MODIFIED RECURSIONS FOR A CLASS OF COMPOUND DISTRIBUTIONS

KARL-HEINZ WALDMANN

Inst. f. Wirtschaftstheorie und Operations Research, Universität Karlsruhe, D-76128 Karlsruhe

ABSTRACT

Recursions are derived for a class of compound distributions having a claim frequency distribution of the well known (a,b)-type. The probability mass function on which the recursions are usually based is replaced by the distribution function in order to obtain increasing iterates. A monotone transformation is suggested to avoid an underflow in the initial stages of the iteration. The faster increase of the transformed iterates is diminished by use of a scaling function. Further, an adaptive weighting depending on the initial value and the increase of the iterates is derived. It enables us to manage an arbitrary large portfolio. Some numerical results are displayed demonstrating the efficiency of the different methods. The computation of the stop-loss premiums using these methods are indicated. Finally, related iteration schemes based on the cumulative distribution function are outlined.

KEYWORDS

Collective risk model, aggregate claims distribution, Panjer's algorithm.

1. INTRODUCTION

Compound distributions are used extensively in modeling the total amount of claims, X, in an insurance portfolio. Based on a claim frequency distribution satisfying the recursion

$$p_n = \left(a + \frac{b}{n}\right) p_{n-1}, \quad n \in \mathbf{N} := \{1, 2, \ldots\} \tag{1}$$

the probability mass function $g(x) = P(X = x), x \in \mathbf{N}$, is often evaluated recursively as

$$g(x) = \sum_{i=1}^{x} \left(a + b\frac{i}{x}\right) f(i) g(x - i) \tag{2}$$

starting with

$$g(0) = p_0,$$

where $f(i)$, $i \in \mathbf{N}$, denotes the probability mass function of the iid claim sizes Y_1, $Y_2, \ldots$.

ASTIN BULLETIN, Vol. 26, No. 2, 1996, pp. 213-224

Applying this well known recursion (see, e.g., Panjer and Willmot (1992), Sundt (1991) for details) to a portfolio with a large number of contracts, the initial value $g(0)$ is close to zero. This fact may cause an underflow (on a computer with standard software) followed by an abort or irregular running of the procedure. Panjer and Willmot (1986) (and Waldmann (1994, 1995) within the setting of an individual life model) suggest the use of a scaling function to stabilize the algorithm with respect to underflow/overflow. Moreover, Panjer and Wang (1993) study the stability of this type of recursion from a more theoretical point of view.

To overcome the problem of underflow in the initial and final stages of the iteration, we reformulate iteration scheme (2) with the probability mass function $g(x)$ replaced by the distribution function $G(x) = P(X \leq x)$. The resulting recursion has the nice property of producing increasing values lying within the unit interval. However, an underflow of the initial values is still possible. Therefore we transform $G(x)$ to $H(x) = [G(x) - G(0)]/G(0)$ avoiding an underflow in the initial stage of the algorithm. The stronger increase of the transformed values $H(x)$ may lead to an overflow in the final stage of the algorithm. This difficulty, however, can be partially managed by retransforming $H(x)$ to $G(x)$ for some $x_0 \in \mathbf{N}$ and continuing with the iteration scheme for $G(x)$. Moreover, the increase of the transformed values $H(x)$ can be diminished by use of a scaling function of type $\exp(-\alpha - \beta x)$ for suitable constants α and β. Scaling functions of this type considerably extend the range of applicability of the recursion but cannot avoid a breakdown by letting the expected number of claims tend to infinity. Therefore, we also present an adaptive transformation of $G(x)$, $x \in \mathbf{N}$, which enables us to manage an arbitrary large portfolio. The flexibility of the transformation results from its recursive definition depending on the initial value and the increase of the iterates. It is realized by dividing the range of $G(0)$, $G(1)$, ... into L layers and iterating in these layers successively. To make each layer representable on the computer, a scaling function is used, which is constant within a layer and suitably adapted by switching from layer ℓ to $\ell + 1$.

