Year : 2010

ON THE EFFICIENCY OF RECURSIVE EVALUATIONS WITH APPLICATIONS TO RISK THEORY

Sébastien VIQUERAT

UNIVERSITE DE LAUSANNE
FACULTE DES HAUTES ETUDES COMMERCIALES

ON THE EFFICIENCY OF RECURSIVE EVALUATIONS WITH APPLICATIONS TO RISK THEORY

THESE

Présentée à la Faculté des HEC
de l'Université de Lausanne

par

Sébastien VIQUERAT

Licencié en sciences économiques
mention sciences actuarielles
de l'Université de Lausanne

Pour l'obtention du grade de
Docteur en Sciences Actuarielles

2010

UNIL | Université de Lausanne
HEC Lausanne
Le Doyen
Bâtiment Internef
CH-1015 Lausanne

IMPRIMATUR

Sans se prononcer sur les opinions de l'auteur, la Faculté des hautes études commerciales de l'Université de Lausanne autorise l'impression de la thèse de Monsieur Sébastien VIQUERAT, licencié en sciences économiques, mention sciences actuarielles, de l'Université de Lausanne, en vue de l'obtention du grade de docteur en Sciences Actuarielles.

La thèse est intitulée :

ON THE EFFICIENCY OF RECURSIVE EVALUATIONS WITH APPLICATIONS TO RISK THEORY

Lausanne, le 22 juillet 2010

Le doyen

Daniel Oyon

Tél. ++41 21 692 33 66 | Fax ++41 21 692 34 35 | hecdoctorats@unil.ch

Jury de thèse

Professeur François Dufresne
Faculté des Hautes Études Commerciales
Université de Lausanne
Directeur de thèse

Professeur Hansjörg Albrecher
Faculté des Hautes Études Commerciales
Université de Lausanne
Expert interne

Professeur Stéphane Loisel
Institut de Science Financière et d'Assurances (ISFA)
Université Claude Bernard Lyon 1
Expert externe

Université de Lausanne
Faculté des Hautes Etudes Commerciales

Doctorat en Sciences Actuarielles

Par la présente, je certifie avoir examiné la thèse de doctorat de

Sébastien VIQUERAT

Sa thèse remplit les exigences liées à un travail de doctorat.
Toutes les révisions que les membres du jury et le-la soussigné-e ont demandées durant le colloque de thèse ont été prises en considération et reçoivent ici mon approbation.

Signature : F. Dufresne Date : 13 juillet 2010

Prof. François DUFRESNE
Directeur de thèse

Université de Lausanne
Faculté des Hautes Etudes Commerciales

Doctorat en Sciences Actuarielles

Par la présente, je certifie avoir examiné la thèse de doctorat de

Sébastien VIQUERAT

Sa thèse remplit les exigences liées à un travail de doctorat.
Toutes les révisions que les membres du jury et le-la soussigné-e ont demandées durant le colloque de thèse ont été prises en considération et reçoivent ici mon approbation

Signature : _____ Date : 13 juillet 2010

Prof. Hansjörg ALBRECHER
Membre interne du jury

Université de Lausanne
Faculté des Hautes Etudes Commerciales

Doctorat en Sciences Actuarielles

Par la présente, je certifie avoir examiné la thèse de doctorat de

Sébastien VIQUERAT

Sa thèse remplit les exigences liées à un travail de doctorat.
Toutes les révisions que les membres du jury et le-la soussigné-e ont demandées durant le colloque de thèse ont été prises en considération et reçoivent ici mon approbation

Signature :  Date : ____16/07/2010____

Prof. Stéphane LOISEL
Membre externe du jury

Avant-propos

Tout d'abord, j'aimerais remercier Monsieur le Professeur François Dufresne sans qui cette thèse n'aurait pas existé. J'ai vraiment eu beaucoup de plaisir à travailler avec lui durant ces cinq dernières années. J'aimerais également remercier Messieurs les Professeurs Hansjörg Albrecher et Stéphane Loisel d'avoir accepté de faire partie de mon jury de thèse.

Durant ces cinq années en tant qu'assistant diplômé, j'ai eu la chance de travailler dans une ambiance formidable et je profite ici pour remercier tous les professeurs, chargés de cours, chercheurs, assistants et secrétaire du Département de Science Actuarielles pour tous les bons moments passés ensemble que se soit au travail ou en dehors du travail. Je crois qu'il est rare de trouver une équipe de travail aussi sympa et souriante. Je remercie également les secrétaires du troisième étage pour leur agréable collaboration et leur bonne humeur ainsi que les dames de la cafétéria de l'Internef pour la pratique quotidienne du portugais.

Finalement, je tiens à remercier mes parents et ma famille pour tout le soutien apporté tout au long de mes études.

