantee type	GMDB	GMWB: 15% among 45-60
	GMDB+GMAB	30% among 61–70
	GMDB+GMWB	30% among 71–80
		20% among 81–85
		GMAB: 50% among 45–60
		30% among 61–70
		15% among 71–80
		5% among 81–85
Annual withdrawal rate	1/Maturity	-
Account value	\$10,000, 20,000,, 500,000	40% between 10,000 and 50,000
		50% between 50,000 and 250,000
		10% above 250,000
Withdrawal benefit base	Initial account value	-
Death benefit base	Initial account value	-
Death benefit guarantee design	Ratcheting or roll-up	Uniform
	Roll-up rate 1%–5%	
Accumulation benefit	Initial account value	-
Accumulation benefit guarantee design	Ratcheting or roll-up	
	Roll-up rate 1%–5%	Uniform
Mortality table	1996 IAM	-

4.2. Representative policies

In Section 3 we had presented a two-stage procedure to select a set of representative policies. In the first stage a balanced sampling with $\pi_k = n/N$ is used to get a set of policies for residual diagnostic; in the second step the inclusion probabilities are set according to the identified residual variances.

We select 1500 policies in both residual diagnostic stage and the selection stage. Note that the number of policies we use is more than those in many of the existing studies, such as Gan and Lin (2015). The reason that a larger number of policies can be afforded is that in our approach the liabilities of the nonselected policies are not estimated. Instead, the total liability is estimated directly through the liabilities of the selected policies. Therefore our approach does not require solving a system of linear equations or other methods to find the relations between the non-selected policies and the selected ones.

We used a balanced sampling algorithm with $\pi_k = 1500/100,000$ in the first stage to select a set of policies to run the fast simulation algorithm and then use their liabilities and covariates to perform a series of residual diagnostics for

various predictor variables. We find that the initial account value accounts the most for the heteroscedasticity among the residuals. So we fit a function for the error standard deviation with account value as the predictor, i.e. $\sigma_k(s) = \gamma(s)g(AV_{k,0})$. Further we assume the function $g(AV_{k,0})$ taking a power form with a constant shift $g(AV_{k,0}) = |AV_{k,0} - \alpha|^{\beta}$. We find that after standardizing the residuals with by the initial account values to the power of 1/10, i.e. $g(AV_{k,0}) = AV_{k,0}^{1/10}$, the residuals variation becomes significantly more homogeneous, which implies a choice of α and β in this case are 0 and 1/10 respectively. Resulted from the first stage study, the inclusion probabilities π_k where k = 1, 2, ..., N used for the second stage sampling are:

$$\pi_{k}(s) = n\sigma_{k}(s) / \sum_{k \in U} \sigma_{k}(s) = n\gamma(s) A V_{k,0}^{\frac{1}{10}} / \sum_{k \in U} \gamma(s) A V_{k,0}^{\frac{1}{10}}$$
$$= n A V_{k,0}^{\frac{1}{10}} / \sum_{k \in U} A V_{k,0}^{\frac{1}{10}}.$$
(4.1)

The left panel in Fig. 4.1 compares the distribution of the unadjusted residuals and the adjusted residuals over different initial account values under a generic outer-loop simulation. The



Fig. 4.2. Distributions of predicted total liability.





overall pattern of the unadjusted and the adjusted residuals are similar among other outer-loops, which validates the separability assumption (3.4). The right panel shows the inclusion probability as a function of the initial account value. Policies with larger initial account values are more likely to be selected into the representative set than those with smaller initial account values. This choice also has a practical implication: policies with larger account values have larger liabilities therefore the approximation accuracy of their liabilities will have material impacts on the total accuracy. Therefore those policies should have higher probabilities to be included in the representative policy set.

4.3. Numerical results

We run both the full nested-simulation and the fast nestedsimulation algorithm with R in parallel using the doSNOW package. The full simulation algorithm is run with 100,000 policies, 1000 outer-loops and 10,000 inner-loops whereas the fast simulation algorithm is run with 1500 selected policies, 200 selected outer-loops and 1000 inner-loops. The full simulation is run using 60 CPU-cores and the proposed fast simulation is run using 4 CPU-cores. We compare these two algorithms from two prospectives: approximation performance and running time. The two predicted total liability distributions are compared in Fig. 4.2.

The overall distribution is well approximated especially for the right tail part which usually is used to calculate risk metrics of the entire distribution. The distribution of the percentage errors of the approximated total liabilities is shown in the left panel in Fig. 4.3. From the numerical study we found the mean relative percentage error is around 0.73%. Most of the percentage errors are distributed in the range of -1% to 1%. The right panel in Fig. 4.3 is the Q–Q plot between the approximated total liability distribution and the true distribution. The two sets of quantiles fall about a straight line implying a good approximation performance.

In the following we compare the two predictive distributions quantitatively by calculating various statistics and their relative errors:

$$100\% imes \left| \frac{\hat{\theta} - \theta}{\theta} \right|,$$

where $\hat{\theta}$ is a statistic such as those in Table 4.2 of the approximated distribution and θ is that of the true distribution. In Table 4.2 several statistics of the predicted liability distributions obtained from the two simulation algorithms are presented. Also reported are the absolute relative errors for those statistics. In terms of the distributional properties we compute different central moments for the two predictive distribution including standard deviation, skewness and kurtosis. Comparing the central moments, the approximated predictive distribution is close to the true distribution. Both distributions have positive skewness and large kurtosis which implies a heavy right tail with a relative large

Table 4.2

Comparison of different statistics.

	Full nested-simulation	Fast nested-simulation	Relative error
Mean	1,435,261,959	1,431,534,562	0.26%
Standard deviation	202,322,678	201,915,195	0.20%
Skewness	2.8614	2.8645	0.11%
Kurtosis	16.6721	17.0674	2.37%
VaR(90)	1,676,803,005	1,673,505,367	0.20%
CVaR(90)	1,905,358,443	1,895,366,512	0.52%
VaR(95)	1,825,330,894	1,803,471,737	1.20%
CVaR(95)	2,071,569,424	2,059,810,942	0.57%
VaR(99)	2,139,601,926	2,116,332,137	1.09%
CVaR(99)	2,519,785,055	2,520,134,600	0.01%

Table 4.3

Comparison of running time.