The paper is organized as follows. The iteration scheme is given in Section 2. Section 3 contains the transformed iteration schemes. Some numerical results are displayed in section 4 demonstrating the efficiency and applicability of the different methods. In Section 5 we extend our approach to a claim frequency distribution satisfying recursion (1) for $n = m + 1$, $m + 2$, ... and some $m \in \mathbf{N}$ only. The calculation of the stop-loss premiums using the methods of Sections 2 and 3 are indicated in section 6. Finally, Section 7 is devoted to a set of iteration schemes based on the cumulative distribution function $\hat{G}(x) := \sum_{i=0}^{x} G(i)$.

2. An iteration scheme for G(x)

In the following let an empty sum $\sum_{i=1}^{0} \ldots$ be defined to be zero. By slightly modifying a standard approach in deriving the iteration scheme for $g(x)$, $x \in \mathbf{N}$, we are in a position to obtain the following recursion for $G(x)$.

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

Theorem 1: $G(x)$, $x \in \mathbf{N}$, can be evaluated recursively as

$$xG(x) = r_1(x) + r_2(x). \tag{3}$$

where $G(0) = p_0$ and, for all $x \in \mathbf{N}$,

$$\begin{aligned} r_1(x) &= r_1(x-1) + G(x-1) \\ r_2(x) &= a\sum_{i=1}^{x-1} f(i) r_2(x-i) + (a+b)\sum_{i=1}^{x} i\, f(i) G(x-i) \end{aligned} \tag{4}$$

with $r_1(0) = 0$.

Proof. Introduce the generating functions $\varphi(z) = \sum_{x=0}^{\infty} g(x)z^x$, $\Phi(z) = \sum_{x=o}^{\infty} G(x)z^x$, and $\hat{\Phi}(z) = \sum_{x=0}^{\infty} \hat{G}(x)z^x$. Note that $\Phi(z) = \varphi(z)/(1-z)$, $\hat{\Phi}(z) = \Phi(z)/(1-z)$. Further, let $\Psi(z) = \sum_{x=0}^{\infty} f(x)z^x$ be the generating function of the claim size distribution.

To derive the recursion formula for $G(x)$ we start with the well known identity $\varphi(z) = \sum_{n=0}^{\infty} p_n \Psi(z)^n$, which can be rewritten as

$$(1-z)\Phi(z) = \sum_{n=0}^{\infty} p_n \Psi(z)^n$$

Differentiating both sides with respect to z we obtain

$$\begin{aligned} (1-z)\Phi'(z) - \Phi(z) &= \sum_{n=1}^{\infty} n p_n \Psi(z)^{n-1}\Psi'(z) \\ &= a\sum_{n=1}^{\infty} (n-1) p_{n-1} \Psi(z)^{n-1}\Psi'(z) \\ &\quad + (a+b)\sum_{n=1}^{\infty} p_{n-1}\Psi(z)^{n-1}\Psi'(z) \\ &= a\Psi(z)[(1-z)\Phi'(z) - \Phi(z)] + (a+b)\Psi'(z)(1-z)\Phi(z) \end{aligned}$$

Now, multiplying both sides by $z/(1-z)$, the last equation can also be written as

$$z\Phi'(z) = z\hat{\Phi}(z) + a\Psi(z)\left[z\Phi'(z) - z\hat{\Phi}(z)\right] + (a+b)z\Psi'(z)\Phi(z)$$

Finally, using $z\Phi'(z) = \sum_{x=1}^{\infty} xG(x)z^x$ and an analogous representation of $z\Psi'(z)$, a comparison of the coefficients of z^x, $x \in \mathbf{N}$, leads to the identity

$$xG(x) = r_1(x) + r_2(x)$$

where

$$r_1(x) = \sum_{j=0}^{x-1} G(j) = r_1(x-1) + G(x-1)$$

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

$$r_2(x) = a\sum_{i=1}^{x-1} f(i)\left[(x-i)G(x-i) - \sum_{j=0}^{x-i-1} G(j)\right] + (a+b)\sum_{i=1}^{x} i\,f(i)G(x-i)$$

