ACT496

Raj Patel

2020-05-07

Contents

Annuity and its Components

Annuities are financial instruments that can guarantee lifetime stream of income during retirement. Two primary reasons why an individual buys annuities are: (1) accumulate money for retirement through tax-deferred savings (2) guaranteed monthly income lasting as long as the individual lives. However, besides these, unlike other investments, annuities also provide a variety of benefit alternatives such as protect against untimely death, provide principal guarantees, assure a specified amount of income when the contract is annuitized and/or guarantee withdrawals for life.

In general, most annuity contracts are issued by life insurance companies. In the contract, contract owner, annuitant and beneficiary are identified.

Contract Owner:

In the contract, the owner of the annuity who pays the premium is the contract owner. Generally, the owner is an individual or a couple. However, a trust or a partnership can be an owner as well (special tax rules apply for these cases).

Annuitant:

In the contract, the annuitant is the person upon whose life the annuity payments are based. Generally, the contract owner is the annuitant. So, the payments continue as long as the owner is alive. In some cases, two people, such as owner and the spouse, are designated as joint annuitants. As a result, the income continues as long as either one survives. Such annuities are called joint or survival annuities

Beneficiary:

Beneficiary is the person who receives any payments that may be due upon the death of the owner or annuitant.

Now that we know the basics of annuities, we will take a look at Variable Annuities.

Variable Annuity:

This is purchased either with a single premium or with periodic payments to help save for retirement. These premiums, based on the choice of contract owners from a wide range of investment options, are then invested. These are invested in a single underlying mutual fund or, in some cases, in a "fund of funds," which is a mutual fund that invests in several other mutual funds or in exchange-traded funds (ETFs). All of these are termed as VA subaccounts.

In this type of annuity, the contract owner determines the point at which accumulated principal and earnings are converted into a stream of income. However, the contract value or the income payments vary based on the investment performance of underlying subaccounts.

Variable Annuity has two phases: (1) Accumulation phase or a savings phase (2) Payout or retirement income phase. As with mutual funds, the investment return of variable annuities fluctuates. During the accumulation phase, the contract value varies based on the performance of the underlying subaccounts chosen. During the payout phase of a deferred variable annuity (and throughout the entire life of an immediate variable annuity), the dollar amount of the annuity payments may fluctuate, again based on how the portfolio performs.

Fees and Expenses

A variable annuity involves direct expenses in the form of insurance charges and indirect expenses in the form of management and other fees and expenses associated with the underlying mutual funds in which the variable annuity subaccounts invest.

The fees and charges commonly associated with variable annuities include mortality and expense risk charges (M&E fees), administrative charges, and distribution charges. In most contracts, the M&E fee pays for three important insurance guarantees:

- The ability to choose a payout option that provides an income that cannot be outlived at rates set forth in the contract at the time of purchase
- A death benefit to protect beneficiaries
- The promise that the annual insurance charges will not increase.

GMDB

Variable annuity contracts have traditionally offered a guaranteed minimum death benefit (GMDB) during the accumulation period that is generally equal to the greater of (a) the contract value at death or (b) premium payments minus any prior withdrawals. The GMDB gives contract owners the confidence to invest in the stock market, important in keeping up with market inflation, as well as the security to know their families will be protected against financial loss in the event of an untimely death.

Some life insurance companies offer death benefits that step up or increase based on pre-determined criteria. Called contract anniversary value or ratchet, these enhanced GMDBs are equal to the greater of (a) the contract value at death, (b) premium payments minus prior withdrawals, or (c) the contract value on a specified prior date. The specified date could be a prior contract anniversary date, such as the date at the end of every seven-year period, every anniversary date, or even more often. A ratchet GMDB locks in the contract's gains on each of the dates specified.

Some insurers offer a rising floor GMDB that is equal to the greater of (a) the contract value at death or (b) premium payments minus prior withdrawals, increased annually at a specified rate of interest. In some cases, a ratchet and a rising floor may be available within the same contract. Some contracts offer a choice of a ratchet or a rising floor.

GMLB

Guaranteed Minimum Income Benefit

A guaranteed minimum income benefit (GMIB) rider is designed to provide the investor with a base amount of lifetime income when they retire regardless of how the investments have performed. It guarantees that if the owner decides to annuitize the contract, payments are based on the amount invested, credited with an interest rate typically 4-5%. An investor must annuitize to receive this benefit and there is typically a seven-ten year holding period before it can be exercised.

Guaranteed Minimum Accumulation Benefit

A guaranteed minimum accumulation benefit (GMAB) rider guarantees that an owner's contract value will be at least equal to a certain minimum percentage (usually 100%) of the amount invested after a specified number of years (typically 7-10 years), regardless of actual investment performance. However, considering the financial risk associated with this, many GMABs require some form of asset allocation.