Contents

List of Figures

List of Tables

Synthesis report

This thesis consists of three essays on recursive evaluations related to the distribution of the aggregate claims amount of a portfolio of insurance policies over a period of time. Each essay corresponds to a chapter. An introductory chapter precedes these essays. In the actuarial literature, we can find several ways to model the distribution of the aggregate claims amount of a portfolio of insurance policies over a period of time. The collective and individual risk models are the most frequently used in actuarial applications.

In the former model, the aggregate claims amount random variable is defined as $S = X_1 + X_2 + \cdots + X_N$, where N and X_i represent the number of claims and the amount of the i-th claim, respectively. It is generally assumed that the X_i's are independent, identically distributed and independent of N. Recursive formulas for some particular distributions of N have been developed by Panjer (1981) and their stability against round-off errors is discussed in Panjer and Wang (1993). Recursive evaluations are useful in practice since they reduce the number of operations, which gives a faster evaluation. However, numerical problems may arise and lead to meaningless results due to the propagation of round-off errors coming from the representation of real numbers by floating-point numbers.

In the individual risk model, the aggregate claims amount random variable is defined as $S = X_1 + X_2 + \cdots + X_m$, where m is the number of policies in the portfolio and X_i is the claim amount of the policy number i. For this model, there exist several exact and approximative recursive evaluations (see e.g. De Pril (1986b, 1988, 1989), Dhaene and Vandebroek (1995), Hipp (1985, 1986), Kornya (1983) and Waldmann (1994)). An extension of this model can be used for the computation of the probability function of the m-fold convolution of a probability function.

The De Pril transform is a useful function for evaluations in relation to the distribution of a sum of independent random variables since the De Pril transform of a convolution of probability functions is the sum of the De Pril transforms of the probability functions.

Moreover, it defines uniquely a probability function given a value of this probability function. A recursive scheme for t-th order cumulative distribution functions that is based on the De Pril transform has been developed by Dhaene et al. (1999). Some quantities can be calculated directly from such functions.

The first chapter of this thesis introduces Panjer's recursion that is one of the well known methods used to evaluate the probability function of the aggregate claims amount. We expose its history in the actuarial literature as well as in the literature of other fields. We discuss its extensions in addition to other methods for the evaluation of such a probability function or related quantities. Further results on recursive evaluations and on the generalizations of Panjer's recursion can be found in Sundt and Vernic (2009).

Chapter 2: "How to get rid of round-off errors in recursive formulas"

In this first essay, we develop efficient computational methods in order to obtain an accurate recursive evaluation of the probability function of a compound distribution under the collective risk model. Such evaluations may be ineffective due to the propagation of round-off errors coming from the representation of real numbers by computers. The propagation of such errors affects the stability of the recursive evaluation and may lead to meaningless values whose relative error may increase without bound. We discuss the utility of the GNU Multiple Precision Arithmetic Library (GMP), which provides efficient computational functions using arbitrary-precision arithmetic. Comparisons between the use of this library and Maple are made to show how GMP is helpful to save time in computations. We also investigate in detail recursive evaluations for compound binomial distributions, which are particularly subject to an undesirable propagation of round-off errors. The efficient computational methods developed in this essay are built on properties of GMP like the management of the precision of variables. Some numerical examples illustrate these methods in order to show their benefits.

Chapter 3: "From approximations of De Pril transforms to approximations of t-th order cumulative distribution functions"

In this essay, we consider recursive evaluations of the t-th order cumulative distribution function based on the De Pril transform of the corresponding probability function. We

expose a more efficient evaluation scheme than the one developed in Dhaene et al. (1999) especially when the De Pril transform converges to zero. We discuss the utility of such an evaluation for the computation of the expected shortfall at a given probability level as well as for computations in relation to stop-loss contracts. An expression for the error resulting from an evaluation of the t-th order cumulative distribution function from an approximation of the De Pril transform is derived. We also develop a bound for this error that can be computed before the evaluation and discuss its implementation to the approximation of quantities related to stop-loss contracts. Finally, we express this error bound for well-known or useful approximations that have been discussed in the actuarial literature. We end this essay by doing some numerical examples of the previous applications.

Chapter 4: "On the stability of recursive evaluations of t-th order cumulative distribution functions"

In this last essay, we study the stability against round-off errors of recursive evaluations of t-th order cumulative distribution functions especially for the individual risk model. First, we present some recursive formulas that depend on the De Pril transform of the probability function. The Dhaene-Vandebroek algorithm is extended to recursive evaluations of t-th order cumulative distribution functions. This algorithm gives in many cases a more efficient way for evaluating such functions. Then, we show that the stability against round-off errors of such recursive evaluations depends essentially on the convergence or divergence rate of the De Pril transform. In particular, we find that the recursive evaluation of the t-th order cumulative distribution function with $t \geq 1$ is strongly stable when the De Pril transform converges to zero. In the other cases, we give methods to determine a precision of the floating-point representation that is necessary to guarantee an accurate evaluation up to any given point. Finally, numerical applications are made at the end of this essay to illustrate these results.