$$= a\sum_{i=1}^{x-1} f(i)r_2(x-i) + (a+b)\sum_{i=1}^{x} i\,f(i)G(x-i) \tag{5}$$

(with $G(0) = p_0$, $r_1(0) = 0$). □

It easily follows from (3) that $r_1(x)$ and $r_2(x)$ are nonnegative and increasing functions of x. $r_1(x)$ can be implemented as a single number to be adapted at each step of iteration. Additionally, if both a and $a + b$ are nonnegative, which holds for the important cases of a Poisson counting distribution ($a = 0$, $b = \lambda$) and a negative binomial counting distribution ($a = p$, $b = p(\gamma - 1)$), $r_2(x)$ is the result of additions and multiplications of nonnegative real-valued numbers.

It is not necessary to recursively determine $r_2(x)$. By rearranging its defining terms in (5) we obtain

Corollary 1: $G(x)$, $x \in \mathbf{N}$, can be evaluated as in Theorem 1 with (4) replaced by

$$r_2(x) = -a\sum_{i=1}^{x-1} f(i)r_1(x-i) + \sum_{i=1}^{x}(ax+bi)f(i)G(x-i) \tag{4'}$$

Note, however, that the numbers to be added/multiplicated in (4′) are no longer nonnegative.

Looking at the binomial counting distribution ($a = -p/(1-p)$, $b = -(n+1)a$), (4) and (4′) have both positive and negative terms. The numerical results, however, which will be displayed in section 4 below give no hint for an instability with respect to rounding errors.

For a geometric counting distribution the recursion for $G(x)$ can already be found in Sundt (1991), p. 114.

Corollary 2: In case of a geometric counting distribution ($a = p$, $b = 0$) it holds that

$$G(x) = 1 - p + p\sum_{i=1}^{x} f(i)G(x-i) \tag{6}$$

with $G(0) = 1 - p$.

Proof. With (10.14) in Sundt (1991) and $r_1(x)$ as in Theorem 1 we infer by induction on x

$$G(x) - p\sum_{i=1}^{x} f(i)G(x-i) = \frac{1}{x}\left(r_1(x) - p\sum_{i=1}^{x} f(i)r_1(x-i)\right) = 1 - p,$$

which is the desired result. □

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

3. Stabilization of the algorithm with respect to underflow/overflow

The recursion for $G(x)$ has the nice property of being monotone, but the initial value p_0 may cause an underflow followed by an abort or irregular running of the procedure. Our first step in guaranteeing a regular running of the procedure is based on the following Theorem.

Theorem 2: The transformed values $H(x) = [G(x) - G(0)]/G(0)$, $x \in \mathbf{N}$, can be computed recursively via

$$xH(x) = h_1(x) + h_2(x), \tag{7}$$

where $H(0) = 0$, and, for $x \in \mathbf{N}$,

$$h_0(x) = h_0(x-1) + xf(x)$$
$$h_1(x) = h_1(x-1) + H(x-1)$$
$$h_2(x) = (a+b)h_0(x) + a\sum_{i=1}^{x-1} f(i)h_2(x-i) + (a+b)\sum_{i=1}^{x-1} i\,f(i)H(x-i)$$

with $h_0(0) = 0$ and $h_1(0) = 0$.

Proof. Set $h_0(0) = h_1(0) = h_2(0) = 0$. Then, together with Theorem 1, for $x \in \mathbf{N}$,

$$h_0(x) := \sum_{i=1}^{x} i\,f(i) = h_0(x-1) + xf(x)$$
$$h_1(x) := [r_1(x) - xG(0)]/G(0) = h_1(x-1) + [G(x-1) - G(0)]/G(0)$$
$$\begin{aligned} h_2(x) &:= x\,H(x) - h_1(x) \\ &= [x\,G(x) - x\,G(0) - r_1(x) + x\,G(0)]/G(0) \\ &= r_2(x)/G(0) \\ &= a\sum_{i=1}^{x-1} f(i)r_2(x-i)/G(0) + (a+b)\sum_{i=1}^{x} i\,f(i)[(G(x-i) - G(0)) + G(0)]/G(0) \\ &= a\sum_{i=1}^{x-1} f(i)h_2(x-i) + (a+b)\left[\sum_{i=1}^{x-1} i\,f(i)H(x-i) + h_0(x)\right] \end{aligned}$$