	Full nested-simulation	Fast nested-simulation
Number of policies	100.000	1500
Number of outer-loops	100,000	200
Number of inner-loops	10,000	1000
Number of CPU cores	60	4
Scenario selection	-	0.008(s)
First stage sampling	-	0.5(s)
First stage nested-simulation	-	1083(s)
First stage model fitting	-	18(s)
Second stage sampling	-	0.5(s)
Second stage nested-simulation	-	1083(s)
Second stage model fitting	-	18(s)
Full simulation	6.02(d)	-
Total running time	6.02(d)	\approx 37(m)

number of outliers. In addition we calculate two most widely used tail risk metrics: value at risk (VaR) and conditional VaR (CVaR) at three security levels: 90%, 95% and 99%. From the results most of the approximation errors are within 1%.

Lastly we compare the running time of these two algorithms. The full algorithms is a brute-force algorithm whereas the fast algorithm contains multiple steps. The running time for each step is reported in Table 4.3 and the total running time is compared to that of the full algorithm. Even with 60 CPU-cores the full algorithm still takes around 6 days to run, while the running time of the fast simulation algorithm only takes about 30 min with a computer containing 4 CPU-cores, the total running time is therefore reduced by more than 230 times. It is worth to remark that in practice heavy computing programs such as nested-simulation are usually done by parallel using both CPU and GPU such as the NVIDIA-CUDA programming model.² Nonetheless the proposed fast simulation algorithm is still expected to be much more efficient than the classic algorithm regardless of the computing platforms as the fast algorithm runs with much fewer numbers of all input variables.

5. Concluding remarks

For both regulatory and hedging purposes the variable annuity portfolio managers are often required to compute the predictive distribution of the total VA liability on a timely basis. The heterogeneity and path-dependency of the VA guarantees cause the traditional nested-simulation extremely time-consuming to run. To address this practically important issue we propose a surrogate model assisted nested-simulation algorithm which incorporates several statistical tools to speed up the computation from running much fewer numbers of different input variables while maintaining high accuracy in approximating the predictive total liability distribution.

Two major classes of surrogate models are considered: a spline regression model is used to select a set of representative outerloops and a linear model is adapted along with the probability sampling framework to select a set of representative policies. Different objectives associated with the two models lead to different selection criteria. When reducing the number of outer-loops, the objective is good approximations to the liabilities at all outerloops. This requires the selected outer-loops to spread out the entire range of the simulated returns. When reducing the number of policies, the main objective is a close estimation of the population total. Based on this, policies whose liabilities are less likely to be correctly estimated are more likely to be included in the representative policies set.

The proposed simulation algorithm is data-driven since the selection of the representative outer-loops and representative policies depend only on the given data. The theoretical justifications of the selection procedures ensure a robust performance of the algorithm for a wide range of VA portfolios. We experiment the algorithm with two generic VA portfolios having different distributions of the attribute variables (Section 4 and Appendix D). The numerical results show that predictive distributions can be closely approximated by the proposed algorithm at a significantly reduced running time. We expect the proposed method also works well for other nonhomogeneous insurance portfolios and different scenario generators since the proposed framework does not assume any specific model for the dynamics of the underlying asset.

This paper may lead to several further research directions. For example one may develop an efficient algorithm that does multiperiod prediction, or one can investigate situations where multiple underlying assets are involved. In these cases a multidimensional clustering technique might be considered. Moreover, one may apply the proposed algorithm to the valuation of other insurance portfolios such as universal life insurance (UL), variable universal life insurance (VUL) portfolios. In terms of application, one may consider applying the proposed framework to calculate other quantities for large VA portfolios such as Greeks. Last but not the least, an ultimate goal for developing an efficient nestedsimulation algorithm is to use it to dynamically hedge the market risk of large VA portfolios. Having an efficient dynamic hedging program is critical in risk managing the liability risk of large VA portfolios, yet it is a challenging task due to the complexity of the products and the non-homogeneity of the VA portfolios. The issue of hedging has been addressed by many studies, and most of them focused on single policies, for example Feng et al. (2016). The question of hedging large VA portfolios will be addressed in Lin and Yang (2020), in which a dynamic hedging program is implemented with the proposed algorithm.

Acknowledgments

This research is partly supported by a grant from the Natural Sciences and Engineering Research Council of Canada [RGPIN-2017-06684]. Shuai Yang acknowledges the support from the Society of Actuaries James C. Hickman Scholarship. We would like to thank the anonymous reviewer for the constructive and thoughtful comments.

Appendix A. Nested-simulation

In this section we describe the algorithm for computing the total variable annuity (VA) liabilities of the entire portfolio. The algorithm and notation follow closely from those in Gan and Lin (2015). The following table summarizes all of the relevant notation.

² https://developer.nvidia.com/cuda-zone.

Notation	Description
St	Value of the underlying asset at time t
A_t	Account value at time t
Et	Withdraw amount at time t
Wt	Withdrawal benefit at time t
D_t	Death benefit at time t
M_T	Accumulation benefit at time t
G_t^W	The guaranteed withdrawal benefit base at time t
G_t^D	The guaranteed death benefit base at time t
G_t^A	The guaranteed accumulation benefit base at time t
ρ^W	The roll-up rate of the withdraw guarantee base
ρ^A	The roll-up rate of the accumulation guarantee base
ρ^{D}	The roll-up rate of the death guarantee base
w	Withdrawal rate
Т	Term of maturity

At a specific time point t we use superscripts - and + to indicate the values of a quantity prior to and after the benefit payments are made at that time. For example, A_t^- is the account value at time t just before any benefit payments are paid and A_t^+ equals A_t^- less all the payments made at time t such as withdrawals.