Chapter 1

Introduction

In the actuarial literature, a recursive procedure for the evaluation of the probability function of a family of compound distributions is known under the name of Panjer's recursion as a reference to Panjer (1981). This recursion can be used for the evaluation of the probability function of the aggregate claims amount under the collective risk model that is defined as

$$S = X_1 + X_2 + \cdots + X_N, \tag{1.1}$$

where the X_i's are assumed to be independent and identically distributed and independent of N. It holds for a class of distributions of N that satisfy

$$p_n = \left(a + \frac{b}{n}\right) p_{n-1}, \quad n = 1, 2, \ldots, \tag{1.2}$$

with initial value $p_0 > 0$ and where p_n denotes $\Pr[N = n]$. This class of distributions is sometimes called Panjer$(a, b, 0)$ class. Panjer's recursion is given by

$$f(x) = \sum_{y=1}^{x} \left(a + b\,\frac{y}{x}\right) g(y) f(x - y), \quad x = 1, 2, \ldots, \tag{1.3}$$

with initial value $f(0) = p_0$ and where g denotes the probability function of the X_i's. We assume in this chapter that g is defined on the positive integers, but similar expressions can be derived in the cases where $g(0) > 0$. The probability function of the non-trivial distributions that satisfy (1.2) are given in Table 1.1 in addition to their parameters a and b and their probability generating function.

We can also use probability generating functions in order to determine the probability function of S. We define the probability generating function of a random variable Y

Table 1.1: Panjer's class distributions

Distribution of N	$p_n = \Pr[N = n]$	a	b	$\mathrm{E}\left[z^N\right]$
Poisson	$e^{-\lambda}\frac{\lambda^n}{n!}$	0	λ	$e^{\lambda(z-1)}$
Negative binomial	$\binom{r+n-1}{n}(1-q)^r q^n$	q	$(r-1)q$	$\left(\frac{1-q}{1-qz}\right)^r$
Binomial	$\binom{m}{n}q^n(1-q)^{m-n}$	$\frac{q}{q-1}$	$(m+1)\frac{q}{1-q}$	$(1-q+qz)^m$

distributed on the nonnegative integers with probability function h as

$$P_Y(u) \;=\; \mathrm{E}\left[u^Y\right] = \sum_{k=0}^{\infty} h(k)u^k. \tag{1.4}$$

We have that

$$P_S(u) \;=\; P_N\left(P_X(u)\right), \tag{1.5}$$

which leads to

$$\sum_{k=0}^{\infty} f(k)u^k \;=\; P_N\left(\sum_{k=0}^{\infty} g(k)u^k\right). \tag{1.6}$$

Therefore, determining f amounts to determining the coefficient of a polynomial that is a function of another polynomial. Recursive formulas for such an operation were derived during the 18th century in pure mathematics literature. From Table 1.1, we observe that the corresponding recursions to (1.3) for compound negative binomial distributions and compound binomial distributions were first developed in sections 68 (p. 53) and 76 (p. 59) of Euler (1748), respectively. For compound Poisson distributions, Euler (1751, p. 10) derived a recursive formula to determine the coefficients $f(k)$ of (1.4) (see also Euler (1755, Ch. 8)).

The family of distributions of Table 1.1 was not discussed for the first time by Panjer (1981). Katz' class whose name refers to Katz (1945) (see also Katz (1965)) is similar to Panjer$(a, b, 0)$ class with a different parametrization. Johnson et al. (2005) also refer to Carver (1919, p. 53) who used the difference equation

$$\frac{p_{n+1} - p_n}{p_n} \;=\; \frac{a-n}{b_0 + b_1 n + b_2 n^2}, \tag{1.7}$$

for smoothing actuarial data. Katz' class contains the distributions for which $b_0 = b_1$ and $b_2 = 0$ in (1.7).

The recursion (1.2) for compound Poisson distributions was largely discussed before Panjer (1981) especially in the biometric literature (see e.g. Adelson (1966), Beall and Rescia (1953, p. 356) and Neyman (1939, p. 47)). The three recursions for the three distributions of Table 1.1 are treated in Khatri and Patel (1961). The Poisson case corresponds to their case A while the binomial negative and binomial cases belong to their case B.

In the actuarial literature, Stroh (1978) deduced the recursive formulas for the compound Poisson distribution and the compound negative binomial distribution using discrete Laplace transforms. Then, Panjer (1980) and Williams (1980) derived the recursive formula for the compound Poisson distribution. Tilley (1980) derived the recursive formulas for each distribution of Table 1.1.