giving the desired recursion. □

The function $h_0(x)$ avoids that the sequence $H(x)$ degenerates to a sequence that has all its elements equal to zero. $h_0(x)$ is further a measure for the increase of the iterates. Since $H(x) \to (1 - p_0)/p_0$ for $x \to \infty$, it may be necessary to retransform $H(x)$ to $G(x)$ for some $x_0 \in \mathbf{N}$ and to continue with the recursive computation of $G(x)$, $x \geq x_0$.

Moreover, the increase of $H(x)$ can be diminished by weighting $H(x)$ by $\exp(-(\alpha + \beta x)$ for suitable parameters $\alpha \in \mathbf{R} := (-\infty, \infty)$, $\beta \geq 0$. The resulting recursion is given in the following Theorem.

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

Theorem 3: For $\alpha \in \mathbf{R}$, $\beta \geq 0$, the transformed values $\tilde{H}(x) = H(x)\mathrm{e}^{-(\alpha+\beta x)}$, $x \in \mathbf{N}$, can be evaluated recursively as

$$x\tilde{H}(x) = \tilde{h}_1(x) + \tilde{h}_2(x) \tag{8}$$

where $\tilde{H}(0) = 0$, and, for $x \in \mathbf{N}$,

$$h_0(x) = h_0(x-1) + x\,f(x)$$

$$\tilde{h}_1(x) = \mathrm{e}^{-\beta}\left[\tilde{h}_1(x-1) + \tilde{H}(x-1)\right]$$

$$\tilde{h}_2(x) = (a+b)\mathrm{e}^{-(\alpha+\beta x)}h_0(x) + a\sum_{i=1}^{x-1}\mathrm{e}^{-\beta i}f(i)\tilde{h}_2(x-i) + (a+b)\sum_{i=1}^{x-1}\mathrm{e}^{-\beta i}f(i)\tilde{H}(x-i)$$

with $h_0(x) = 0$ and $\tilde{h}_1(0) = 0$.

The parameter α of the scaling function $\exp(-\alpha - \beta x)$ gives a constant weight to $h_0(x)$ and can be used to reduce the order of $h_0(x)$. In addition, the parameter β can be utilized to diminish the increase of $h_0(x)$ and the resulting $\tilde{H}(x)$. The parameter β, however, is much more sensitive than α. If β is too large, isotonicity of $\tilde{H}(x)$ does no longer hold for all $x \in \mathbf{N}$. In such a case things may change and the transformation may lead to an earlier abort on account of an underflow.

The use of an exponential scaling function considerably extends the range of applicability of the recursion but cannot avoid a breakdown by letting the expected number of claims tend to infinity. We next present an adaptive transformation of $G(x)$, $x \in \mathbf{N}$, which enables us to manage an arbitrary large portfolio. The flexibility of the transformation results from its recursive definition depending on the initial value and the increase of the iterates. It is realized by dividing the range of $G(0)$, $G(1), \ldots$ into L layers and iterating in these layers successively. To make each layer representable on the computer, a scaling function is used, which is constant within a layer and suitably adapted by switching from layer ℓ to $\ell + 1$.

Let ω and Ω denote the smallest and greatest positive numbers, respectively, that can be represented on the computer using standard software. We interpret the interval $[\omega, \Omega]$ as the size of a layer. Further we introduce a subinterval $[10^{-t}, 10^{T}]$ of $[\omega, \Omega]$ for suitable constants t, $T > 0$. The interval $[10^{-t}, 10^{T}]$ is the region in a layer, in which the iteration is started (resp. restarted) and continued (up to some value greater than 10^{T}). Clearly, to avoid rounding errors, the set $[10^{-t}, 10^{T}]$ has to be chosen 'smaller' than $[\omega, \Omega]$.