Guaranteed Minimum Withdrawal Benefit

A guaranteed minimum withdrawal benefit (GMWB) rider guarantees that a certain percentage (usually 5-7%) of the amount invested can be withdrawn annually until the entire amount is completely recovered, regardless of market performance.

Annuity Sales in US (in $ Billions)

	Variable	Fixed	Total
2015	133.0	102.7	235.7
2016	104.7	117.4	222.1
2017	98.2	105.3	203.5
2018	100.2	133.6	233.8
2019	101.9	139.8	241.7

Within Variable Annuity(VA) in US, Jackson has been the leader in the VA market for seven straight years. The top three VA sellers were Jackson, Equitable Financial and TIAA. Together, they represented 36% of the total VA market in 2019.

While within Fixed Annuity ib US, sales reached $139.8 billion in 2019, up 5% from the prior year. AIG led the sales for second consecutive year, selling Fixed Annuities worth $13.2 billion. While the other two in top three were New York Life and Allianz Life of North America. Together, they represented 23% of the U.S. fixed annuity market.

Valuation:

Consider a VA with subaccount invested in a mutual fund, S_t with return μ and standard deviation σ. We can model S_t using a binomial tree. Let us take a simple look at 'Return of Premium' guarantee. Let's say we are considering this a liability of GMAB(Guaranteed Minimum Accumulation Benefit) policy.

In this case, if S_T is the account value at T and g is the minimum guaranteed rate of return compunded continuously, the the liability is: $\max\{e^{gT}S_0 - S_T, 0\}$. On looking at the liability closely, we can say that this resembles a Put Option with K = $e^{gT}S_0$

Simplified Worked Example

Question

Let's say we are interested in valuing a single premium variable annuity with a Return of Premium GMAB rider maturing in 3 years. The subaccount of the VA with initial value of $100 (the single premium) is invested in an index fund with the same initial value and the fund follows a recombining binomial tree with annual time steps.

For each period, the return $u = 1.2$ when the index value goes up and the return $d = 0.85$ when the index value goes down. The annual interest rate compounded continuously is $r = 5\%$.

The benefit of the GMAB rider is equal to the greater of (a) the subaccount value or (b) the total premiums paid at the end of 3 years. We assume that there is no mortality risk, i.e the annuitant will be living at the end of 3 years.

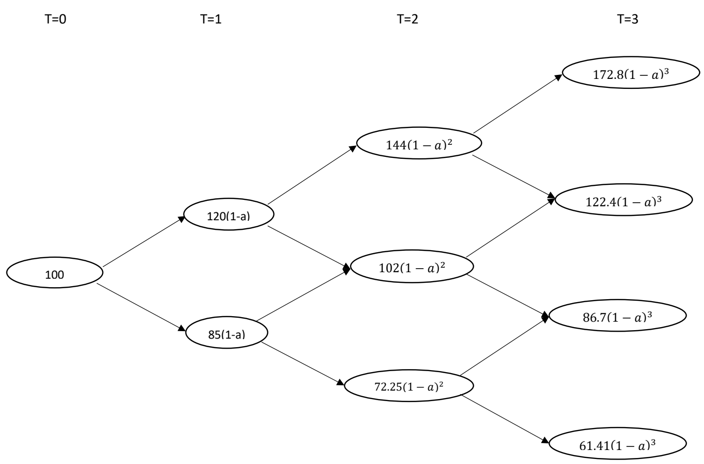
In this case, we want to find the fair annual fee for the GMAB as a percentage of subaccount value.

Solution

To find the fair price, we need to equate the present value of the liability of the guarantee at T = 3 with the total fee received by the insurer under the risk neutral measure.

So, firstly, using $u = 1.2$, $d = 0.85$ and the annual interest rate compounded continuously $r = 5\%$, we find the risk neutral probability which is as follows: $p = \frac{e^r - d}{u-d}$, which gives us $p = 0.575$

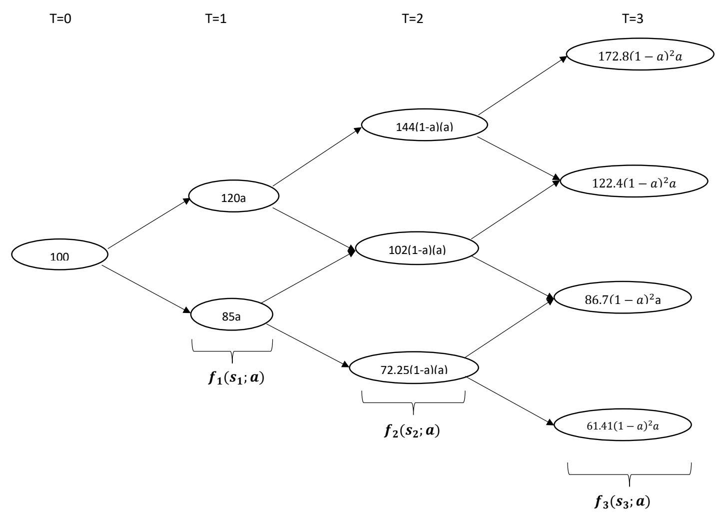
Following that, if we look at the binomial tree of the subaccount after applying a fee of "a%", we have this:



Based on that, the liability of the guarantee at T=3 is as follows:

S_3	A$(s_3; a)$
S_0u^3	max(0,100-172.8$(1-a)^3$)
S_0u^2d	max(0,100-122.4$(1-a)^3$)
S_0ud^2	max(0,100-86.7$(1-a)^3$)
S_0d^3	max(0,100-61.41$(1-a)^3$)

Now, if we look at the fee collected by the insurer, it is as follows:



Now, to compute the fair price, we want to equate the present value of the liability of the guarantee at T = 3 with the total fee received by the insurer under the risk neutral measure. So, the valuation equation will be:

$$e^{-r(\delta t)}\mathbb{E}_{\mathbb{Q}}[f_1(s_1; a)] + e^{-r(2\delta t)}\mathbb{E}_{\mathbb{Q}}[f_2(s_2; a)] + e^{-r(3\delta t)}\mathbb{E}_{\mathbb{Q}}[f_3(s_3; a)] = e^{-r(3\delta t)}\mathbb{E}_{\mathbb{Q}}[A(s_3; a)]$$

Using the risk-neutral probability, and we need to find "a". Using numerical techniques (in Excel, we can use 'Goal Seek'), we can find "a".

Upon finding "a = 1.9262%", we get the following:

Liability of guarantee at T=3

Value of S at different nodes	Value	Expected Value	Present Value at t=0
172.80	0.00	0.00	0.00
122.40	0.00	0.00	0.00
86.70	18.21	1.89	1.63
61.41	42.07	3.23	2.78

Total fee received by the Insurer

Time	Node	Value at the node	Expected Value	Present Value at t=0
1	u	2.31	1.33	1.26
1	d	1.64	0.70	0.66
2	uu	2.72	0.90	0.81
2	ud	1.93	0.47	0.43
2	dd	1.36	0.25	0.22
3	uuu	3.20	0.61	0.52
3	uud	2.27	0.32	0.27
3	udd	1.61	0.17	0.14
3	ddd	1.14	0.09	0.08

In this case, summation of Present Value for both the tables is $4.41

Liability using Simulation

Now that we saw how to calculate the fair annual fee for a simplified case, we will try to understand how to calculate the liability of a VA (without fees) when the underlying asset follows Geometric Brownian Motion (GBM).

In the previous section, we saw how to model GMAB with a 'Return of Premium' guarantee and observed that the liability of GMAB in that case was simply a put option. So, in this section, we will focus on GMWB and GMDB.

Liability of a Plain Vanilla GMWB (Approach 1):

Firstly, a Plain Vanilla GMWB has no bonus or step-up feature. Furthermore, it is designed for a single annuitant and the total amount that can be withdrawn is fixed at the inception of the policy. For pricing, we also make some additional assumptions:

- Interest rates are constant.
- Underlying asset follows a lognormal distribution.
- No lapses or mortality decrements are considered.
- Amount withdrawn each period is equal to maximum amount permitted to withdraw under the contract that doesn't incur surrender penalty.
- The contract matures when the guaranteed total benefit amount is withdrawn.

Under these assumptions, a VA with GMWB can be viewed as a term certain annuity plus a call option. Since we assume that the policyholder exercises the GMWB from the first contract year until the end with no surrenders or deaths, the call option has a fixed exercise time at maturity. This makes it easy to value. Furthermore, the cash flows from the term certain annuity are also fixed, making its present value easily computable.