Panjer (1981) derived also an integral equation in the cases where the X_i's are continuously distributed on the positive real numbers with probability density function $g(x)$, $x > 0$, and when the distribution of N satisfies (1.2). Together with $f(0) = p_0$, this integral equation is

$$f(x) \;=\; p_1 g(x) + \int_{y=0}^{x} \left(a + b\,\frac{y}{x}\right) g(y) f(x-y)\,dy, \quad x > 0, \tag{1.8}$$

which can be obtained by determining the coefficient of e^{zx} on both sides of

$$M_S'(z) \;=\; aM_X(z)M_S'(z) + (a+b)M_X'(z)M_S(z), \tag{1.9}$$

where $M_Y(u) = \mathrm{E}\left[e^{uY}\right]$ denotes the moment generating function of Y. An integral equation for the Poisson case and the negative binomial case were discussed in Plackett (1969, p. 3) and Seal (1971, p. 90), respectively. We refer to Ströter (1985) for a numerical solution of such an integral equation. Another approach consists in the discretization of the probability density function of the X_i's and then to use (1.3) for the recursive evaluation.

Sundt and Jewell (1981) generalized the results of Panjer (1981). They derived a recursive formula when (1.2) holds from an integer k on such that

$$p_n \;=\; \left(a + \frac{b}{n}\right) p_{n-1}, \quad n = k+1, k+2, \ldots, \tag{1.10}$$

with $p_k > 0$. They obtained that

$$\begin{aligned} f(x) \;=\; & p_1 g(x) + \sum_{n=2}^{k} \left(p_n - \left(a + \frac{b}{n}\right) p_{n-1}\right) g^{*n}(x) \\ & + \sum_{y=1}^{x-1} \left(a + b\,\frac{y}{x}\right) g(y) f(x-y), \quad x = 1, 2, \ldots, \end{aligned} \tag{1.11}$$

with $f(0) = p_0$ and where g^{*n} denotes the n-fold convolution of g with itself. They also extend Panjer's recursion for the cases where the X_i's are distributed on the set of all integers. Panjer(a, b, k) class was characterized by Hess et al. (2002). It contains the distributions of N that satisfies (1.10) with $p_n = 0$ for $n = 0, 1, \ldots, k-1$ and $p_k > 0$. For such distributions, (1.11) becomes

$$f(x) = p_k g^{*k}(x) + \sum_{y=1}^{x-k} \left(a + b\,\frac{y}{x}\right) g(y) f(x-y), \quad x = k, k+1, \ldots. \tag{1.12}$$

Gerhold et al. (2010) discussed a stable recursion for some distributions of Panjer$(a, b, 1)$ class.

Ambagaspitiya (1995) derived a recursion for f when the probability function of N satisfies

$$p_n(a,b) = \left(h_1(a,b) + \frac{h_2(a,b)}{n}\right) p_{n-1}(a+b, b), \quad n = k, k+1, \ldots, \tag{1.13}$$

where a and b are parameters of N (see also Gathy and Lefèvre (2010)). It is given by

$$\begin{aligned} f(x;a,b) &= \sum_{n=1}^{k-1} \left(p_n(a,b) - \left(h_1(a,b) + \frac{h_2(a,b)}{n}\right) p_{n-1}(a+b,b)\right) g^{*n}(x) \\ &\quad + \sum_{y=1}^{x} \left(h_1(a,b) + h_2(a,b)\,\frac{y}{x}\right) g(y) f(x-y; a+b, b), \quad x = 1, 2, \ldots, \end{aligned} \tag{1.14}$$

with initial values $f(0; a, b) = p_0(a, b)$.

De Pril (1985) showed that h^{*n}, the n-fold convolution of a discrete probability function $h(y)$, $y = k, k+1, \ldots$, where k is an integer, can be evaluated recursively by

$$h^{*n}(x) = \frac{1}{h(k)} \sum_{y=1}^{x-nk} \left(\frac{n+1}{x-nk}\,y - 1\right) h(y+k) h^{*n}(x-y), \quad x = nk+1, nk+2, \ldots, \tag{1.15}$$

with initial value $h^{*n}(nk) = (h(k))^n$. Notice that h^{*n} is the probability function of a shifted compound binomial distribution with parameters $m = n$ and $q = 1 - h(k)$. If $k = 0$, (1.15) is the same as (1.3) when N is distributed according to a binomial distribution. Therefore, this recursion can also be used to evaluate the probability function of the aggregate claims amount in the individual risk model defined as

$$S = X_1 + X_2 + \cdots + X_m, \tag{1.16}$$

where the X_i's are assumed to be independent and identically distributed. If the X_i's are not identically distributed and have probability function h_i, a recursive evaluation is

obtained using the De Pril transform of each h_i. This function is denoted function by φ_i and can be determined by

$$\varphi_i(y) = \frac{1}{h_i(0)}\left[yh_i(y) - \sum_{x=1}^{y-1}\varphi_i(y-x)h_i(x)\right], \quad y = 1, 2, \ldots. \tag{1.17}$$