In addition to t and T, the number L of layers depends on p_0. Set

$$c := -\log_{10} p_0$$

(i.e. $10^{-c} = p_0$). Then L can be chosen such that

$$t + (L-2)(T+t) < c \leq t + (L-1)(T+t)$$

holds. If $L = 1$, i.e. $p_0 \geq 10^{-t}$, there is no need for a transformation. Therefore assume $L > 1$. Finally, let ξ be the largest $x \in \mathbf{N}$ with $f(x) > 0$ (and ∞ if there is no such one).

The resulting transformed iterates $G^*(0)$, $G^*(1)$, ... can now be recursively defined as follows:

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

(a) *Layer 1.* Set

$$G^*(0) = 10^{-t}$$

and compute $G^*(1), G^*(2), \ldots$ up to some x_1, say, with $G^*(x_1) > 10^T$ according to

$$xG^*(x) = r_1^*(x) + r_2^*(x), \tag{9}$$

where

$$r_1^*(x) = r_1^*(x-1) + G^*(x-1)$$
$$r_2^*(x) = a\sum_{i=1}^{x-1} f(i) r_2^*(x-i) + (a+b)\sum_{i=1}^{x} i\, f(i) G^*(x-i)$$

with $r_1^*(0) = 0$.

(b) *Layer ℓ.* $(2 \le \ell \le L-1)$. Reset

$$G^*(x) = 10^{-(T+t)} G^*(x)$$
$$r_\nu^*(x) = 10^{-(T+t)} r_\nu^*(x), \quad \nu \in \{1, 2\}$$

for all $x_{\ell-1} - \xi < x \le x_{\ell-1}$ (with $G^*(x)$, $r_1^*(x)$, $r_2^*(x)$ equal to zero if they are less than ω) and compute $G^*(x_{\ell-1}+1), G^*(x_{\ell-1}+2), \ldots$ up to some x_ℓ, say, with $G^*(x_\ell) > 10^T$ according to (9).

(c) *Layer L.* Set $\gamma = c - t - (L-2)(T+t)$. Reset

$$G^*(x) = 10^{-\gamma} G^*(x)$$
$$r_\nu^*(x) = 10^{-\gamma} r_\nu^*(x), \quad \nu \in \{1,2\}$$

for all $x_{L-1} - \xi < x \le x_{L-1}$ and compute $G^*(x_{L-1}+1)$, $G^*(x_{L-1}+2)$, ... according to (9).

Summarizing the steps of iteration we immediately obtain

Theorem 4: Let $L > 1$. Evaluate $G^*(0)$, $G^*(1)$, ... as above. Then, for all $x > x_{L-1} - \xi$,

$$G(x) = G^*(x)$$
$$r_\nu(x) = r_\nu^*(x), \quad \nu \in \{1,2\}.$$

Further, for all $s = 1, \ldots, L-1$, and all $x_{L-s-1} - \xi < x \le x_{L-s} - \xi$

$$G(x) = 10^{-\gamma-(s-1)(T+t)} G^*(x)$$
$$r_\nu(x) = 10^{-\gamma-(s-1)(T+t)} r_\nu^*(x), \quad \nu \in \{1,2\}$$

(with $x_0 = 0$ and $G(x') = 0$ for $x' < 0$).

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

4. Numerical results and discussion

Our numerical results are ascertained with a computer program written in Turbo Pascal 5.0. We used real-valued variables of type 'extended' having a range from $1.9 * 10^{-4951}$ to $1.1 * 10^{4932}$. Thus $\omega = 1.9 * 10^{-4951}$ and $\Omega = 1.1 * 10^{4932}$.

