MODELING METHODOLOGY

A primer on model efficiency techniques for valuation of large life insurance portfolios

Authors

Steven Morrison PhD
Moody's Analytics Research

Contact Us

Americas
+1.212.553.1658
clientservices@moodys.com

Europe
+44.20.7772.5454
clientservices.emea@moodys.com

Asia (Excluding Japan)
+85 2 2916 1121
clientservices.asia@moodys.co

Japan
+81 3 5408 4100
clientservices.japan@moodys.com

Overview

This paper provides an introduction to various techniques for efficient calculation of the market-consistent value of a portfolio of insurance policies. Two standard approaches to portfolio valuation are considered: (1) Use of different scenarios through different policies; (2) Portfolio compression through the use of model points. Additionally, the use of proxy functions is introduced as a novel approach to valuation of individual policies.

Techniques are illustrated and their performance compared using a case study based on a relatively simple portfolio of options.

The best performing method involves valuing all policies in the portfolio, but using different scenario sets for different policies. This method takes advantage of diversification of error across different policy valuations so that the overall portfolio value is much more accurate than the underlying policy values. This technique relies on being able to express the overall portfolio value as the sum over separate policy valuations. It's effectiveness for any particular portfolio will depend on the extent to which these underlying policies are similar.

Another technique that exploits the similarities between different policies involves grouping into a smaller number of model points. This technique performs well in the case study considered, but not as well as the method of valuing all policies using different scenarios sets for different policies. In grouping similar policies as model points we sacrifice some of the diversification benefit that can be achieved, and this loss in potential diversification benefit is not compensated by the reduction in the number of policies that are valued.

Finally, we considered the use of 'proxy functions', where Monte Carlo valuations are replaced by simple functions that can be evaluated rapidly. Monte Carlo valuations are only used to fit these proxy functions, with the number of 'fitting policies', and the number of 'fitting scenarios' per fitting policy, being factors that can be varied in order to achieve a satisfactory trade-off between accuracy and runtime. In the examples considered, the best performing combination of these factors involves valuing all policies in the portfolio with a small number of risk-neutral scenarios per policy. In this case, fitting the proxy function doesn't change the overall portfolio valuation, but does provide a more accurate valuation for the underlying policies.

Table of Contents

1. Overview

This paper provides an introduction to various techniques for efficient calculation of the market-consistent value of a portfolio of insurance liabilities. For a typical life insurance portfolio, this calculation involves a large amount of computational effort. In general, the size of the computational requirement scales with two different factors:

1. Number of policies

   Even if the accurate valuation of a single policy can be carried out relatively quickly, the valuation of a portfolio of such policies may be extremely time consuming simply due to the portfolio size.

2. Number of valuation scenarios

   Valuation of liabilities with fixed cash flows can be valued by projecting and discounting cash flows under a single scenario. For more general 'variable liabilities' containing embedded options and guarantees, this discounting calculation needs to be carried out many times under different risk-neutral stochastic scenarios, with the accuracy of the resulting estimated values scaling with the number of such scenarios used.

Various 'model efficiency' techniques have been proposed that attempt to reduce computation time, via a combination of reducing the number of policies and/or number of scenarios used in the calculation. Although these techniques vary in the details of their implementation, they are all motivated by the same basic philosophy: the values of the many different policies in a typical life insurance portfolio are related (due to the fact that different policies shares similar characteristics), and thus the 'brute force' valuation of all policies involves a degree of redundant calculation. Model efficiency techniques attempt to exploit the similarities between different policies in order to reduce the number of policies and/or risk-neutral scenarios used in the valuation calculation.

This paper provides an introduction to various model efficiency techniques, illustrated using a simple example portfolio.

The remainder of the paper is organized as follows:

- Section 2 describes the example portfolio.
- Section 3 describes 'brute force' valuation of the portfolio, and introduces a simple but effective model efficiency technique based on using different scenarios through different policies.
- Section 4 describes an alternative model efficiency technique, based on compressing the portfolio size by grouping similar policies into model points.
- Section 5 introduces a novel use of 'proxy functions' for the valuation of individual policies.
- Section 6 concludes by providing a comparison of the various techniques considered.

2. A simple example

To introduce the various techniques available, we will implement these using a relatively simple case study. The policies are chosen to have embedded options, so as to illustrate the additional challenges that these bring and to illustrate the effect of using different numbers of valuation scenarios. However, they are simple enough that they can be valued analytically (under a simple choice of valuation model). Such analytical solutions allow us to quickly assess the accuracy of valuations produced using more time-consuming simulation techniques.

We will also assume that all policies in the portfolio have the same characteristics, except for their maturity.

Note that neither of these simplifying assumptions (availability of analytical solutions, and relative homogeneity of the portfolio) are required in order to implement any of the techniques considered here, but both are extremely useful in developing intuition for how and why the techniques work, and to highlight the similarities and differences between them.

The portfolio considered contains 1,000 policies, with each policy paying the value of some underlying fund at some fixed maturity subject to a fixed guaranteed minimum payout. Policies are chosen to have the following characteristics:

- Same underlying fund across all policies; Current fund value = 1.
- Maturities in the range 1- 14 years[1]. All maturities are assumed to be an integer number of months[2].
- For any particular policy, the guaranteed minimum payout is equal to the current forward price of the fund corresponding to that policy's maturity.

The net asset-liability position is thus a portfolio of short positions in 'forward at-the-money' put options, which (under a Black-Scholes model for the underlying fund) can be valued analytically.

3. Valuation of all policies

Although the portfolio can be valued analytically in this case, we will use Monte Carlo simulation to perform the valuation (as we would normally use for liabilities with embedded options and guarantees). Risk-neutral scenarios for the underlying fund value are generated using a Black-Scholes model with risk-free rate = 2% and volatility = 10%. Scenarios are generated at monthly time-steps[3] and the resulting guarantee payouts calculated for each policy.

We are immediately faced with a choice here. Do we use a single set of scenarios to value all policies, or use different scenario sets for different policies? We will consider both choices below.