De Pril (1989) proved that the De Pril transform of S denoted by φ is given by

$$\varphi(y) = \sum_{i=1}^{m}\varphi_i(y), \quad y = 1, 2, \ldots, \tag{1.18}$$

which leads to the recursion

$$f(x) = \frac{1}{x}\sum_{y=1}^{x}\varphi(y)f(x-y), \quad x = 1, 2\ldots, \tag{1.19}$$

with initial value $f(0) = \prod_{i=1}^{m} h_i(0)$. This last formula was given by Chan (1982a,b). Another recursive scheme for this model was derived by Dhaene and Vandebroek (1995). That is

$$f(x) = \frac{1}{x}\sum_{i=1}^{m}v_i(x), \quad x = 1, 2\ldots, \tag{1.20}$$

where the coefficients $v_i(x)$ are determined by

$$v_i(x) = \frac{1}{h_i(0)}\sum_{y=1}^{x}h_i(y)\left(yf(x-y) - v_i(x-y)\right), \quad x = 1, 2, \ldots, \tag{1.21}$$

and with initial values $f(0) = \prod_{i=1}^{m} h_i(0)$ and $v_i(0) = 0$, $i = 1, \ldots, m$. This algorithm, which was previously derived by Waldmann (1994) for the individual life model, is in several situations more efficient than an evaluation using (1.19). Notice that if $h_i(x) = h_j(x)$ for all x and $i \neq j$, (1.19) and (1.20) simplify to (1.15) with $k = 0$. In the actuarial literature, we can also find some approximations of the probability function for the individual risk model (see De Pril (1989), Kornya (1983) and Hipp (1986)). Comparisons between these exact and approximative evaluations are made in Dhaene et al. (2006) and Dhaene and Vandebroek (1995).

Some generalizations for the condition on the probability function of N were discussed in the actuarial literature. Hesselager (1994) and Panjer and Willmot (1982) developed recursive formulas when p_n satisfies

$$p_n = \frac{\sum_{j=0}^{k} a_j n^j}{\sum_{j=0}^{k} b_j n^j}\, p_{n-1}, \quad n = 1, 2, \ldots. \tag{1.22}$$

Another extension was made by Schröter (1990) and generalized by Sundt (1992). The latter showed that

$$f(x) = \sum_{y=1}^{x} f(x-y) \sum_{j=1}^{k} \left(a_j + \frac{b_j}{j} \frac{y}{x} \right) g^{*j}(y), \quad x = 1, 2, \ldots, \tag{1.23}$$

with initial value $f(0) = p_0$ when

$$p_n = \sum_{j=1}^{k} \left(a_j + \frac{b_j}{n} \right) p_{n-j}, \quad n = 1, 2, \ldots, \tag{1.24}$$

with $p_0 > 0$. When N is a phase-type variable, its probability generating function is rational and can be written as

$$P_N(z) = \frac{\sum_{j=0}^{k} \alpha_j z^j}{1 - \sum_{j=1}^{k} \beta_j z^j}. \tag{1.25}$$

This is equivalent to writing that

$$p_n = \alpha_n + \sum_{j=1}^{k} \beta_j p_{n-j}, \quad n = 0, 1, 2, \ldots. \tag{1.26}$$

and the probability function of S can be evaluated by

$$f(x) = \sum_{j=1}^{k} \alpha_j g^{*j}(x) + \sum_{y=1}^{x} f(x-y) \sum_{j=1}^{k} \beta_j g^{*j}(y), \quad x = 1, 2, \ldots. \tag{1.27}$$

with initial value $f(0) = p_0 = \alpha_0$. This recursion is given by Eisele (2006) who also extends it to the cases where the X_i's are continuous random variables.

If N follows a Poisson distribution and the X_i's are discrete phase-type variables that satisfy

$$\frac{d}{dz} P_X(z) = \frac{\sum_{j=1}^{k} \gamma_j z^{j-1}}{1 - \sum_{j=1}^{k} \delta_j z^j}, \tag{1.28}$$

De Pril (1986a) gave a more efficient recursive formula than (1.3) which is

$$f(x) = \sum_{j=1}^{k} \left(\frac{\lambda}{x} \gamma_j + \left(1 - \frac{j}{x} \delta_j \right) \right) f(x-j), \quad x = 1, 2, \ldots, \tag{1.29}$$

with initial value $f(0) = e^{-\lambda}$. An extension of this recursion to the distributions of Panjer's class is made by Hipp (2006) who also extends it when g is the probability density function of a continuous phase-type distribution.

If we consider multivariate distributions, we can obtain similar recursive evaluations. We refer e.g. to Hesselager (1996) and Vernic (1999) for bivariate compound distributions and to Eisele (2008), Sundt (1999a, 2000) and Sundt (2002, Section 9) for multivariate distributions.

Nowadays, compound distributions and convolutions appear in most of the fields of actuarial practice. We give here two practical examples. The individual risk model is generally used to model the distribution of the loss of a life insurance or a pension fund that depends on the sum of the sum at risk of the policies for which a claim occurs over a given period of time. In the standard model of the Swiss Solvency Test, we need to compute the expected shortfall of a distribution that involves the evaluation of the probability function of a compound Poisson distribution. However, recursive formulas are not the only way that can be used for the evaluation of the probability function of a compound distribution. We discuss now other methods for such an evaluation.