We consider as a starting point the portfolio of $\bar{n} = 31$ independent life insurance policies discussed in Gerber (1979), p. 53. Each policy is supposed to have an amount at risk $i \in I = 1, \ldots, 5$ and a mortality rate $\bar{q}_j$ with $j \in J = 1, \ldots, 4$. Further $\bar{n}_{ij}$ denotes the number of all policies with amount at risk i and mortality rate $\bar{q}_j$. Note that the expected number of claims is 1.4.

This individual life model is approximated by a compound Poisson model with $a = 0$, $b = \bar{n}\lambda = \sum_{i \in I} \sum_{j \in J} \bar{n}_{ij} \bar{q}_j$ and $f(i) = \sum_{j \in J} \bar{n}_{ij} \bar{q}_j / \bar{n}\lambda$, $i \in I$, and by a compound binomial model with $a = -\lambda/(1 - \lambda)$, $b = (\bar{n} + 1)\lambda/(1 - \lambda)$, and $f(i)$ as in the compound Poisson model (cf., e.g., Kuon, Radke, and Reich (1993)).

Since the portfolio consists of 31 policies only, there is no need for a stabilization of (3) with respect to underflow. We therefore expand the portfolio by considering $k\bar{n}_{ij}$ policies in place of $\bar{n}_{ij}$ (for all $i \in I$ and $j \in J$).

The recursions (7) and (8) on which our transformed iterates $H(x)$ and $\tilde{H}(x)$ are based have been carried out up to some $x_0 \in \mathbf{N}$ guaranteeing $G(x_0) > 10^{-4000}$. For $x > x_0$, after retransforming the relevant data, recursion (3) has been applied to determine $G(x)$ directly. Moreover, we have distinguished between recursions (4) and (4′) when calculating $r_2(x)$. Based on (4) and (4′) also the transformed iterates $H(x)$ and $\tilde{H}(x)$ have been studied separately.

We say that a recursion is stable if the algorithm does not stop with an under flow or overflow and that both $| E'(X) - E''(X) | / E''(X) \leq 10^{-5}$ and $| \mathrm{Var}'(X)^{1/2} - \mathrm{Var}''(X)^{1/2} | / \mathrm{Var}''(X)^{1/2} \leq 10^{-5}$ hold, where $E'(X)$ and $\mathrm{Var}'(X)$ are determined with the help of the probability mass function of X and $E''(X)$, $\mathrm{Var}''(X)$ result from the moments of the counting distribution and claim size distribution together with the properties of expectation and variance. The maximal k and the associated number of policies we have obtained in this way are displayed in Table 2. Although the recursions for $G(x)$ and $H(x)$ nearly work within the same range of k, there is an essential difference. By increasing k, the recursion for $G(x)$ aborts with an underflow resulting from the initial value $G(0)$, the one for $H(x)$ starts with stable initial values and aborts with an overflow. The reduction of the

TABLE 1
Gerber's sample portfolio of 31 policies

Mortality Rate	Amount at Risk				
	1	2	3	4	5
0.03	2	3	1	2	–
0.04	–	1	2	2	1
0.05	–	2	4	2	2
0.06	–	2	2	2	1

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

TABLE 2
STABILITY OF THE ITERATION SCHEMES (3), (7) AND (8)

recursion	compound Poisson model		compound binomial model	
	maximal k	number of policies	maximal k	number of policies
$G(x)$	8000	248 000	7000	217 000
$H(x)$	8000	248 000	7000	217 000
$\tilde{H}(x)$, $\alpha = 10000$, $\beta = 0$	15 000	465 000	14 000	434 000
$\tilde{H}(x)$, $\alpha = 10000$, $\beta = 0.5$	65 000	2 015 000	62 000	1 922 000

increase of $H(x)$ as realized by use of $\tilde{H}(x)$ then gives for both the compound Poisson model and the compound binomial model (independent of the use of (4) or (4′)) stable solutions for a portfolio with nearly 2 million contracts.

We already mentioned that an L layer model can be applied to an arbitrary large portfolio. To give some insight into the increase of the number L of layers when the number of contracts is increased, we have used the interval $[10^{-t}, 10^{T}] = [10^{-4000}, 10^{+4000}]$ for carrying out the iterations.