3.1 Common scenarios through all policies

As a starting point, we value the portfolio using a single common set of 10,000 risk-neutral scenarios for all policies. This will provide a benchmark for all other techniques to follow.

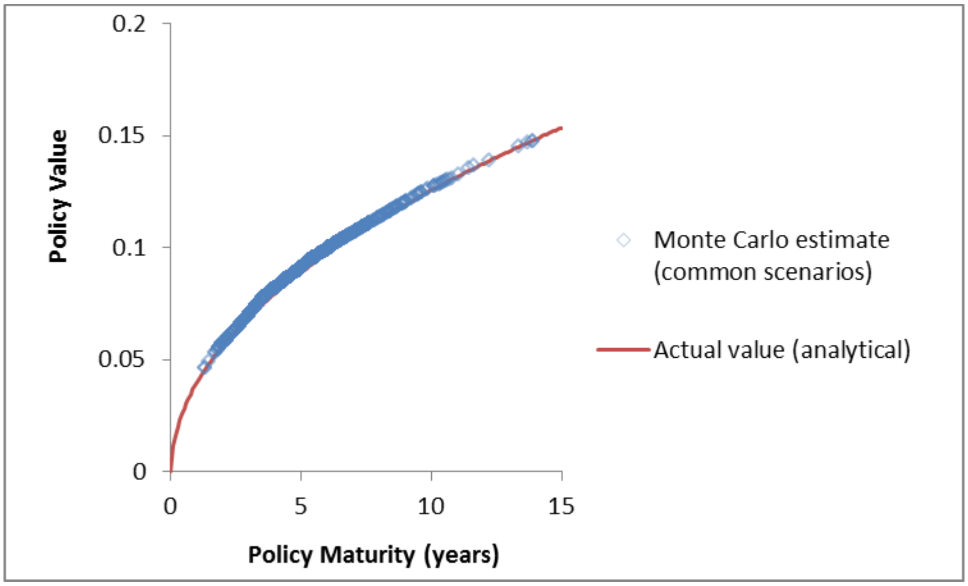
Figure 1 shows estimated policy values plotted against policy maturity. Monte Carlo estimates are compared to actual values calculated using analytical formulae.

[1] Maturities in years where sampled randomly from a lognormal distribution with mean maturity of 5 years and standard deviation of log(maturity) = 4.

[2] In practice, this means that for any given monthly maturity there are a number of identical policies.

[3] Risk-neutral scenarios are sampled in antithetic pairs.

Figure 1: Policy value vs policy maturity (estimated using common scenarios through all policies)

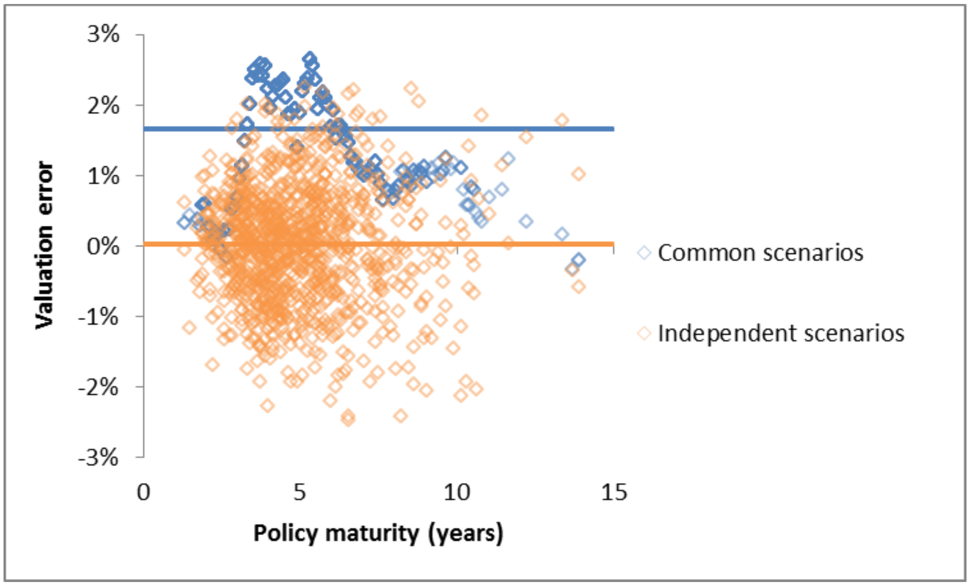


The total portfolio value is estimated as 88.67, compared to the actual value of 87.23. We overestimate the portfolio value by 1.7% in this case, due to the fact that a finite (albeit large) number of scenarios are used and therefore the estimate is subject to sampling error. In this case, we estimate that the standard error in the estimated value is 0.58, so this particular set of 10,000 scenarios results in an estimated value that is a 2.5 standard deviations higher than the true value. In this case, we seem to be slightly unlucky in our choice of random number seed. If we were to repeat this exercise many times using different random number streams, on average we expect the estimated value to exactly equal the actual value (i.e. this particular estimator is unbiased).

Although we are ultimately interested in the portfolio value, it is informative to look at the accuracy of estimated values on individual policies. Figure 2 plots the errors on individual policy valuations against policy maturity[4]. The errors produced using common scenarios through all policies are indicated by blue diamonds, with the solid blue line showing the average policy error (corresponding to the overall portfolio level error of 1.7%).

[4] Errors here are expressed relatively to the average policy value.

Figure 2: Policy valuation errors vs policy maturity



Although the errors on individual policies look reasonable, unfortunately they are almost all of the same sign (all positive, with the exception of 3 policies). There is a clear relationship between errors on different policies, since they were produced using the same paths for the same underlying fund.

2.1 Different scenarios through different policies

For comparison, the orange diamonds in Figure 2 show valuation errors when each policy valuation is carried out using a different, independent, set of 10,000 scenarios (rather than a common set across all policies). The size of errors on individual policies are clearly of the same order of magnitude as before[5], but there is no relationship between errors on different policies (since they are produced using independent scenario sets). Since estimates of the value of individual policies are independent, we benefit from diversification when these values are aggregated up to portfolio level. In this case, the average error i.e. overall portfolio error (shown by the solid orange line) is close to zero.