A first method consists in using the convolution formula which is

$$f(x) = \sum_{n=0}^{\infty} p_n g^{*n}(x), \quad x = 0, 1, 2, \ldots, \tag{1.30}$$

where

$$g^{*n}(x) = \sum_{y=1}^{x} g(y) g^{*(n-1)}(x-y), \quad x = 1, 2, \ldots, \tag{1.31}$$

with initial value

$$g^{*0}(x) = \begin{cases} 1, & x = 0 \\ 0, & x = 1, 2, \ldots \end{cases}. \tag{1.32}$$

However, an evaluation using (1.30) is generally more demanding in terms of number of operations than Panjer's recursion. They require $O(n^3)$ and $O(n^2)$ operations to obtain $f(0), f(1), \ldots, f(n)$, respectively.

A second method that requires only $O(n \log n)$ operations is to use the fast Fourier transform (FFT). It consists in evaluating the probability function of S using the inverse discrete Fourier transform by

$$f(x) = \frac{1}{n} \sum_{k=0}^{n-1} \phi_S\left(\frac{2\pi k}{n}\right) e^{\frac{-i2\pi xk}{n}}, \quad x = 0, 1, 2, \ldots, n-1, \tag{1.33}$$

where $i = \sqrt{-1}$, n is the number of points that are evaluated and ϕ_S is the characteristic function of S which is defined by

$$\phi_S(z) = \mathrm{E}\left[e^{izS}\right] = P_N\left(\phi_X(z)\right). \tag{1.34}$$

The terms $\phi_S\left(\frac{2\pi k}{n}\right)$ of (1.33) can be determined using (1.34) and the discrete Fourier transform

$$\phi_X(z) = \sum_{k=0}^{n-1} g(k) e^{\frac{i2\pi zk}{n}}. \tag{1.35}$$

If n is chosen to be a power of two, (1.33) and (1.35) can be computed efficiently using an FFT algorithm. This method, which can be used for any random variable with probability generating function P_N, leads to an approximation whose accuracy depends on the choice of n since it introduces an aliasing error which is equal to

$$f(x) - \Pr[S = x] = \sum_{j=1}^{\infty} \Pr[S = x + jn], \quad x = 0, 1, 2, \ldots, n-1, \tag{1.36}$$

where $\Pr[S = s]$ denotes the exact value of the probability function of S at s. We refer to Bühlmann (1984) for a comparison between evaluations using Panjer's recursion and a method based on the fast Fourier transform algorithm. An exponential tilting procedure was proposed by Grübel and Hermesmeier (1999) in order to reduce the aliasing error (see also Embrechts and Frei (2009)).

There also exist approximations for the distribution function based on the first central moments of S (see e.g. Kaas et al. (2008, Sections 2.5 and 3.7)). Monte Carlo simulation can also be used to approximate any quantity related to S. However, some quantities like stop-loss premiums can be obtained from t-th order cumulative distribution functions that can be evaluated recursively from the De Pril transform by

$$\Gamma^t f(x) = \frac{1}{x} \sum_{y=1}^{x} (\varphi(y) + t)\, \Gamma^t f(x-y), \quad x = 1, 2 \ldots, \tag{1.37}$$

with initial value $\Gamma^t f(0) = f(0)$.

Chapter 2

How to get rid of round-off errors in recursive formulas

2.1 Introduction

The collective and individual risk models are frequently used in risk theory applications especially to represent the aggregate claims amount of a portfolio of insurance policies over a period of time. In the collective model, this amount is modelled by a compound distribution which is used, e.g., to compute stop-loss premiums as well as risk margins to satisfy solvency criteria. In real life applications, the main problem with compound distributions is the evaluation of their probability function since it generally involves a large number of operations which depends on the range of the support of the considered random variables.

In the actuarial literature, a recursive procedure for the evaluation of the probability function of a family of compound distributions is known under the name of Panjer's recursion as a reference to Panjer (1981). This family of distributions is called Panjer's class. However, such recursive formulas for Panjer's class distributions were discussed separately in Khatri and Patel (1961) and the recursion for the compound Poisson distribution was treated e.g. in Adelson (1966), Beall and Rescia (1953) and Neyman (1939). In pure mathematics, these recursive formulas appeared in sections 68 and 76 of Euler (1748) and in Euler (1751, p. 10) for compound negative binomial, compound binomial and compound Poisson distributions, respectively. In the actuarial literature, Stroh (1978) deduced the recursions for the compound Poisson and the compound negative binomial distributions using discrete Laplace transforms. Then, Panjer (1980) and Williams (1980)

derived the recursive formula for compound Poisson distributions. Tilley (1980) derived the recursive formula for each distribution of Panjer's class. The results of Panjer (1981) are generalized in Sundt and Jewell (1981).