A.1. Economic scenario generators

We assume a regime-switching lognormal model for the realworld dynamics of the underlying asset and use the following to generate outer-loops.

$$R_0 = 1; \ R_t = R_{t-\frac{1}{12}} \exp\left(\mu_k - \frac{1}{2}\nu_k^2 + \nu_k Z_t^{\mathbb{P}}\right)$$

given the regime at $t - \frac{1}{12}$ is k ,

where $Z_t^{\mathbb{P}}$, $t = \frac{1}{12}, \ldots, 1$ is a sequence of i.i.d. standard normal random variables. The value of μ_k , ν_k depends on the regime at time $t - \frac{1}{12}$. We assume there are two regimes k = 1, 2 and the transition probabilities are $p_{12} = 0.0398$ and $p_{21} = 0.3798$. We use the parameters given by Hardy (2001) where $\mu_1 = 0.0126$, $\mu_2 = -0.0185$, $\nu_1 = 0.0350$ and $\nu_2 = 0.0748$. As mentioned in Hardy (2001) the parameters are estimated using monthly data, therefore in the simulation studies we first simulate monthly returns then convert them into annual returns.

At t = 1 along each outer-loop, multiple inner-loops are simulated using a risk-neutral model. For consistency purposes the risk-neutral dynamics of the underlying asset is also assumed to follow a regime-switching lognormal model. Unlike the Black–Scholes framework, the market is incomplete under the regime-switching model hence the equivalent risk-neutral measure is not unique. Therefore the choice of the risk-neutral parameters is not unique. Bollen (1998) proposed a method to set the values for the risk-neutral parameters by replacing the drifts with the risk-free rate and using the same transition probabilities and volatilities as in the real-world model so that the 'regime risk is not priced'. Based on this we use the following model to generate inner-loops.

$$S_{1} = 1; \ S_{t+1} = S_{t+1-\frac{1}{12}} \exp\left(\frac{r}{12} - \frac{1}{2}\nu_{k}^{2} + \nu_{k}Z_{t}^{\mathbb{Q}}\right)$$
given the regime at $t + 1 - \frac{1}{12}$ is k ,

where $Z_t^{\mathbb{Q}}$, $t = \frac{1}{12}$, ... is a sequence of i.i.d. standard normal random variables. We assume the effective interest rate is r = 3% per year and all other values being the same as those in the \mathbb{P} -model.

A.2. VA liability calculation

In the first step the account values are projected under all realworld scenarios from t = 0 to t = 1 for all policies. Denote $G_0^{W/D/A}$ the guarantee bases at time t = 0 and $G^E = wA_0$ the guaranteed annual withdrawal amount ; $\rho^{D/A}$ the roll-up rate of the death and accumulation benefits guarantee bases; w the withdraw rate, then the withdraw guarantee base at time t = 1 before payments are made is $G_1^{D/A-} = (1+\rho^{D/A})G_0^{D/A}$. Note that if the guarantee is of ratcheting-type then its corresponding $\rho = 0$. The account value at time t = 1 after withdrawals equals

$$A_1^+ = \max\left(A_0 \frac{R_1}{R_0} - wA_0, 0\right)$$
(A.1)

The guarantee withdrawal base, i.e. the remaining amount can be withdrew, is reduced by the withdraw amount

$$G_1^{W+} = G_0^W - wA_0$$

After the withdraw benefit payments are made the death and accumulation guarantee bases will be reduced by the withdraw amounts which is fixed throughout the term since we assume the policyholder takes out the maximum available amount at the end of each year, i.e.

$$G_1^{D/A+} = G_1^{D/A-} - wA_0$$

If the guarantee is of ratcheting type, then the guarantee base will be adjusted after the withdrawals are made, i.e.

$$G_1^{D/A+} = \max(G_1^{D/A-} - wA_0, A_1^+)$$

The second step, after account values are projected to t = 1, is to calculate the fair value of the embedded guarantees. The liabilities are computed from running inner loops which correspond to the risk-neutral simulations. We outline how cash flow evolves under a single risk-neutral path in a general setting during time interval $[t^+, t+1^+]$. At time $t+1^-$, similar to the above procedure, we first adjust all the guarantee bases according to the contract design.

$$G_{t+1}^{W-} = G_t^{W+}$$

$$G_{t+1}^{D/A-} = G_t^{D/A+} (1 + \rho^{D/A})$$

Again for ratcheting type guarantee bases, set $\rho = 0$. From t^+ to $t + 1^-$ the account value becomes

$$A_{t+1}^{-} = A_t^{+} \frac{S_{t+1}}{S_t}$$

Notice that different than (A.1) the account values under the inner-loops evolve according to the risk-neutral model. At time t + 1 the policyholder withdraws $E_{t+1} = \min(G_{t+1}^{W-}, G^E)$ from the account which brings the account value to

$$A_{t+1}^+ = \max \left(A_{t+1}^- - E_{t+1}, 0 \right)$$

The withdraw benefit can be expressed as the payoff function of a put option with strike price E_{t+1}

$$W_{t+1} = \max(E_{t+1} - A_{t+1}^{-}, 0)$$

The guarantee bases are then adjusted by the withdrawal amount.

$$G_{t+1}^{W/D/A+} = G_{t+1}^{W/D/A-} - E_{t+1}.$$

If death happens during the period then only the death benefit payment is made at the end of the period and the benefit equals

$$D_{t+1} = \max \left(G_{t+1}^{D-} - A_{t+1}^{-}, 0 \right).$$

After all payments made at t + 1 if the policy are still enforced and if the benefit bases are of ratcheting type then

$$G_{t+1}^{D/A+} = \max(G_{t+1}^{D/A-} - E_{t+1}, A_{t+1}^+).$$

At maturity for those contracts with GMAB rider the benefit at maturity is

$$M_T = \max\left(G_T^{A+} - A_T^+, 0\right)$$

At the end the liability of the guarantee rider for policyholder p in the kth real-world scenario is the present value of all the projected withdraw, death and maturity benefits

$$L_{t} = \sum_{s=t+1}^{T} \sum_{s=t-1}^{s-t-1} p_{x+t} (1 - q_{x+s-1}) W_{s} e^{-r(s-t)} + \sum_{s=t+1}^{T} \sum_{s=t-1}^{s-t-1} p_{x+t} q_{x+s-1} D_{s} e^{-r(s-t)} + M_{T} e^{-r(T-t)}$$