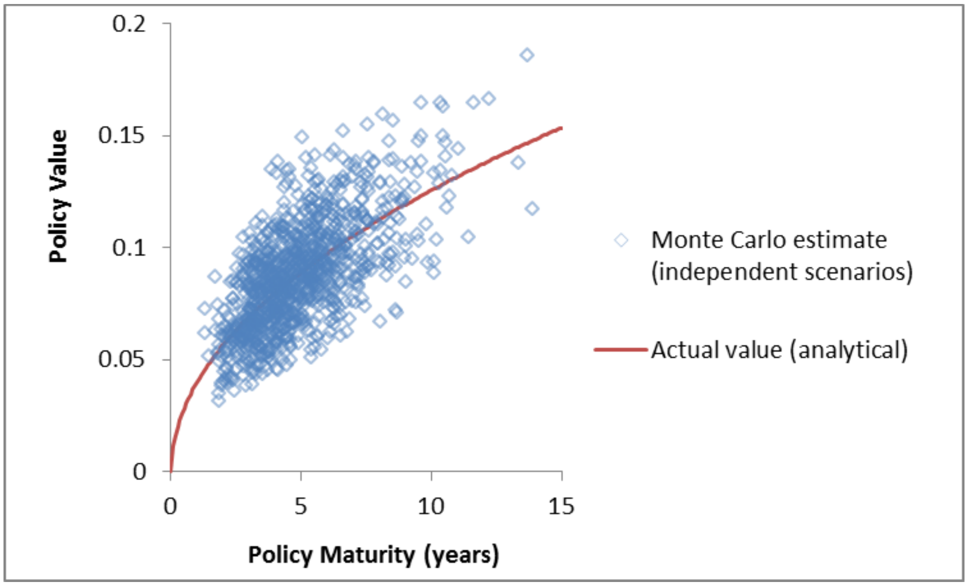
Using independent scenario sets, with 10,000 scenarios per policy, the standard error in the estimated portfolio value is estimated as just 0.02 (compared to 0.58 using 10,000 common scenarios across all policies). This corresponds to as variance reduction of $(0.58/0.02)^2$=636.

Put another way, we estimate that only 16 scenarios per policy are required (assuming independent scenario sets per policy) to give a comparable standard error to that using 10,000 common scenarios across all policies[6]. Figure 3 shows estimated policy values plotted against policy maturity in this case.

[5] Note that the errors using common scenarios for maturities around 4-6 years are relatively large (in the sense that they are at the upper end of the range of errors produced using independent scenarios), and this indicates again that we have been somewhat unlucky in the choice of random number seed here. It is also worth noting that this portfolio contains a relatively high proportion of policies in this range of maturities.

[6] 10,000 / 636 = 16 (rounded to the nearest integer).

Figure 3: Policy value vs policy maturity (estimated using independent scenarios through different policies)



The errors on any individual policy here are huge compared to the corresponding estimates in Figure 1. Each policy value here is estimated using only 16 scenarios, rather than 10,000. However, when we aggregate to portfolio level, the overall standard error in the portfolio value estimate is estimated as 0.57, approximately the same as achieved using 10,000 common scenarios through all policies.

In running only 16 scenarios, rather than 10,000 scenarios, through the asset/liability cash flow model, we clearly achieve a huge reduction in run-time for this particular part of the calculation. The estimated 'model efficiency' gain here is 10,000/16 = 625 (relative to the benchmark case of using common scenarios through all policies)[7]. Note that this number is based on counting the number of scenarios run through the cash flow model only and ignores the cost of generating the economic scenarios. In this case, the number of economic scenarios required is 1,000 x 16 = 16,000 rather than 10,000, so the ESG model run-time is actually somewhat longer (by a factor of 1.6) despite the fact that the cash flow model run-time is a lot shorter (by a factor of 625). In practice, the generation of economic scenarios is often computationally 'cheap' compared to the calculation of asset/liability cash flows, and the cost of scenario generation can usually be considered a second order factor in the overall computational cost.

4. Valuation using model points

So far, we have considered valuation of all policies, with varying numbers of risk-neutral scenarios used to value these. Using different scenario sets for different policies results (rather than common scenarios across all policies) results in huge gains in model efficiency in the sense that a far smaller number of risk-neutral scenarios are required to give a similar level of statistical accuracy. Intuitively, when we use common scenarios for all policies we fail to get full benefit from diversification of statistical errors across similar policies.

Recognizing the similarity between different policies, an alternative approach to model efficiency is to group highly similar policies together and represent the group by a single policy type for the purpose of valuation, thus reducing the number of different policies that need to be valued. The representative policies are commonly known as 'model points'.

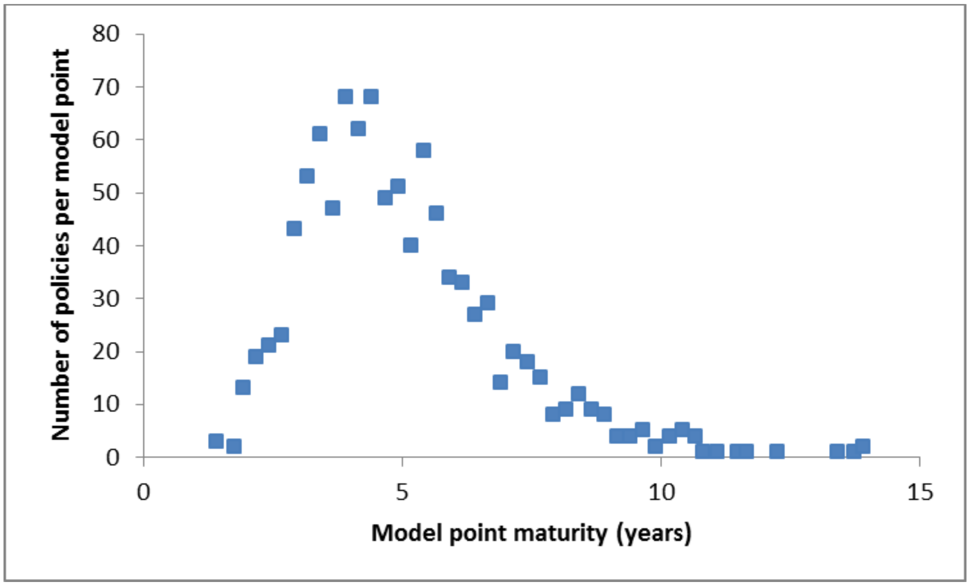
In the simple example considered here, the only distinguishing factor between different policies is their maturity. We therefore group policies with 'similar' maturities and construct representative model points per group, as follows:

- Two policies are in the same group if their maturities agree when rounded up to the nearest quarter.
- Within each group we create a representative model point with maturity equal to the average maturity of all policies in that group (and rounded to the nearest month).