Now, to price, we assume that GMWB is exercised at time 0. The GMWB guaranteed withdrawal rate is denoted by g and the guaranteed Maximum Annual Withdrawal Amount (MAWA) is denoted by $w = g * A_0$ (where A_0 is the initial account value invested in a fund whose value at time t is S_t). The annual withdrawal amount is assumed to be equal to w and withdrawals last for T years from time 0 such that: $T = \frac{A_0}{w}$. Now, if we use quarterly steps, then the length of a single period is $h = \frac{1}{4}$. Each withdrawal amount is equal to w*h, and so, the total number of time steps is: $N = [\frac{T}{H}]$

Now, following each withdrawal, the account value is reduced but it can never fall below zero. Once withdrawals deplete the account, the account value will be set to zero and remain zero for the rest of the term. For convenience, we introduce a so called shadow account B_t which tracks the fund performance until the maturity regardless of whether the actual account has terminated or not. Now, $B_t = A_t$ when the actual account value A_t is larger than zero. However, the shadow account's value becomes negative when the actual account value A_t is zero. So, their relation can be expressed as: $A_t = max(B_t, 0)$

We assumed the policyholder withdraws w*h at the end of each time step if the account value is positive. However, even when the account value becomes zero, the policyholder will continue to receive w*h at the end of each time step for N times in total. As a result, cash flows from these N payments is equivalent to a T -year annuity certain with a periodic payment of w*h. Besides these cash flows, at maturity, the policyholder may also have a positive account balance $A_T = max(B_T, 0)$ which can be seen as a call option payoff.

The market value of this option is certainly less than the initial account value A_0 as the annuity account has fees(although we will not be considering them as we are only interested in calculating liability) and withdrawals deducted periodically. Now, the withdrawals taken by the policyholder at a fixed amount do not change with the account value. So, we can consider them as an annuity. The present value of the annuity certain is:

$$\sum_{i=1}^{N}(wh)e^{-irh} = (wh) * \frac{1-e^{-rT}}{e^{rh}-1}$$

Based on the above equation, we see that the annuity certain does not depend on the account value. To calculate the present value of the remaining liability, we need to find the expected discounted payoff of the account balance (which was similar to a call option) at maturity under the risk-neutral measure $\mathbb{Q}$. We can write that as:

$$\mathbb{E}_Q[e^{-rT}max(B_T, 0)]$$

So, the total present value for liabilities for a Plain Vanilla GMWB is:

$$\boxed{PV = (wh) * \frac{1-e^{-rT}}{e^{rh}-1} + \mathbb{E}_Q[e^{-rT}max(B_T, 0)]}$$

Example : Here, we show pricing of a liability of a GMWB with following parameters for the underlying asset (which we assume follows a Geometric Brownian Motion(GBM))

Parameter	Input
S_0 (Initial Asset Value)	\$100
μ	0.1
σ	0.18
T	10 (years)
δT	0.004 (1 day, considering 250 business days in a year)
r	0.04
w (withdrawal rate)	0.10
h (time step)	1
Simulations	1000

Based on the above inputs, we get the following result for 1000 simulations for pricing a GMWB liability:

The present value of the annuity certain: \$80.78

The total present value for 1000 scenarios of the remaining liability (the policyholder may have a positive account balance), which is the expected discounted payoff of the account balance at maturity under the risk-neutral measure: \$74304.39

Therefore, the expected present value of the remaining liability for a single policy: \$74.30