Appendix B. Pseudocode of the proposed algorithm

Algorithm 1 Selection of Representative Outer-loops

Input: Vector of real-world asset returns $\mathbf{R} = \{R_1, ..., R_M\}$. Number of representative outer-loops m. **Initialize:** Vector $\mathbf{r} = (\max \mathbf{R}, \min \mathbf{R})$. Run k-means algorithm with k = m with \mathbf{R} , record cluster centers $\mathbf{c} = \{c_1, ..., c_m\}$. **for** i = 1 : m **do** Find $R_i^* \in \mathbf{R}$ that is the closest to c_i . $\mathbf{r} = (\mathbf{r}, R_i^*)$. **end for Output:** Representative outer-loops \mathbf{r} .

Algorithm 2 Selection of Representative Policies

Step 1: Residual Diagnostic **Input:** Representative outer-loops \mathbf{r} . Number of representative policies n. The attribute matrix A. Number of inner-loop l. Run the balanced sampling algorithm with $\pi_0 = (n/N, ..., n/N)$ and A.

Output:

A random sample of policies P_0 .

Initialize:

An $n \times m$ matrix L_0

Run the nested simulation algorithm with P_0 , r and l inner-loops for each outer-loop.

Output:

Matrix *L*⁰ containing simulated liabilities.

for j = 1 : n **do**

Obtain the simulated liabilities and the predicted account values in the selected scenarios of policy j, $L_0[j, :]$ and AV_j .

Fit a spline regression using AV_j and $L_0[j, :]$.

Obtain the fitted value $\hat{L}_0[j, :]$ from the fitted spline model. end for

for s = 1 : m **do**

Run regression between $L_0[:, s]$ and $A[\mathbf{P}_0, :]$.

end for

Using the residuals from the regression procedure, identify the parameters α and β in $g(AV) = |AV - \alpha|^{\beta}$ where $\sigma_k(s) = \gamma(s)g(AV_k)$ for k = 1, ..., N. ## Step 2: Representative Policies

Run the balanced sampling algorithm with π $(ng(AV_1)/\sum g(AV_j), ..., ng(AV_N)/\sum g(AV_j))$ and A.

Output:

A set of representative policies **P**.

Algorithm 3 Predictive Distribution of the Total VA Liability

Step 1: Run Reduced Nested Simulation Input: Representative outer-loops r. Number of inner-loop *l*. Representative policies **P**. Initialize: An $n \times m$ matrix L Run the nested simulation algorithm with *P*, *r* and *l* inner-loops for each outer-loop. **Output:** Matrix *L* containing simulated liabilities. Initialize: An $n \times M$ matrix \hat{L} . **for** i = 1 : n **do** Obtain the simulated liabilities and the predicted account values in the selected scenarios of policy *j*, L[j, :] and AV_{j} . Fit a spline regression using AV_i and L[j, :]. Using the fitted parameters, compute the approximated liabilities $\hat{L}[i, :]$ of policy *i* for all scenarios. end for **Output:** Matrix \hat{L} containing all approximated liabilities. ## Step 2: Compute Approximated Total Liabilities Input: The $n \times M$ liability matrix \hat{L} . The inclusion probability of the selected policies π^* . Initialize: A vector \hat{L} of length *m*. **for** s = 1 : m **do** $\hat{\boldsymbol{L}}[s] = \sum_{j=1}^{n} \hat{\boldsymbol{L}}[j, s] / \boldsymbol{\pi}^*[j]$ end for **Output:** The approximated total liability for all scenarios, \hat{L} .

Appendix C. Proof of Theorem 2.2

The following lemma states the infinitive norm for various matrices that will be used in the proof of Theorem 2.2. The proof of these results can be found in Cardot (2000) (Lemma 6.2), Zhou et al. (1998), (Lemma 6.3 and 6.4) and Claeskens et al. (2009) (Lemma A1).

Recalling the definition of N, N(x), $\rho(x)$ in Section 2.2, define $G_{k,n} = (N^t N)/n$, $G = \int_a^b N^t(x)N(x)\rho(x)dx$, $H_{k,n} = G_{k,n} + \lambda D_q/n$ and $H = G + \lambda D_q/n$. As in Lemma A3 of Claeskens et al. (2009), define the constant $K_q = (K + p + 1 - q)(\lambda \tilde{c}_1)^{1/2q}n^{-1/2q}$. We state the following results:

Lemma C.1. $||D_q||_{\infty} = O(\delta^{-2q+1}), ||G_{k,n}^{-1}||_{\infty} = O(\delta^{-1}), ||H_{k,n}^{-1}||_{\infty} = O(\delta^{-1}), \max_{1 \le i, j \le K+p+1} |\{H_{k,n}^{-1} - H^{-1}\}_{i,j}| = o(\delta^{-1}) \text{ for } K_q < 1 \text{ and } ||H_{k,n}^{-1}||_{\infty} = O(\delta^{-1}(1 + K_q^{2q})^{-1}), \max_{1 \le i, j \le K+p+1} |\{H_{k,n}^{-1} - H^{-1}\}_{i,j}| = o(\delta^{-1}(1 + K_q^{2q})^{-1}) \text{ for } K_q \ge 1. ||G_{k,n} - G||_{\infty} = ||H_{k,n} - H||_{\infty} = o(\delta) \text{ and } ||G_{k,n}^{-1} - G^{-1}||_{\infty} = o(\delta^{-1}).$

Let $\hat{f}_{reg}(x)$ denote the regression spline estimator which equals $N(x)(N^tN)^{-1}N^tY$. The following lemma states the relation between the regression spline estimator and the penalized spline estimator $\hat{f}(x) = N(x)(N^tN + \lambda D_q)^{-1}N^tY$.