[7] Slightly different to the previously calculated 'variance reduction' due to rounding.

This grouping exercise results in 46 distinct model points. Figure 4 shows the number of policies per model point, plotted against the maturity of the model point. For example, there are 68 policies in the original portfolio with maturities in the range 3 years 10 months to 4 years (inclusive), and the average maturity of this group of 68 policies is 3 years 11 months (rounded to the nearest month). Rather than valuing each of these 68 policies separately, we can approximate this calculation by a valuation of a single policy (of maturity 3 years 11 months) and simply multiply the valuation by 68 when we aggregate to calculate the portfolio value.

Figure 4: Number of policies per model point vs maturity of model point



As before, we can choose to run common scenarios through all model points, or different scenarios through different model points. Given previous results, we expect the latter option to be more efficient. However, it is instructive to consider the former option for completeness.

If we run 10,000 common scenarios through all 46 model points, the resulting portfolio value is estimated as 87.30, with a standard error of 0.58. Note that the standard error is similar to that obtained when 10,000 common scenarios were run through all 10,000 policies. This is intuitive – there is no diversification benefit in using common scenarios across all policies and so the overall accuracy is similar whether they are valued individually or grouped. The benefit of grouping here is not to reduce the required number of scenarios (per policy) but rather to reduce the total number of policies that need to be valued, in this case resulting in a model efficiency gain of 1,000/46 = 22.

As suggested by the results of the previous section, a better approach might be to use different scenarios for different model points thus allowing us to benefit from diversification of error across different model points. Indeed, in this case we estimate that only around 600 scenarios are required per model point in order to give a similar level of statistical accuracy in the overall portfolio value. By reducing both the number of policy valuations and the number of scenarios per policy, the overall model efficiency is now (10,000/600) x (1,000/46) = 362. This is a huge efficiency gain, but not as big as the gain achieved by running different scenario sets through all 1,000 policies – in that case, the model efficiency gain was (10,000/16) x (1,000/1,000) = 625. In grouping policies into model points here, we give up some of the potential diversification benefit that can be achieved by using different scenarios for different policies, and that loss in potential diversification benefit isn't compensated by the reduction in the number of policies.

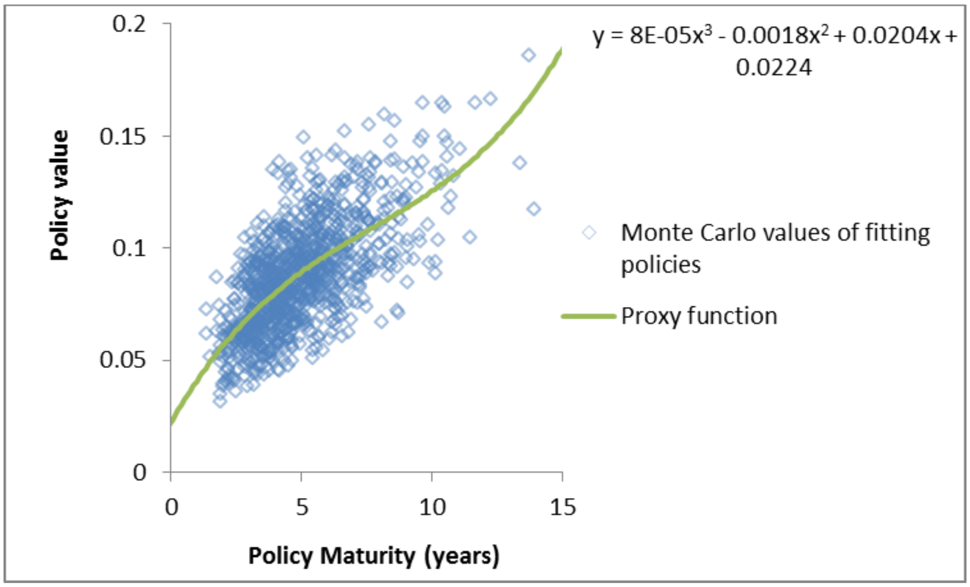
So far, our discussion of model accuracy has focused on the standard error in the estimated value. However, when we group policies into model points, another type of error is introduced. The portfolio of model points is only an approximation to the full portfolio, and as such its value will (in general) not agree exactly with the value of the full portfolio, even in the limit as the number of risk-neutral scenarios tends to infinity. The grouping of policies into model points also introduces *bias* in the valuation. In the example considered here, we can measure this bias exactly using the analytical formulae that are available: the portfolio of model points has an analytical value of 87.31, compared to the full portfolio value of 87.23. By grouping polices into model points, we introduce an upward bias of 0.08 (0.1% of the actual portfolio value), which in this case is relatively small compared to the statistical error introduced by using a finite set of scenarios (recall the standard error was around 0.58 in the examples considered).

5. Valuation using proxy functions

So far, we have assessed accuracy in terms of the overall portfolio valuation, rather than on the values of the underlying policies. Indeed, the 'best' performing method considered so far does a rather poor job of valuing individual policies, as indicated by Figure 3. This method works because the large errors on individual policies diversify well when aggregated over the portfolio.

This diversification of error can be exploited further by fitting a function through the noisy data to reveal the functional relationship between policy value and policy maturity. For example, Figure 5 shows a cubic function fitted through this data using least squares regression.