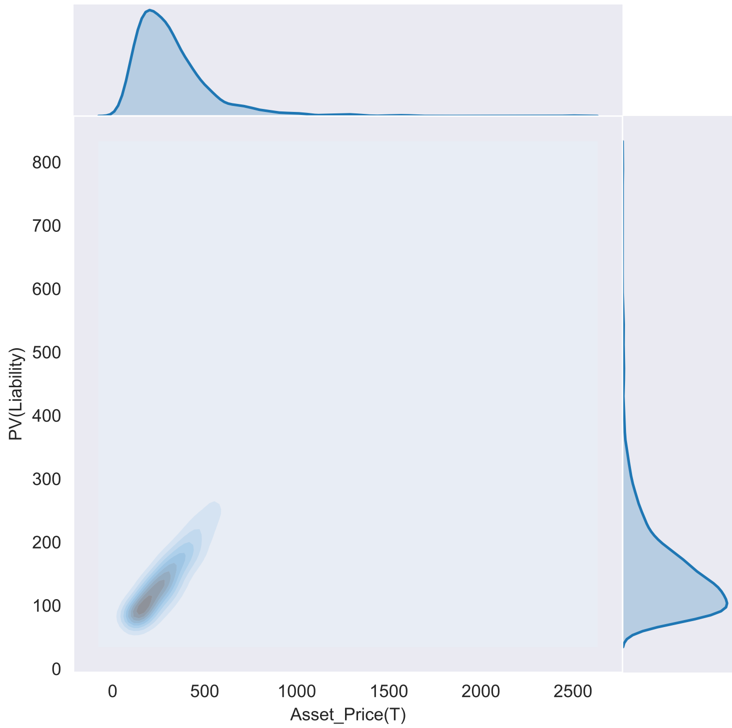


Figure 1: Kernel Density Estimation for GMWB using Approach 1

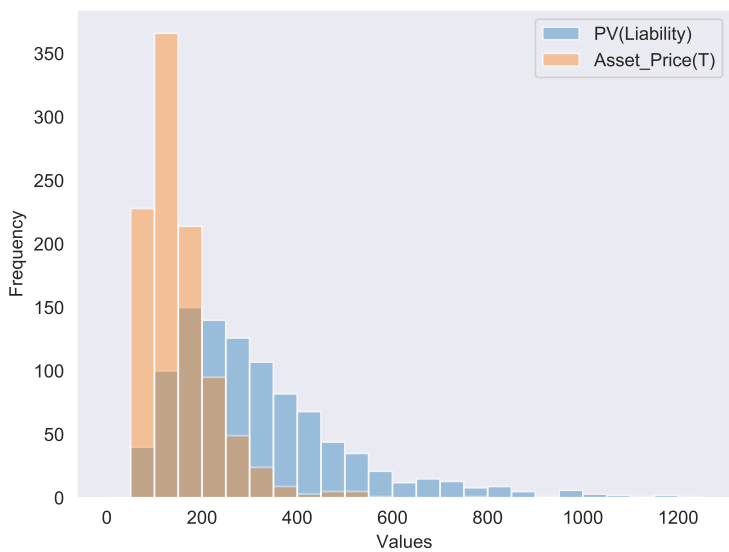


Figure 2: Histogram based on 1000 simulations for GMWB using Approach 1

Hence, the expected present value of the total liability for a single policy: $\boxed{\$155.08}$

Liability of a Plain Vanilla GMWB (Approach 2):

Under this approach, we will exploit the similarity of a GMWB with a put option. The advantage of using this approach over the previous one is the flexibility to deal with withdrawal strategies. Under this, we can incorporate dynamic withdrawals and lapses.

Assume that the policyholder makes equal withdrawals wh at the end of each time step. Now, the account value will not be zero unless these withdrawals are made. Withdrawals are made only at the end of end of a time step, so we will keep a track as to when the account value becomes 0. Moreover, we denote the number of timestep as the subscript to denote the account value at the end of that time step.

Following that, we define a random variable t^* as the time step when the account value becomes zero. So, it is:

$$t^* = \begin{cases} t & A_{t-1} > 0, A_t = 0, 0 < t \leq N \\ N & A_N > 0 \end{cases}$$

The shadow account value B is same as the actual account value when A is positive. However, B can be negative when A changes to 0. So, we continue to deduct withdrawals from B however, A does not change once it is set to 0.

Next, we denote the present value of the guaranteed benefit payments once the account is depleted by G. Initially, there are $N = \frac{T}{h}$ guaranteed benefit payments of wh. So, it time i, it would be:

$$G_i = wh \sum_{j=1}^{N-i} e^{-rjh}$$

where $i = 1, \ldots, N-1$

At N, as there will be no benefit payments left, $G_N = 0$.

Hence, the payoff at time t^*h will be: $max(0, G_{t^*} - B_{t^*})$. However, if the policyholder delays annuitization, the account value at maturity may be larger than withdrawal amount and hence there might be a benefit amount left. In that case, the payoff will be: $max(0, G_N - B_N)$.

So, under risk neutral measure $\mathbb{Q}$, the present value of the liability would be:

$$PV(liability) = \boxed{\mathbb{E}_{\mathbb{Q}}[e^{-rt^*h} max(0, G_{t^*} - B_{t^*})\mathbb{I}_{0<t^*\leq N}] + \mathbb{E}_{\mathbb{Q}}[e^{-rT} max(0, G_N - B_N)\mathbb{I}_{G_N>0, B_N>0}]}$$

Liability of a GMDB:

Here, we will show how to price a GMDB that guarantees a return of at least the original invested premium compounded at some annual growth rate g. So, the payout to the policyholder is: $max(e^{g\tau} AV_0, AV_\tau)$.

Here, the time of death τ is a random variable, $g \geq 0$ is the guaranteed instantaneous growth rate, AV_0 is the initial account value while AV_τ is the account value at the time of the death of the policyholder. Here, to simplify, we consider that there are no fees.

Unlike standard financial options, these GMDBs have have stochastic maturity but their exercise is triggered by involuntary death. As a result, in the literature they are called Titanic options. However, for simplicity, we assume deterministic life time such that $\tau = T$ for the policyholder.