For $k = 10^4, 10^5, 10^6$ (which corresponds to $3.1 \cdot 10^5, 3.1 \cdot 10^6, 3.1 \cdot 10^7$ contracts) the number L of layers needed is displayed in Table 3.

5. MODIFICATION OF THE CLAIM NUMBER DISTRIBUTION

The class of counting distributions can be extended by supposing the recursion

$$p_n = \left(a + \frac{b}{n}\right) p_{n-1}, \quad n = m+1,\ m+2,\ \ldots \tag{10}$$

to hold for some $m \in \mathbf{N}_0 := 0, 1, 2, \ldots$ only. In this more general situation the iteration scheme for the probability mass function of X, $g(x)$, reads (cf., e.g., Panjer and Willmot (1992), Corollary 6.16.1)

TABLE 3
THE NUMBER L of layers

k	claim frequency distribution	
	Poisson	binomial
10^4	$L = 1$	$L = 1$
10^5	$L = 8$	$L = 8$
10^6	$L = 77$	$L = 78$

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

$$g(x) = \frac{1}{x}\left\{\sum_{i=1}^{x}(a+bi)f(i)g(x-i)\right\} + \sum_{n=1}^{m} q_n f^{n*}(x),$$

where f^{n*} denotes the n-fold convolution of f with itself and where q_n is defined by

$$q_n := p_n - \left(a + \frac{b}{n}\right)p_{n-1}, \quad n = 1, \ldots, m.$$

Essentially the same arguments as in the proof of Theorem 1 give

Theorem 5: $G(x)$, $x \in \mathbf{N}$, can be evaluated recursively as

$$xG(x) = r_1(x) + r_2(x),$$

where

$$r_1(x) = r_1(x-1) + G(x-1) + \sum_{n=1}^{m} q_n\, x\, f^{n*}(x), \quad x \in \mathbf{N},$$

(with $G(0) = p_0$, $r_1(0) = 0$) and $r_2(x)$ as in (4)).

Being interested in extending Theorem 2, we only have to replace the recursion for $h_1(x)$ by $h_1(x) = h_1(x-1) + H(x-1) + \sum_{n=1}^{m}(q_n/p_0)\, x\, f^{n*}(x)$. Analogously, in Theorem 3, we have to redefine $\tilde{h}_1(x)$ by $\tilde{h}_1(x) = \mathrm{e}^{-\beta}[\tilde{h}_1(x-1) + \tilde{H}(x-1)] + \mathrm{e}^{-(\alpha+\beta x)}\sum_{n=1}^{m}(q_n/p_0)\, x\, f^{n*}(x)$. The L-layer approach does not work in case of the claim frequency distribution (10).

6. Evaluation of the stop-loss premiums

Let us begin with the claim frequency distribution (1). It is well known that the stop-loss premium $\mathrm{SL}(\tau)$, $\mathrm{SL}(\tau) := \sum_{x=\tau+1}^{\infty}(x-\tau)g(x)$, with retention $\tau \in \mathbf{N}$ can be written as

$$SL(\tau) = E(X) - \tau + \hat{G}(\tau - 1), \quad \tau \in \mathbf{N}. \tag{11}$$

Using Theorem 1 to determine $G(x)$, then $r_1(x) = \hat{G}(x-1)$ is obtained as a byproduct. Thus the results of Sections 2 and 3 can also be utilized to compute the stop-loss premiums for specified retentions. In particular, using Theorem 1, $\mathrm{SL}(\tau) = E(X) - \tau + r_1(\tau)$. Using the transformed iterates $H(x)$ and $\tilde{H}(x)$, $r_1(x)$ follows from $r_1(x) = (h_1(x) + x)p_0$ and $r_1(x) = [\mathrm{e}^{\alpha+\beta x}\tilde{h}_1(x) + x]p_0$, respectively. Applying the L-layer method, $r_1(x)$ results from $r_1^*(x)$ and is given explicitly in Theorem 4.