Figure 5: Fitted proxy function (1,000 actual policies as fitting policies, with 16 scenarios per policy)



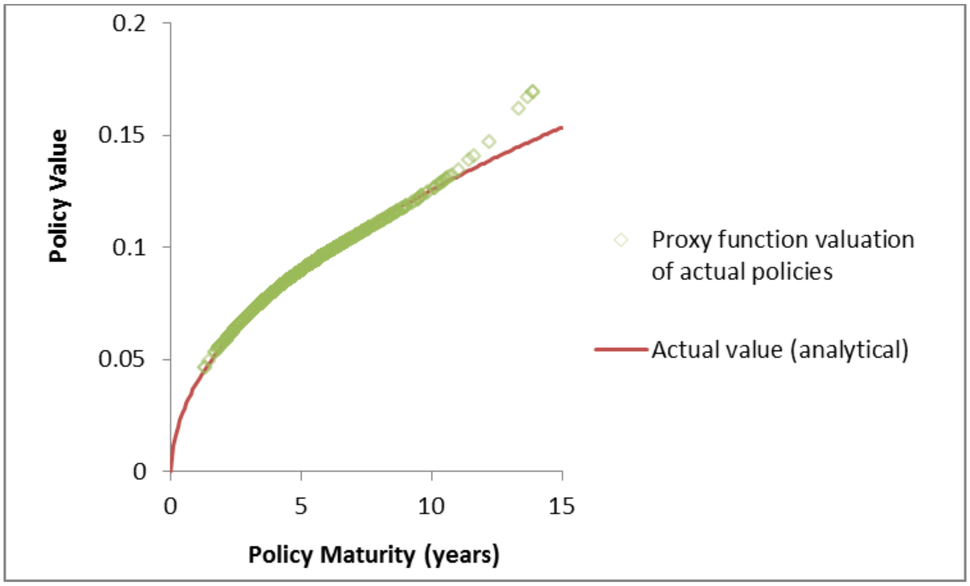
In this paper, we call this fitted function a 'proxy function' for the policy value - this function provides a proxy to the value of a policy as a function of its maturity.

Note that the term proxy function is more commonly used in life insurance modelling to refer to functions describing the value of a portfolio of policies as a function of underlying risk factors e.g. interest rates, equity levels, implied volatilities, etc. In recent years, such proxy functions have been successfully used to reduce computation time arising in various 'stochastic-on-stochastic' problems, where portfolios need to be valued many times as these risk factors change (in particular the calculation of 1-year VaR capital). In this paper, we use the term proxy function to described the value of individual policies as a function of their specific characteristics (in this case, the policy maturity) assuming that current (initial) risk factors are fixed, in order to perform a single portfolio valuation. More generally, we can think of proxy functions describing individual policies as a function of both risk factors *and* policy characteristics. This more general type of proxy function provides a promising solution for some of the most demanding computational problems arising in life insurance modelling, such as the multi-period projection of Variable Annuity Greeks for the purpose of quantifying hedge effectiveness[8].

Having fit the proxy function, we can use it to evaluate each of the 1,000 actual policies in the portfolio. These evaluations are compared to actual values in Figure 6.

[8] See "Proxy Methods for Hedge Projection: Two Variable Annuity Case Studies" (Moody's Analytics, May 2016).

Figure 6: Evaluated proxy function vs actual values (1,000 actual policies as fitting policies, with 16 scenarios per policy)



At a glance, the proxy function does a good job of approximating the value of most of the policies in the portfolio, but overestimates the value at higher policy maturities. The relatively poor performance of the proxy function at high maturities in this case can be explained by the fact that there is a relatively small amount of data use to fit the function in this region. Note that only 8 out of 1,000 policies have a maturity greater than 11 years.

Note that if we use the proxy function to value all policies and aggregate these to calculate the value of the whole portfolio, the value obtained is *exactly the same* as the portfolio value estimated using the original (noisy) policy level data that was used to fit the proxy function. Under a least squares regression, the average residual (difference between fitting data and fitted function) is zero. So, in this case, the use of the proxy function adds no accuracy to the valuation of the portfolio but does significantly improve the accuracy of individual policy valuations.

The training data used in the above example corresponds to actual policies, but other choices are possible. For example, Figures 7 and 8 show the same analysis performed using the 46 model points discussed in section 4 as training policies[9]. Recall that in this case the number of risk-neutral scenarios per model point is 600 (chosen such that the accuracy of the overall portfolio value is similar to that obtained using 16 scenarios through all 1,000 policies). The number of data points used to fit the proxy function is much lower here, but the accuracy of each point is much higher.

[9] We have assumed equal weighting on all model points in performing the least squares fit here (rather than weighting by the number of policies per model point for example).

Figure 7: Fitted proxy function (46 model points as fitting policies, with 600 scenarios per policy)