As $\tau = T$, the payoff for the policyholder is:

$$max(e^{gT} AV_0, AV_T) = \boxed{(e^{gT} AV_0 - AV_T)^+ + AV_T}$$

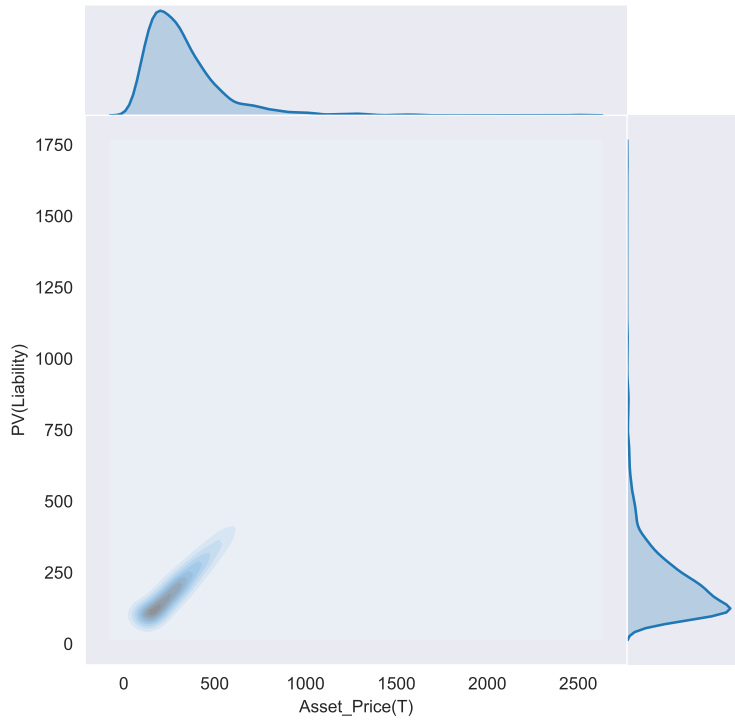


Figure 3: Kernel Density Estimation for GMDB

,where AV_t denotes the policyholder's account value at time t.

From the above result, we see that $(e^{gT}AV_0 - AV_T)^+$ is a weighted European Put Option with strike price $K = e^{gT}AV_0$ and the underlying asset AV_T.

Example : Here, we show pricing of a liability of a GMDB with following parameters for the underlying asset (which we assume follows a Geometric Brownian Motion(GBM))

Parameter	Input
S_0 (Initial Asset Value)	\$100
μ	0.1
σ	0.18
T	10 (years)
δT	0.004 (1 day, considering 250 business days in a year)
r	0.04
Simulations	1000

Based on the above inputs, we get the following result for 1000 simulations for pricing a GMDB liability:

Total liability at the end of 10 years for all 1000 scenarios: \$333341.67

Present value of this total liability for these 1000 scenarios: \$223445.60

Therefore, expected present value of liability for a single policy is $\boxed{\$223.44}$

Now that we know how to find present value of the liability for GMAB, GMWB and GMDB, we will move to efficient simulation techniques.

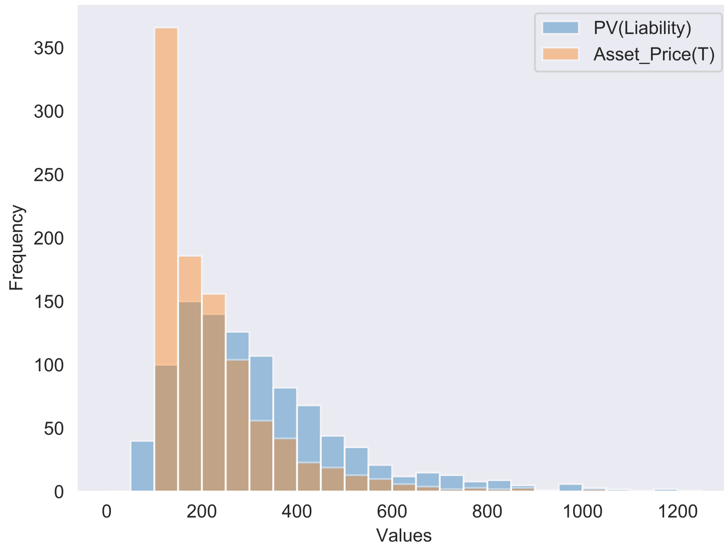


Figure 4: Histogram based on 1000 simulations for GMDB

Nested Simulation for Valuation:

Although we observed the simple case of finding the present value of liabilities for different riders, in more realistic scenarios, closed form expressions are not available for calculating the VA liabilities. To overcome that, companies use the nested-simulation algorithm.

The nested simulation is a two-step simulation algorithm comprising of an outer loop and an inner loop. Without loss of generality, we consider time t=0 as the valuation time and time t=1 as the future time. Now, as part of the first step of this algorithm, each policy's account value is projected from time t=0 to t=1 using many outer-loops (where each outer-loop represents a real world scenario). In the second step, a large number of inner loops are simulated at each outer-loop to calculate the VA liability. This is done by averaging the present value of all the insurer's cash flows occuring as part of the inner-loop simulation. The sum of PV of these cash flows/liabilities for different outer-loop simulations for these policies give the distribution of total liability at time t=1 and this distribution is defined as predictive total liability distribution.