Lemma C.2. The following relation holds:

$$\hat{f}(x) = \hat{f}_{reg}(x) - \frac{\lambda}{n} N(x) H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t Y,$$

=

Proof. Apply the inverse matrix identity $(A + B)^{-1} = A^{-1} - (A + B)^{-1}BA^{-1}$ with $A = N^t N$ and $B = \lambda D_q$,

$$(N^{t}N + \lambda D_{q})^{-1} = (N^{t}N)^{-1} - (N^{t}N + \lambda D_{q})^{-1}\lambda D_{q}(N^{t}N)^{-1}.$$

The penalized regression estimator can be written as

$$\hat{f}(x) = N(x)(N^{t}N^{-1} - (N^{t}N + \lambda D_{q})^{-1}\lambda D_{q}(N^{t}N)^{-1})N^{t}Y = N(x)(N^{t}N)^{-1}N^{t}Y - \frac{\lambda}{n}N(x)\left(\frac{N^{t}N + \lambda D_{q}}{n}\right)^{-1}D_{q}\left(\frac{N^{t}N}{n}\right)^{-1}\frac{1}{n}N^{t}Y = \hat{f}_{reg}(x) - \frac{\lambda}{n}N(x)H_{K,n}^{-1}D_{q}G_{K,n}^{-1}\frac{1}{n}N^{t}Y. \quad \Box$$

With the above lemmas, we are ready to prove Theorem 2.2.

Proof. According to Lemma C.2,

$$\hat{f}(x) = \hat{f}_{reg}(x) - \frac{\lambda}{n} N(x) H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t Y.$$
(C.1)

Thus, the derivative of the penalized estimator is

$$\hat{f}^{(i)}(x) = \hat{f}^{(i)}_{reg}(x) - \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t Y.$$

The bias of the penalized estimator's derivative is then given by

$$\mathbb{E}\left(\hat{f}^{(i)}(x)\right) - f^{(i)}(x) \\ = \mathbb{E}\left(\hat{f}^{(i)}_{reg}(x)\right) - f^{(i)}(x) - \frac{\lambda}{n}N_{p+1-i}(x)\Delta_{i}H_{K,n}^{-1}D_{q}G_{K,n}^{-1}\frac{1}{n}N^{t}f. \quad (C.2)$$

Zhou and Wolfe (2000) show that

$$\mathbb{E}\left(\hat{f}_{reg}^{(i)}(x)\right) - f^{(i)}(x) = b^{(i)}(x; p+1-i) + o(\delta^{p+1-i}),$$

where $b^{(i)}(x; p + 1 - i) = -\frac{f^{(p+1-i)}(x)(\kappa_{j+1}-\kappa_j)^{p+1-i}}{(p+1-i)!}B_{p+1-i}\left(\frac{x-\kappa_j}{\delta_{j+1}}\right)$ for $x \in [\kappa_j, \kappa_{j+1}]$ and $j = 0, \ldots, K$. $B_l(\cdot)$ is a Bernoulli polynomial of order l, see Abramowitz and Stegun (1965). Next we derive the asymptotic property for the third term in (C.1). Write the third term as

$$\begin{split} &\frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t f \\ &= \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n} \frac{1}{n} N^t (f - s_f + s_f) \\ &= \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t (f - s_f) \\ &+ \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t s_f, \end{split}$$
(C.3)

where $s_f(\cdot) = N(\cdot)\beta$ is the best L_{∞} approximation to f. We first derive the order for the first term in (C.3) Due to definition we have $\|\Delta_i\|_{\infty} = O(\delta^{-i})$, Agarwal and Studden (1980) show in their Lemma 6.10 that $G_{K,n\,n}^{-1} N^t(f-s_f) = o(\delta^{p+1})$ and $G_{K,n\,n}^{-1} N^t(f-s_f) = o(\delta^q)$.

It follows from Lemma C.1 that if $K_q < 1$, then

$$\begin{aligned} &\|\frac{\lambda}{n}N_{p+1-i}(x)\Delta_{i}H_{K,n}^{-1}D_{q}G_{K,n}^{-1}\frac{1}{n}N^{t}(f-s_{f})\|_{\infty} \\ &\leq \frac{\lambda}{n}\|N_{p+1-i}(x)\|_{\infty}\|\Delta_{i}\|_{\infty}\|H_{k,n}^{-1}\|_{\infty}\|D_{q}\|_{\infty}\|G_{K,n}^{-1}\frac{1}{n}N^{t}(f-s_{f})\|_{\infty} \\ &\leq \frac{\lambda}{n}O(\delta^{-i})O(\delta^{-1})O(\delta^{-2q+1})o(\delta^{p+1}) = o(\lambda n^{-1}\delta^{p-2q+1-i}). \end{aligned}$$

Similarly if $K_q \ge 1$, then

$$\begin{split} &\|\frac{\lambda}{n}N_{p+1-i}(x)\Delta_{i}H_{K,n}^{-1}D_{q}G_{K,n}^{-1}\frac{1}{n}N^{t}(f-s_{f})\|_{\infty} \\ &= o(\lambda n^{-1}\delta^{-q}(1+K_{q}^{2q})^{-1}\delta^{-i}) \\ &= o((\lambda/n)^{1/2}\delta^{-q}K_{q}^{q}(1+K_{q}^{2q})^{-1}\delta^{-i}) = o((\lambda/n)^{1/2}\delta^{-q-i}), \end{split}$$
(C.4)

where the last equation follows from the fact that $K_q^q (1+K_q^{2q})^{-1} \le 1/2$ if $K_q \ge 1$. In the next we derive the order for the second term in (C.3). As in Claeskens et al. (2009), apply the mean value theory to the second term, write