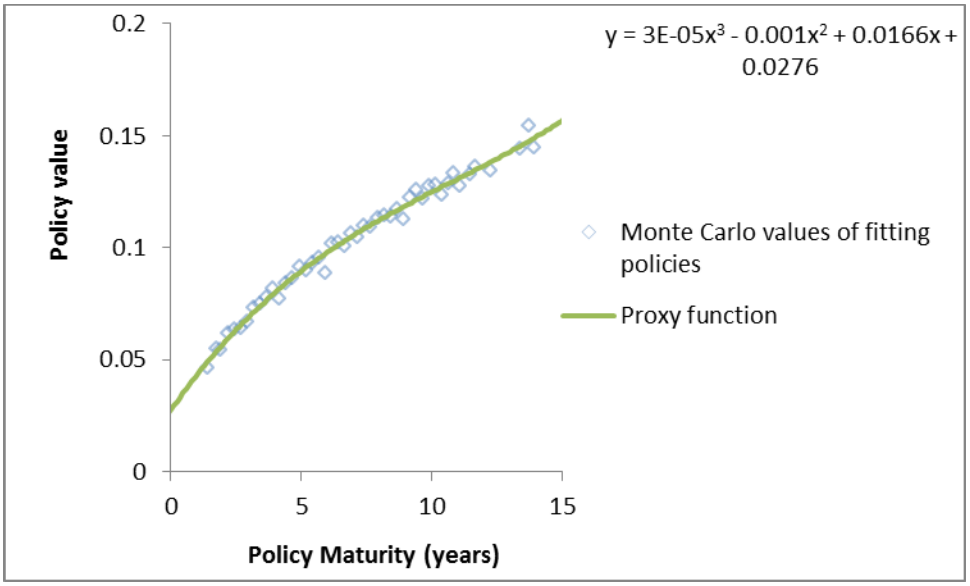
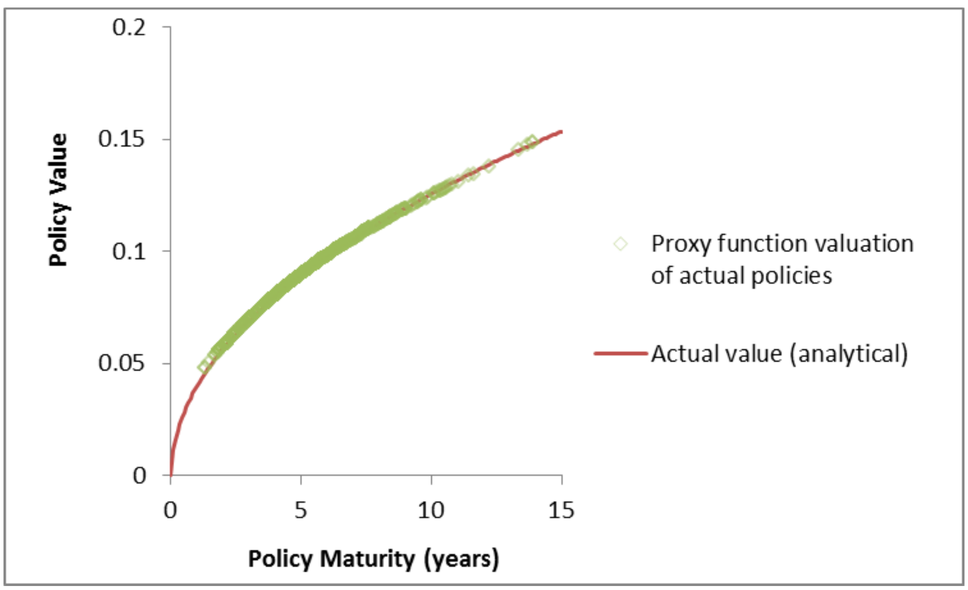


Figure 8: Evaluated proxy function vs actual values (46 model points as fitting policies, with 600 scenarios per policy)



Finally, Figures 9 and 10 show the same analysis performed using 46 policies with maturities chosen to uniformly span the range 0 to 15 years as training policies. Consistently with the model point example, the number of risk-neutral scenarios per fitting policy is 600. We have considerable freedom in choosing fitting policies. Indeed, fitting policies do not need to correspond to actual policies in the portfolio (nor do they have to be chosen to 'represent' the portfolio in the sense of model points).

Figure 9: Fitted proxy function (46 uniformly spaced policies as fitting policies, with 600 scenarios per policy)

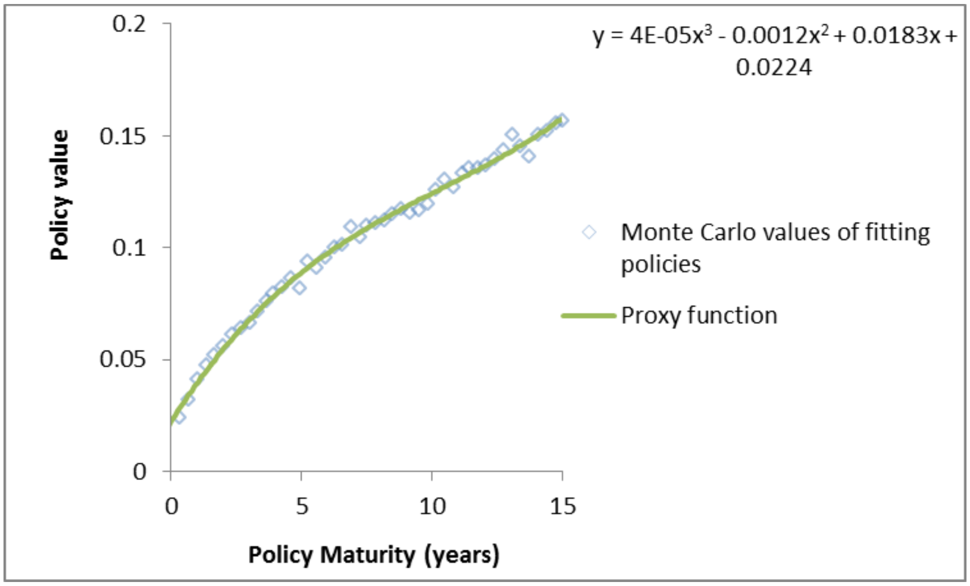
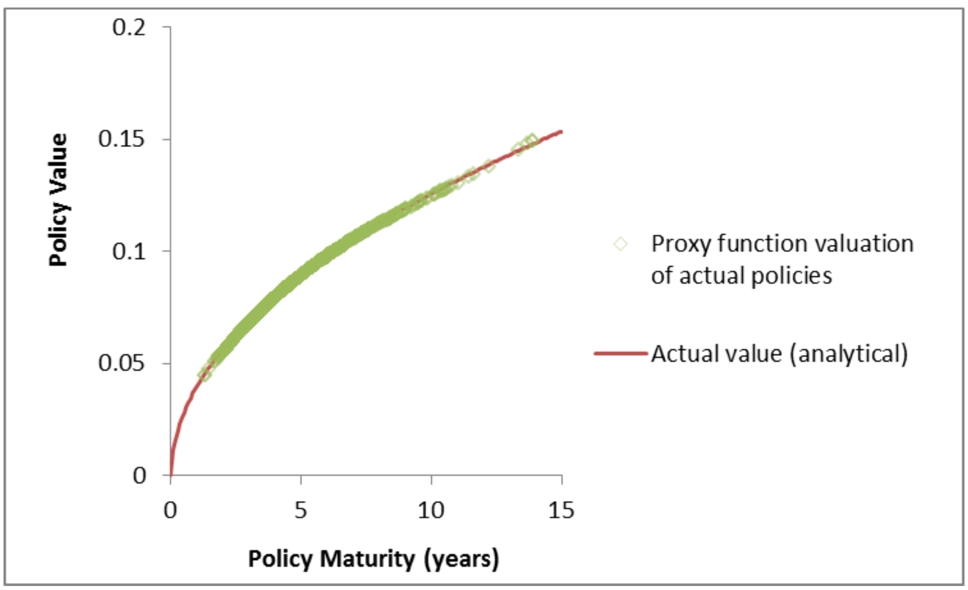