Due to demographic variation and different designs, a portfolio contains highly non-homogeneous contracts. As a result, running the nested simulation for a large VA portfolio can be time consuming. For a portfolio with 200,000 policies with 1000 outer-loops and 5000 inner-loops in each outer-loop, if we assume that the computer can process 2,000,000 simulations per second, then the algorithm will run for $\frac{200{,}000*1000*5000}{2{,}000{,}000}$ seconds which is almost 6 days. As a result, this is just not feasible from a risk management perspective.

Recent Developments and thier limitations:

Considering the need for better methods for liability valuation, quite a lot of research has been done in the domain. Among that, most of the work relies on Least Square Monte Carlo (LSMC) method. The idea under this method is to approximate the VA liability by regressing the guaranteed payoffs on a set of polynomials. The procedure starts at the termination date and propogates backward through time.

However, there are couple of issues associated with this method. First is the unstable performance across different circumstances due to the accumulation of the estimation error through the backward procedure. While another issue is the determination of the order of polynomial basis function. When using this method on VA portfolios, these issues can even get magnified as VA portfolios contain a large number of non-homogeneous contracts.

However, research of efficient nested-simulation for large VA portfolios has been quite recent. Gan (2013), Gan and Lin (2015), Hejazi and Jackson (2016) and Gan and Valdez (2018) have all proposed unique ideations on efficient nested-simulation for VA portfolions. However, a common issue in these research has been the theoretical justification on the estimator of the quantity of interest. All of these primarily focused on compressing the number of policies. So, for the case where the underlying investment by the policyholders involves multiple assets, these methods will be inefficient as there will be large number of inner and outer loop projections.

Surrogate Modelling Approach by Lin and Yang:

To address all the limitations in the work done till date, Lin and Yang (2020) proposed a method in which both, the number of inner and outer loop projections and the number of policies are reduced. Under this, they first propose to use a model assisted population sampling framework to reduce the number of policies. Following that, they propose to use spline regression with scenario clustering to reduce the number of inner and outer loop projections. Since they incorporate different statistical models in this algorithm, they call this a surrogate model assisted nested simulation algorithm.

Moving beyond Linearity while modelling:

Before going into the Surrogate Modelling approach, we need to understand what spline regression is. Within regression, once we relax the linearity assumption, there are multiple tools we can use. The first basic extension is Polynomial Regression.

Polynomial Regression is an extension of Linear Regression where alongside the linear predictors, we also use powers of predictors to model the response variable. So, instead of the standard model where y_i is the response, x_i is the predictor, β_i is the coefficient of the predictor and ϵ_i is the error:

$$y_i = \beta_0 + \beta_1 x_i + \epsilon_i$$

we have

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \cdots + \beta_m x_i^m + \epsilon_i$$

When the degree m is large, we can fit a highly non-linear function using Polynomial Regression.

However, Polynomial Regression imposes a global structure on the entire dataset. To overcome that, we can use Step Functions. Step Functions dissect the range of variable into K distinct regions. Once it is divided into K regions, it fits a different constant on each region. However, although the Step Functions help us overcome the global structural issue, they can only fit constants on different regions and hence can't help us capture attributes like trend in the data.

To overcome both the issues, Regression Spline is very useful and is in fact an extension of the above mentioned tools. Regression Spline divides the range of predictor into K distinct regions. Within each region, it fits a polynomial function. While fitting, these polynomials are constrained so that they join smoothly at the boundaries. As a result, in Regression Spline, the more the regions we separate the variable into, the more flexible the fit is.

A slightly different approach is to use Smoothing Spline. In Smoothing Spline, to fit the parameter $\beta's$ for our function, we use penalized least squares. Let's say we are trying to find g(x) that fits the observed data well. In that case, we would want to find function g that minimizes:

$$\sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int g''(t)^2 dt$$

where λ is a nonnegative tuning parameter selected using Cross Validation. The function g that minimizes the above equation is called the smoothing spline.

Spline Regression Approach:

Under this approach, firstly, a set of representative outer loops is selected and then VA liabilities are calculated at those outer loops through multiple inner loops. This is done using scenario clustering. Once these are calculated, these VA liabilities are used to calculate true VA liabilities for all the outer loops. This is done by fitting a function on account value and different outer loops for a policy. This fitting can be done using Regression Splines. Previously, there have been attempts to perform similar operation using Polynomial Regression. However, in case of Polynomial regression, as we observed that the predictors impose a global structure on the response variable, it is not flexible enough to capture the dependency over the entire range. As a result, to avoid imposing such a global structure, Regression Spline is suggested by Lin and Yang in their paper.