Recursive evaluations are very useful since they reduce largely the number of operations in comparison to the evaluation using convolutions. Therefore, the evaluation is faster especially when the expected value of the number of claims is large. The computer programming implementation is also easier because we can set a stop condition when the values of the probability function become negligible. However, when we are using floating-point numbers, meaningless results may arise with recursive formulas because round-off errors occur from the representation of real numbers by computers. At each stage of the recursion, an error occurs and affects further computations with more or less effects on the accuracy of additional points. Throughout this paper, the word *accuracy* is defined to be the number of decimal digits that are exact in the evaluation of a given point.

Panjer and Wang (1993) focused on the stability against round-off errors for the different cases discussed in Panjer (1981). They showed that the recursive evaluation for compound Poisson and compound negative binomial distributions are stable while the recursive evaluation is unstable for the compound binomial case. An unstable evaluation means that the magnitude of the relative error blows up such that we get meaningless results. Even worse we may obtain an overflow.

Kornya (1983) derived an algorithm to compute the aggregate claims amount distribution of a traditional life insurance portfolio. If we assume that policies have independent and identical claim amount distributions, the aggregate claims amount random variable is distributed according to a compound binomial distribution. This means that the probability function of an n-fold convolution can be evaluated using the recursive formula for the compound binomial case discussed in Panjer (1981). This result can be found in De Pril (1985). De Pril (1986b) derived another algorithm to evaluate the aggregate claims amount distribution in the individual risk model. It is improved in De Pril (1989). Dhaene and Vandebroek (1995) and Waldmann (1994) discussed recursive evaluations for the same distribution but that reduce the number of arithmetic operations. Such evaluations are also unstable and applying them to calculate the probability function could be worthless due to the propagation of round-off errors.

As pointed out by Shiu (1983), the recursive formula for compound binomial distributions is a particular case of the J. C. P. Miller's formula. This formula is well-known by numerical analysts and computer scientists and many results can be found in the numerical

analysis literature. Olver (1964) analyzed the error accumulation in Miller's algorithm. The computational behavior of the solutions of a second order recurrence relation is discussed in Gautschi (1967). Oliver (1967) developed a relative stability theory and studied the propagation error for recurrence relation of greater order. Zahar (1977) discussed the stability of a generalization of Miller's algorithm. More general results can be found in Cash (1980).

In order to obtain an accurate evaluation efficiently, we use an arbitrary-precision arithmetic library called GNU Multiple Precision Library[1] (GMP), which is an efficient computational tool especially for basic arithmetic operations. With GMP, we can compute with an arbitrary precision that is only limited by the available memory of the computer. GMP also avoids the use of scaling functions because the range of numbers that can be represented by GMP floating-point variables is more than sufficient to represent the values of compound distributions in comparison to standard programming languages. The scaling functions which are discussed in Panjer and Willmot (1986) and Waldmann (1994) are used to avoid underflows or overflows in recursive evaluations. Throughout this paper, the word *precision* is defined to be the number of bits that is used to represent a real number by a floating-point number.

In Section 2.2, we define the collective risk model. In Section 2.3, we make a review on recurrence equations and expose their solutions with the numerical problems that they may involve. We introduce the arbitrary-precision arithmetic and the GMP library in Section 2.4 where we describe its most useful functions and how the precision of variables can be controlled. In Section 2.5, we study how we can evaluate more efficiently the probability function of a compound distribution using Panjer's recursive formula for the stable cases. In Section 2.6, we investigate in details the recursive evaluation of the probability function of a compound binomial distribution and find an efficient computational method which leads to an accurate approximation of this function using GMP. We conclude this paper by doing some remarks on the extension of our results to more general recursive evaluations of the aggregate claims amount distribution. We also discuss the effects of the use of the floating-point representation with other methods of evaluation of this distribution.

[1]The manual by Granlund (2007) explains in details how to use GMP, the functions that are defined and the algorithms that it uses to be efficient in computations.

2.2 The collective risk model

The collective risk model is used to represent the aggregate claims amount of a portfolio of insurance policies over a period of time. The aggregate claims amount random variable S is defined as

$$S = X_1 + X_2 + \cdots + X_N, \tag{2.2.1}$$

with the standard convention that the value of an empty sum is zero ($N = 0$). The random variables N and X_k, representing the number of claims and the amount of the k-th claim, respectively, are assumed to be mutually independent. We also assume that $X_1, X_2, \ldots$ are identically distributed on the positive integers. Notice that if the distribution of these random variables is continuous, a discretization of its probability density function will be necessary.

The probability function of S can be determined by

$$f_S(x) = \sum_{n=0}^{\infty} p_n f_X^{*n}(x), \quad x = 0, 1 \ldots, \tag{2.2.2}$$

where $p_n = \Pr[N = n]$ and f_X^{*n} is the n-fold convolution of f_X with itself and with f_X^{*0} being the probability function of a degenerate distribution at zero.