Figure 10: Evaluated proxy function vs actual values (46 uniformly spaced policies as fitting policies, with 600 scenarios per policy)



The analysis in this Section so far considers a single proxy function fit in each case i.e. a fit to one particular set of fitting policy values, produced using one particular random number stream. However, these fitting policy values are estimated using a relatively small number of risk-neutral scenarios (particularly in the case where actual policy values were used to fit the proxy function). As we generate different estimates (by using random number streams) the estimated values of fitting policies changes and hence so does the fitted proxy function. The proxy function fits are subject to statistical error and in order to quantify this, we have repeated these fits 1,000 times using different risk-neutral scenarios, and summarized the overall quality of fit using two statistics:

1. The *average* absolute error in the policy valuations. Distributions of this statistic are compared across different valuation methods in Figure 11.

2. The *maximum* absolute error in the policy valuations. Distributions of this statistic are compared across different valuation methods in Figure 12.

In both cases, error are expressed relative to the average policy value.

Figure 11: Distributions of average absolute policy valuation errors (1/5/25/50/75/95/99th percentiles and averages)

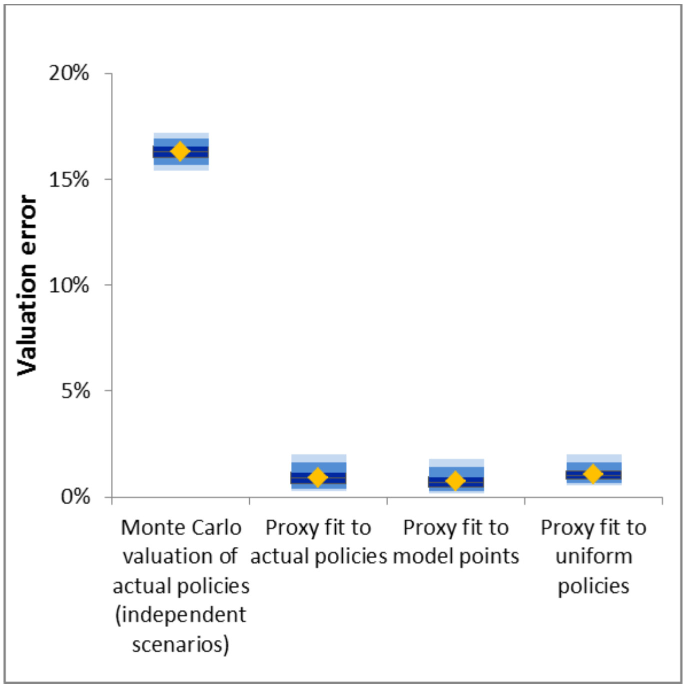
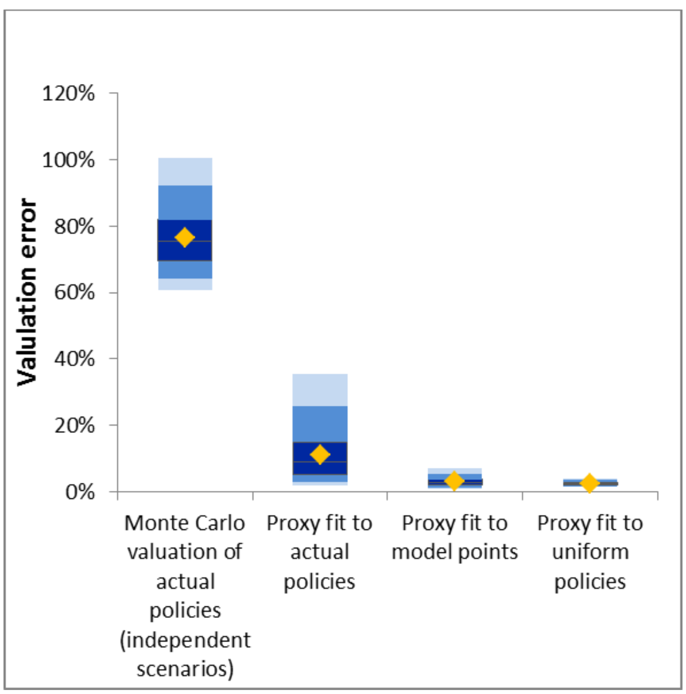


Figure 12: Distributions of maximum absolute policy valuation errors (1/5/25/50/75/95/99th percentiles and averages)