In case of the more general claim frequency distribution (10), $r_1(x) = \hat{G}(x-1) + \sum_{n=1}^{m} q_n \sum_{i=1}^{x} i\, f^{n*}(i)$. The transformed iterates $H(x)$ and $\tilde{H}(x)$ (with the recursions for $h_1(x)$ and $\tilde{h}_1(x)$ as defined in Section 5) give $r_1(x)$ as in the case of the claim frequency distribution (1). The L-layer approach, however, does not work.

https://doi.org/10.2143/AST.26.2.563220 Published online by Cambridge University Press

7. Iteration schemes based on $\hat{G}(x)$

The iteration schemes which will be presented in this section are based on the cumulative distribution function $\hat{G}(x)$. Forming the first and second differences of $\hat{G}(x)$, $\Delta\hat{G}(x) := \hat{G}(x+1) - \hat{G}(x) = G(x+1)$ and $\Delta^2\hat{G}(x) := \Delta\hat{G}(x+1) - \Delta\hat{G}(x) = g(x+2)$, we immediately obtain the distribution function $G(x)$ and the probability mass function $g(x)$, respectively.

Note that $\hat{G}(x)$ has the nice property of being an increasing and convex function. In case of the claim frequency distribution (1) the recursion reads

Theorem 6: Let $m = 0$. Then $\hat{G}(x)$, $x \in \mathbf{N}$, can he evaluated recursively as

$$x\hat{G}(x) = \hat{r}_1(x) + \hat{r}_2(x), \tag{12}$$

where $\hat{G}(0) = p_0$ and, for all $x \in \mathbf{N}$,

$$\hat{r}_1(x) = \hat{r}_1(x-1) + 2\hat{G}(x-1)$$

$$\hat{r}_2(x) = a\sum_{i=1}^{x-1} f(i)\hat{r}_2(x-i) + (a+b)\sum_{i=1}^{x} i\, f(i)\hat{G}(x-i)$$

with $\hat{r}_1(0) = 0$.

Proof. Starting with the identity (cf. proof of Theorem 1)

$$(1-z)^2\hat{\Phi}(z) = \sum_{n=0}^{\infty} p_n \Psi(z)^n$$

similar arguments as in the proof of Theorem 1 give the desired recursion. □

Since (3) and (12) (formally) differ in a factor 2 only, the methods of Section 3 can be adapted easily. Using (11) also the stop-loss premiums follow immediately by retransforming the transformed iterates $\hat{H}(x)$, $\tilde{H}(x)$ and $\hat{G}^*(x)$, say, to $\hat{G}(x)$.

In case of the more general claim frequency distribution (10) the iterates $\hat{G}(x)$ and its transformed versions can be obtained in a straightforward manner following the approach given in section 5.

References

Gerber, H.U. (1979) An Introduction to Mathematical Risk Theory. Huebner Foundation Monograph 8, Philadelphia.

Kuon, S., Radke, A. and Reich, A. (1993) An Appropriate Way to Switch from the Individual Risk Model to the Collective One. *ASTIN Bulletin* **23**, 23-54.

Panjer, H.H. and Wang, S. (1993) On the stability of recursive formulas. *ASTIN Bulletin* **23**, 227-258.

Panjer, H.H. and Willmot, G.E. (1986) Computational aspects of recursive evaluation of compound distributions. *Insurance: Mathematics and Economics* **5**, 113-116.

Panjer, H.H. and Willmot, G.E. (1992) Insurance Risk Models. Society of Actuaries, Schaumburg, IL.

Sundt, B. (1991) An Introduction to Non-Life Insurance Mathematics (2. Auflage). Verlag Versicherungswirtschaft, Karlsruhe.

WALDMANN, K.-H. (1994) On the exact calculation of the aggregate claims distribution in the individual life model. *ASTIN Bulletin* **24**, 89-96.

WALDMANN, K.-H. (1995) Exact calculation of the aggregate claims distribution in the individual life model by use of an *n*-layer model. *Blätter der DGVM, Band XXII,* 279-287.