$$\begin{aligned} \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} D_q G_{K,n}^{-1} \frac{1}{n} N^t s_f \\ &= \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} \Delta_q^t \int_a^b N_{p+1-q}^t(x) s_f^{(q)}(x) dx \\ &= \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H_{K,n}^{-1} \Delta_q^t W s_f^{(q)}(\tau), \end{aligned}$$
(C.5)

where $s_f^{(q)}(x) = (N(x)\beta)^{(q)} = N_{p+1-q}(x)\Delta_q\beta$ with $\beta = G_{k,n}^{-1}N^t s_f/n$, $W = \text{diag}\{\sum_{l=j}^{j+p-q} \int_{k_l}^{k_{l+1}} N_{j,p+1-q}(t)dt\}$ a diagonal matrix and $\tau = (\tau_{-p+q}, \ldots, \tau_K)^t$ with $\tau_j \in [\kappa_j, \kappa_{j+p+1-q})$ for $j = -p + q, \ldots, K$. Rewrite (C.5) as

$$\frac{\lambda}{n} N_{p+1-i}(x) \Delta_i (H_{K,n}^{-1} - H^{-1}) \Delta_q^t W S_f^{(q)}(\tau) + \frac{\lambda}{n} N_{p+1-i}(x) \Delta_i H^{-1} \Delta_q^t W S_f^{(q)}(\tau).$$
(C.6)

Denote the second term in (C.6) by $b_{\lambda}^{(i)}(x)$ which goes to 0 as $n \to \infty$. Next we derive the order for $\frac{\lambda}{n}N_{p+1-i}(x)\Delta_i(H_{K,n}^{-1} - H^{-1})\Delta_q^t W s_f^{(q)}(\tau)$. It is easy to see that $||W||_{\infty} = O(\delta)$ from its definition. Claeskens et al. (2009) showed that $||s_f^{(q)}(\tau)||_{\infty} = O(1)$, together with Lemma C.1 we have if $K_q < 1$, then

$$\begin{split} &\frac{\lambda}{n} N_{p+1-i}(x) \Delta_i (H_{K,n}^{-1} - H^{-1}) (\Delta_q)^t W s_f^{(q)}(\tau) \\ &= \frac{\lambda}{n} O(\delta^{-i}) o(\delta^{-1}) O(\delta^{-q}) O(\delta) = o(\lambda n^{-1} \delta^{-q-i}), \\ &\text{if } K_q \ge 1, \text{ then} \\ &\frac{\lambda}{n} N_{p+1-i}(x) \Delta_i (H_{K,n}^{-1} - H^{-1}) (\Delta_q)^t W s_f^{(q)}(\tau) \\ &= \frac{\lambda}{n} O(\delta^{-i}) o(\delta^{-1} (1 + K_q^{2q})^{-1}) O(\delta^{-q}) O(\delta) \\ &= o(\lambda n^{-1} \delta^{-i-q} (1 + K_q^{2q})^{-1}) = o((\lambda/n)^{1/2} \delta^{-q-i}), \end{split}$$

in which the last equation follows from the same argument used for (C.4). As a result,

$$\begin{split} \mathbb{E}\left(\hat{f}^{(i)}(x)\right) - f^{(i)}(x) &= b^{(i)}(x; p+1-i) + b^{(i)}_{\lambda} + o(\lambda n^{-1}\delta^{p-2q+1-i}) \\ &+ o(\lambda n^{-1}\delta^{-q-i}) \\ &= O(\lambda n^{-1}\delta^{p-2q+1-i}) + O(\lambda n^{-1}\delta^{-q-i}) \\ &= O(\lambda n^{-1}\delta^{-q-i}) \text{ if } K_q < 1; \\ \mathbb{E}\left(\hat{f}^{(i)}(x)\right) - f^{(i)}(x) &= b^{(i)}(x; p+1-i) + b^{(i)}_{\lambda} + o((\lambda/n)^{1/2}\delta^{-q-i}), \\ &+ o((\lambda/n)^{1/2}\delta^{-q-i}) \\ &= O((\lambda/n)^{1/2}\delta^{-q-i}) \text{ if } K_q \ge 1. \end{split}$$

For the asymptotic properties of the variance, from $\hat{f}^{(i)}(x) = N_{p+1-i}(x)\Delta_i(N^tN + \lambda D_q)^{-1}N^tY$, we have

$$\operatorname{Var}\left(\hat{f}^{(i)}(x)\right) = \frac{\sigma^{2}}{n} N_{p+1-i}(x) \Delta_{i} H_{k,n}^{-1} G_{k,n} H_{k,n}^{-1} (\Delta_{i})^{t} N_{p+1-i}(x)^{t}$$

$$= \frac{\sigma^{2}}{n} N_{p+1-i}(x) \Delta_{i} H_{k,n}^{-1} (H_{k,n} - \lambda D_{q}/n) H_{k,n}^{-1} (\Delta_{i})^{t} N_{p+1-i}(x)^{t}$$

$$= \frac{\sigma^{2}}{n} N_{p+1-i}(x) \Delta_{i} H_{k,n}^{-1} (\Delta_{i})^{t} N_{p+1-i}(x)^{t}$$

$$- \frac{\sigma^{2}}{n} N_{p+1-i}(x) \Delta_{i} H_{k,n}^{-1} \frac{\lambda D_{q}}{n} H_{k,n}^{-1} (\Delta_{i})^{t} N_{p+1-i}(x)^{t}.$$



Fig. D.1. Inclusion probabilities.