To illustrate the reduction in inner and outer-loop projection, they use the following 2 generic but common VA policies:

(1) 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.

(2) 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.

Following that, two sets of nested simulation are ran for both the policies based on the regime switching lognormal model proposed by Hardy in 2001 using different sets of parameters to calculate the predictive distributions of their VA liabilities at t=1. Based on the simulation, the following observations were made: (1) the larger the number of inner-loops, the smoother the scatterplot. This is because running more loops reduces the estimation error. (2) the dependency between the predicted account value and the predicted liability is not linear and the pattern varies across policies.

To overcome this tradeoff between run time and accuracy, spline regression is used. In the context of VA, regression spline is fitted using account values and simulated liabilities. Practically, the liabilities do not need to be very accurate in order to fit the spline model and hence we can use less number of inner loops thereby enabling the simulation algorithm to run faster.

Now that we have seen the main idea, we will start investigating this method more comprehensively. Initially, the approach starts with an aim to reduce the number of outer loops. For a nested simulation algorithm with M outer-loops, each policy will have M predicted account value obtained through each outer-loop. At each account value, predicted liability is computed using multiple inner-loops. Now, by using spline regression, the running time may be reduced (because of smaller number of inner-loops). However, it may still be time consuming to simulate inner-loops for all outer-loops (this still is a case for a single policy which will be augmenting soon). To overcome this Lin and Yang propose an approach in their paper in which predicted liabilities are simulated only at some selected account values. Using those and their liabilities, a spline model is fit to estimate the liabilities at all outer-loops. Ideally this model should be close to a model fitted using the entire set of observations so that the estimated liabilities have higher accuracy.

To select the predicted account values, we use a clustering based method. In our context, let $x = (x_1, \ldots, x_M)$ be the predictive account values coming from a univariate distribution X. Let $Y = (Y_1, \ldots, Y_M)$ be the corresponding responses which in our case would be predictive liabilities of a policy at different outer-loops. Then relation between x and Y is modelled by:

$$Y_i = f(x_i) + \epsilon_i$$

where i=1,...,M. Then, to reduce running time we select a subset of predictive account values $x^* = (x_1^*, \ldots, x_m^*)$ where $m << M$. This helps in fitting spline model to a smaller amount of data. Assuming that the model we fit is $\hat{f}$, we want to minimize:

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

For setting the selection criteria, an upper bound is provided in the paper which depends on the selected predictors x^*. Based on the theorem shown in the paper, let $\{C_1, \ldots, C_m\}$ be a partition of x such that $x_j^* \in C_j$ for j = 1,...,m and each C_j contains M_j elements, $\sum_{j=1}^{m} M_j = M$. When $p \geq 3$, the following holds:

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

Now, as the partition becomes finer, the third term on right side (fourth order term) converges to 0 faster than the second term. Hence, we will ignore that and propose a method for selecting x^* that controls the first and second term. Futhermore, it is shown in the paper that asymptotically, the MSE of the derivatives of the penalized least square estimator does not depend on the predictor. As a result, the upper bound can always be written in a way that x^* only appears in the $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$. Based on this result, Lin and Yang present an approach to select x^* such that $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$ is minimized. Although this method is not optimal in minimizing the objective function, it is very simple to implement in different situations.

To implement this, find a vector which $\mathbf{x} = (x_1, \ldots, x_m)$ that minimizes $\sum_{j=1}^{m} \sum_{i=1}^{M_j} (x_i - x_j^*)^2$ using k-means clustering algorithm. This algorithm finds a partition whose objective is minimizing the within-cluster sum of squares.

Practically, running the k-means algorithm with the predicted account values for different policies can result in different sets of training data. This can be due to couple reasons: (1) k-means clustering algorithm may start at randomly initialized cluster centres and stop at local optimum hence giving varying results. (2) clustering depends on policy's predictive account values which are different among different policies. This is inefficient as nested simulation algorithm runs for different set of algorithms for different policies. To overcome that limitation, the paper selects the training set based on the underlying asset's return from different outer loops which are referred to as representative outer-loops. To better control the variability at the tails, the max and min returns are also included. By doing this, one gets the same set of training data for all the policies in the portfolio.

In the paper, using 1000 inner-loop simulations and 200 outer-loops out of 1000, spline curve is fitted using 10 B-spline basis functions with equidistant knots. To compare that, the predictive VA liabilities obtained from the 1000/10,000 simulation algorithm are used. 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.