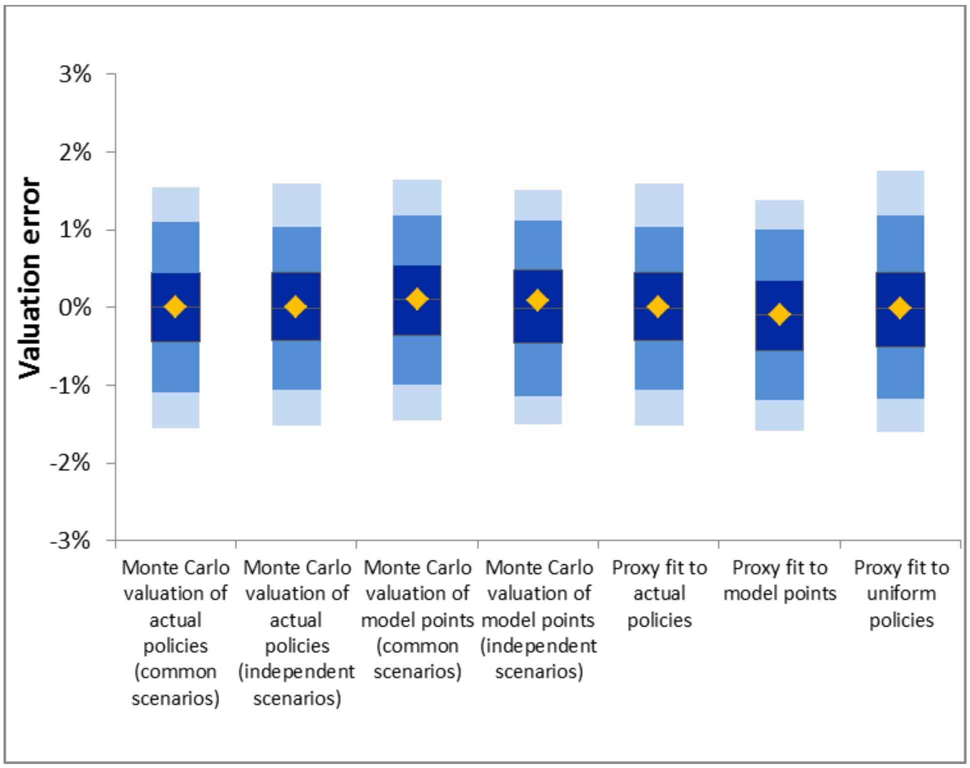
In summary, the fits to individual policies using Monte Carlo (with only 16 risk-neutral scenario per policy) are consistently poor, as expected given the relatively low number of risk-neutral scenarios used. By fitting a proxy function through these inaccurate valuations results we significantly improve the accuracy of the valuations *on average*, but very large errors can remain on individual policies – these largest errors tend to occur at the largest policy maturities, where less fitting data is available to control the quality of fit. By fitting to a smaller number of fitting policies (but more accurate valuations in each policy), the resulting proxy function is consistently accurate across all policies. Similar results are obtained whether we use model points or uniformly spaced policies to fit the proxy function – the average absolute error tends to be slightly higher if uniformly spaced policies are used, though the maximum absolute error appears to be more stable (and slightly lower on average).

6. Summary

This paper has described and compared various techniques for valuation of a portfolio of insurance liabilities containing embedded options and guarantees. These techniques exploit the relationship between different policies in order to improve the accuracy of estimated values.

Results for different techniques were compared using a simple case study. Each technique was configured to achieve a similar overall level of accuracy (see Figure 13 which shows the estimated distribution of valuation errors).

Figure 13: Distributions of portfolio valuation errors (1/5/25/50/75/95/99th percentiles and averages)



Given all techniques produce similar distributions for the portfolio value, we assess performance by measuring the computational effort required to achieve this level of accuracy. Table 1 compares the main factors affecting computational effort: the number of policies that are valued (using Monte Carlo simulation) and the number of risk-neutral scenarios used to value each policy. The product of these factors provides a measure of the computational cost of running the asset/liability cash flow model. The relative efficiency of different techniques is quantified by comparing this computational cost with the cost of the 'base case' (where all policies are valued using Monte Carlo simulation with a common set of scenarios used for all policies, which is the slowest of all methods explore here). For completeness, we also quantify the computational cost of generating all required risk-neutral scenarios.

Table 1: Comparison of computational requirements using different valuation methods

METHOD		NUMBER OF POLICIES VALUED USING SIMULATION	NUMBER OF RISK-NEUTRAL SCENARIOS PER POLICY	TOTAL NUMBER OF POLICY CASH FLOW EVALUATIONS	CASH FLOW MODEL EFFICIENCY GAIN (RELATIVE TO BASE CASE)	TOTAL NUMBER OF RISK-NEUTRAL SCENARIOS GENERATED	SCENARIO GENERATOR EFFICIENCY LOSS (RELATIVE TO BASE CASE)
VALUATION OF ALL POLICIES USING MONTE CARLO SIMULATION	Common scenarios across all policies	1,000	10,000	10,000,000	1	10,000	1
	Different scenarios for different policies	1,000	16	16,000	625	16,000	1.6
VALUATION OF MODEL POINTS USING MONTE CARLO SIMULATION	Common scenarios across all policies	46	10,000	460,000	22	10,000	1
	Different scenarios for different policies	46	600	27,600	362	27,600	2.76
VALUATION OF ALL POLICIES USING PROXY FUNCTIONS	Proxy function fitted to all policies	1,000	16	16,000	625	16,000	1.6
	Proxy function fitted to model points	46	600	27,600	362	27,600	2.76
	Proxy function fitted to uniformly picked policies	46	600	27,600	362	27,600	2.76

For the case study considered here, the best performing method involves valuing all policies in the portfolio, but using different scenario sets for different policies. This method takes advantage of diversification of error across different policy valuations so that the overall portfolio value is much more accurate than the underlying policy values. This technique relies on being able to express the overall portfolio value as the sum over separate policy valuations. It's effectiveness for any particular portfolio will depend on the extent to which these underlying policies are similar.

Another technique that exploits the similarities between different policies involves grouping into a smaller number of model points. This technique performs well in the case study considered, but not as well as the method of valuing all policies using different scenarios sets for different policies. In grouping similar policies as model points we sacrifice some of the diversification benefit that can be achieved, and this loss in potential diversification benefit is not compensated by the reduction in the number of policies that are valued.

Finally, we considered the use of 'proxy functions', where Monte Carlo valuations are replaced by simple functions that can be evaluated rapidly. Monte Carlo valuations are only used to fit these proxy functions, with the number of 'fitting policies', and the number of 'fitting scenarios' per fitting policy, being factors that can be varied in order to achieve a satisfactory trade-off between accuracy and runtime. In the examples considered, the best performing combination of these factors involves valuing all policies in the portfolio with a small number of risk-neutral scenarios per policy. In this case, fitting the proxy function doesn't change the overall portfolio valuation, but does provide a more accurate valuation for the underlying policies.