Using Lemma C.1 one may write, if
$$K_q < 1$$
, then

$$\begin{split} & \frac{\sigma^2}{n} N_{p+1-i}(x) \Delta_i H_{k,n}^{-1} (\Delta_i)^t N_{p+1-i}(x)^t \\ & - \frac{\sigma^2}{n} N_{p+1-i}(x) \Delta_i H_{k,n}^{-1} \frac{\lambda D_q}{n} H_{k,n}^{-1} (\Delta_i)^t N_{p+1-i}(x)^t \\ & \leq O(n^{-1} \delta^{-2i-1}) + \frac{\lambda}{n^2} O(\delta^{-i}) O(\delta^{-1}) O(\delta^{-2q+1}) O(\delta^{-1}) O(\delta^{-i}) \\ & = O(n^{-1} \delta^{-2i-1}) + O(\lambda n^{-2} \delta^{-2q-2i-1}), \end{split}$$

and if $K_q \geq 1$, then

2

$$\frac{\sigma^2}{n} N_{p+1-i}(x) \Delta_i H_{k,n}^{-1}(\Delta_i)^t N_{p+1-i}(x)^t - \frac{\sigma^2}{n} N_{p+1-i}(x) \Delta_i H_{k,n}^{-1} \frac{\lambda D_q}{n} H_{k,n}^{-1}(\Delta_i)^t N_{p+1-i}(x)$$

$$\leq O(n^{-1}\delta^{-2i}\delta^{-1}(1+K_q^{2q})^{-1})$$

$$+ \frac{\lambda}{n^2} O(\delta^{-i}) O(\delta^{-1}(1 + K_q^{2q})^{-1}) O(\delta^{-2q+1}) O(\delta^{-1}(1 + K_q^{2q})^{-1}) O(\delta^{-i})$$

$$= O(n^{-1}\delta^{-2i-1}K_q^q(1 + K_q^{2q})^{-1}(\lambda/n)^{-1/2})$$

$$+ O(\lambda n^{-2}\delta^{-2q-2i-1}K_q^{2q}(1 + K_q^{2q})^{-2}(\lambda/n)^{-1})$$

$$= O((\lambda n)^{-1/2}\delta^{-2i-1}) + O((\lambda n)^{-1}\delta^{-2q-2i-1})$$

Appendix D. Simulation study of another synthetic VA portfolio whose attributes follow uniform distributions

In order to show the wide applicability of our proposed algorithm, we present and evaluate the performance of the algorithm for another VA portfolio containing 100,000 policies that is similarly constructed in Gan and Lin (2015). The attributes of this portfolio are the same as given in the second column of Table 4.1. Different from the portfolio constructed in Section 4, in this section the distributions of the attribute variables are assumed uniform.

From the residual diagnostic stage, we find the account value is mostly associated to the variation of the residual and the variation is larger in the regions where the account values are either small or large. After standardizing the residuals by $g(AV_{k,0}) = |AV_{k,0} - 2.5 \times 10^5|^{1/10}$ the variation among different account values becomes much more homogeneous. This implies the inclusion probability for unit *k* in the population is

$$\pi_k = n \frac{|AV_{k,0} - 2.5 \times 10^5|^{1/10}}{\sum_k |AV_{k,0} - 2.5 \times 10^5|^{1/10}},$$

where k = 1, 2, ..., N. In other words, policies with either smaller or larger account values have higher probabilities to be included in the representative set in order to control the total anticipated MSE of the estimator. The inclusion probabilities are plotted against the initial account values in Fig. D.1

The two predictive distributions obtained from the full simulation algorithm and the proposed simulation algorithm are compared in Fig. D.2. From the histograms, the relative errors and the Q–Q plot it can be seen that the predictive distribution is well approximated. The relevant statistics of the predictive distributions are presented in Table D.1. Similar to the synthetic portfolio studied in Section 4 almost all of the relative errors are within 1% (see Fig. D.3).

Appendix E. Numerical studies with guarantee fees included

For illustration purposes, in the paper we assume no fees for the guarantee riders. In the following we briefly report the results of several more numerical studies in which the guarantee fees are considered. We assume that at the end of each period fees are deducted from the policy's subaccount before any guarantee payments are made (withdrawal benefit, death benefit, etc.).

In particular, three fee schemes are studied: 100 bps (1%) fee for all policies; 200 bps fee for all policies and variable fee



Fig. D.2. Distributions of predicted total liability.



Fig. D.3. Relative error.

 Table D.1

 Comparison of different statistics.

-			
	Full nested-simulation	Fast nested-simulation	Relative error
Mean	2,315,647,767	2,323,979,465	0.36%
Standard deviation	346,641,044	346,383,216	0.07%
Skewness	2.6962	2.7234	1.01%
Kurtosis	15.8542	15.9712	0.74%
VaR(90)	2,732,596,388	2,739,858,371	0.27%
CVaR(90)	3,105,938,460	3,113,825,169	0.25%
VaR(95)	2,971,191,573	2,959,823,966	0.38%
CVaR(95)	3,379,334,909	3,390,704,638	0.34%
VaR(99)	3,491,688,367	3,486,979,014	0.13%
CVaR(99)	3,379,334,909	3,390,704,638	0.34%

Table E.1

Relative errors of different risk metrics under different fee schemes.

Risk metrics	Flat fee of 100 bps	Flat fee of 200 bps	Variable fees
Mean	0.39%	0.11%	0.92%
Standard deviation	1.63%	1.18%	1.51%
Skewness	1.93%	0.88%	0.63%
Kurtosis	2.72%	1.31%	1.01%
VaR(90)	0.40%	0.15%	0.94%
CVaR(90)	1.07%	0.29%	1.91%
VaR(95)	0.46%	0.15%	0.99%
CVaR(95)	0.66%	0.21%	1.83%
VaR(99)	0.43%	0.21%	1.21%
CVaR(99)	0.39%	0.25%	0.42%

depending on the guarantee type. For variable fee scheme, we assume the followings:

- GMDB roll-up: 50 bps per year,
- GMDB ratcheting: 100 bps per year,
- GMWB: 50 bps per year,
- GMAB roll-up: 50 bps per year,
- GMAB ratcheting: 100 bps per year.

For example, if a policy has both a GMDB of ratcheting type and a GMWB rider, then the total rider fee will be charged at 150 bps per year. The relative errors, similar to those reported in Table 4.2 in Section 4, under different fee schemes are given in Table E.1. The numerical results show that the proposed algorithm performs well with different fee schedules. In general, the relative errors for variable fees are higher due to the extra non-homogeneity added into the portfolio attributes.

References

- Abramowitz, M., Stegun, I.A., 1965. Handbook of Mathematical Functions: With Formulas, Graphs, and Mathematical Tables. Courier Corporation. Agarwal, G.G., Studden, W., 1980. Asymptotic integrated mean square error using
- least squares and bias minimizing splines. Ann. Statist. 1307–1325.
- Aloise, D., Deshpande, A., Hansen, P., Popat, P., 2009. NP-hardness of Euclidean sum-of-squares clustering. Mach. Learn. 75 (2), 245–248.
- Areal, N., Rodrigues, A., Armada, M.R., 2008. On improving the least squares Monte Carlo option valuation method. Rev. Deriv. Res. 11 (1–2), 119.
- Bacinello, A.R., Millossovich, P., Olivieri, A., Pitacco, E., 2011. Variable annuities: A unifying valuation approach. Insurance Math. Econom. 49 (3), 285–297.
- Bauer, D., Ha, H., 2015. A least-squares Monte Carlo approach to the calculation of capital requirements. In: World Risk and Insurance Economics Congress, Munich, Germany, pp. 2–6.
- Bauer, D., Reuss, A., Singer, D., 2012. On the calculation of the solvency capital requirement based on nested simulations. ASTIN Bull.: J. IAA 42 (2), 453–499.
- Bollen, N.P., 1998. Valuing options in regime-switching models. Journal of Derivatives 6, 38–50.
- Cardot, H., 2000. Nonparametric estimation of smoothed principal components analysis of sampled noisy functions. J. Nonparametr. Stat. 12 (4), 503–538.
- Cathcart, M., Morrison, S., 2009. Variable annuity economic capital: the least-squares Monte Carlo approach. Life Pensions 36–40.
- Claeskens, G., Krivobokova, T., Opsomer, J.D., 2009. Asymptotic properties of penalized spline estimators. Biometrika 96 (3), 529–544.
- De Boor, C., 1978. A Practical Guide to Splines. Springer, New York.
- Deville, J.-C., Tillé, Y., 2004. Efficient balanced sampling: the cube method. Biometrika 91 (4), 893–912.
- Feng, R., Cui, Z., Li, P., 2016. Nested stochastic modeling for insurance companies. Retrieved from https://www.soa.org/globalassets/assets/files/staticpages/research/nested-stochastic-modeling-report.pdf.
- Friedman, J.H., Stuetzle, W., 1981. Projection pursuit regression. J. Amer. Statist. Assoc. 76 (376), 817–823.
- Gan, G., 2013. Application of data clustering and machine learning in variable annuity valuation. Insurance Math. Econom. 53 (3), 795–801.
- Gan, G., Lin, X., 2015. Valuation of large variable annuity portfolios under nested simulation: a functional data approach. Insurance Math. Econom. 62, 138–150.
- Gan, G., Valdez, E.A., 2018. Regression modeling for the valuation of large variable annuity portfolios. N. Am. Actuar. J. 22 (1), 40–54.
- Godambe, V., 1955. A unified theory of sampling from finite populations. J. R. Stat. Soc. Ser. B Stat. Methodol. 17 (2), 269–278.
- Ha, H., Bauer, D., 2015. A least-squares Monte Carlo approach to the pricing of GMWBs. Retrieved from http://www.wriec.net/wp-content/uploads/2015/07/ 8B1_Ha.pdf.
- Hardy, M.R., 2001. A regime-switching model of long-term stock returns. N. Am. Actuar. J. 5 (2), 41–53.
- Hejazi, S.A., Jackson, K.R., 2016. A neural network approach to efficient valuation of large portfolios of variable annuities. Insurance Math. Econom. 70, 169–181.

- Hörig, M., Murray, K., Phelan, E., Leitschkis, M., 2013. An application of Monte Carlo proxy techniques to variable annuity business: A case study. Retrieved from http://de.milliman.com/uploadedFiles/insight/2013/monte-carlo-proxytechniques.pdf.
- Huang, Y.T., Kwok, Y.K., 2016. Regression-based Monte Carlo methods for stochastic control models: Variable annuities with lifelong guarantees. Quant. Finance 16 (6), 905–928.
- Isaki, C.T., Fuller, W.A., 1982. Survey design under the regression superpopulation model. J. Amer. Statist. Assoc. 77 (377), 89–96.
- Koursaris, A., 2011. A Least Sqaures Monte Carlo Approach to Liability Proxy Modelling and Capital Calculation. Barrie & Hibbert.
- Lin, X.S., Yang, S., 2020. Efficient dynamic hedging for large variable annuity portfolios with multiple underlying assets. Working Paper.
- Longstaff, F.A., Schwartz, E.S., 2001. Valuing American options by simulation: a simple least-squares approach. Rev. Financ. Stud. 14 (1), 113–147.
- MacQueen, J., et al., Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, Oakland, CA, USA, pp. 281–297.
- Nedyalkova, D., Tillé, Y., 2008. Optimal sampling and estimation strategies under the linear model. Biometrika 95 (3), 521–537.
- O'Sullivan, F., 1986. A statistical perspective on ill-posed inverse problems. Statist. Sci. 502–518.
- Särndal, C.-E., Swensson, B., Wretman, J., 2003. Model Assisted Survey Sampling. Springer Science & Business Media.
- Stone, C.J., 1982. Optimal global rates of convergence for nonparametric regression. Ann. Statist. 1040–1053.
- Tillé, Y., 2006. Sampling Algorithms. Springer.
- Zhou, S., Shen, X., Wolfe, D., 1998. Local asymptotics for regression splines and confidence regions. Ann. Statist. 26 (5), 1760–1782.
- Zhou, S., Wolfe, D., 2000. On derivative estimation in spline regression. Statist. Sinica 93-108.