For the distributions of N satisfying

$$p_n = \left(a + \frac{b}{n}\right) p_{n-1}, \ n = 1, 2, \ldots, \text{ and } p_0 > 0, \tag{2.2.3}$$

Panjer (1981) showed that f_S can be evaluated recursively by

$$f_S(x) = \sum_{y=1}^{x} \left(a + b\,\frac{y}{x}\right) f_X(y) f_S(x - y), \quad x = 1, 2, \ldots, \tag{2.2.4}$$

with initial value $f_S(0) = p_0$. Sundt and Jewell (1981) showed that the Poisson, the binomial and the negative binomial distributions are the only three distributions that satisfy (2.2.3), if we exclude the degenerate distribution at zero which is not the most interesting distribution to use in this model. Panjer (1981) determined the coefficients a and b corresponding to each of these three distributions. He expressed (2.2.4) for each one including the relation where N follows a geometric distribution, which is a special case of the negative binomial distribution. Observe that if there exists an integer s such that $f_X(x) = 0$ for all $x > s$, the recursive formula (2.2.4) becomes

$$f_S(x) = \sum_{y=1}^{x \wedge s} \left(a + b\,\frac{y}{x}\right) f_X(y) f_S(x - y), \quad x = 1, 2, \ldots, \tag{2.2.5}$$

where $x \wedge s = \min(x, s)$, which avoids some useless computations.

2.3 Recurrence equations and stability

We present now some general results on recurrence equations and explain the causes of the instability in the evaluation of some of their solutions. We consider the s-th order recurrence equation

$$f(x) = \sum_{y=1}^{s} a_y(x) f(x-y), \quad x = 1, 2, \ldots, \tag{2.3.1}$$

where the coefficients $a_y(x)$ are known and $a_s(x) \neq 0$ for all x. The general solution, $g(x)$, of (2.3.1) can be expressed as

$$g(x) = \sum_{j=1}^{s} c_j u_j(x), \quad x = 1, 2, \ldots, \tag{2.3.2}$$

where $u_1, u_2, \ldots, u_s$ are linearly independent functions and compose the fundamental set of solutions of (2.3.1). Then, we need a set of s initial values to be able to determine the coefficients c_j, $j = 1, 2, \ldots, s$. If we consider (2.2.5), the coefficients that gives the probability function of S are determined from the implicit initial values $f_S(0) = p_0$ and $f_S(x) = 0$, $x = -1, -2, \ldots, -s+1$.

A solution h of (2.3.1) is said to be *dominated* by another solution g of (2.3.1) if

$$\lim_{x \to +\infty} \frac{h(x)}{g(x)} = 0. \tag{2.3.3}$$

In the fundamental set of solutions there is one solution which dominates the $s-1$ other solutions of the set, let it be u_1. This solution is said to be *dominant* while the solutions $u_2, \ldots, u_s$ are said to be *subordinate* or non-dominant. In other words, the solution (2.3.2) is dominant if $c_1 \neq 0$ and subordinate if $c_1 = 0$.

When the floating-point representation is used in computations, Cash (1980) gave the result that the recursive evaluation of a dominant solution is stable against round-off errors while it is unstable for a subordinate solution. Referring to Oliver (1967), the evaluation of a solution is *stable* or effective if the relative error grows linearly with respect to the number of stages while it is *unstable* or ineffective if the relative error grows more than linearly with respect to the number of stages. If a recursive evaluation is unstable, an insufficient accuracy may arise in the results which could even be meaningless. Inspired by Gautschi (1967, p. 25) we give now an illustration in order to explain why such results may arise. Let h be a subordinate solution of (2.3.1) with initial values

$$h(x) = 0, \ x = -1, -2, \ldots, -s+1 \text{ and } h(0) = \alpha_0. \tag{2.3.4}$$

Due to the floating-point representation of the initial value α_0 by $\tilde{\alpha}_0$, a round-off error occurs which is equal to $\epsilon_0 = \alpha_0 - \tilde{\alpha}_0$. Then, the error at the first point, $\epsilon_1 = h(1) - \tilde{\alpha}_1$, where $h(1)$ and $\tilde{\alpha}_1$ are the exact value and its evaluation, respectively, has two sources: the evolution of ϵ_0 and the round-off error coming from the representation of $h(1)$ with a limited precision. Assuming that we compute the following stages of the recursion with an "infinite" precision, the solution, $\tilde{h}$, of (2.3.1) with initial values

$$\tilde{h}(x) = 0,\ x = -1, -2, \ldots, -s+2,\ \tilde{h}(0) = \tilde{\alpha}_0 \text{ and } \tilde{h}(1) = \tilde{\alpha}_1, \tag{2.3.5}$$

is generally dominant ($\tilde{c}_1 \neq 0$). Therefore, the relative error of the perturbed solution $\tilde{h}$ defined by $\left|\frac{\tilde{h}(x)-h(x)}{h(x)}\right|$ increases without bound since we have

$$\lim_{x\to+\infty} \left|\frac{\tilde{h}(x) - h(x)}{h(x)}\right| = +\infty \tag{2.3.6}$$

because $\tilde{h}(x) - h(x)$ is dominant. The magnitude of this error depends on the magnitude of the relative errors of $\tilde{h}(0)$ and $\tilde{h}(1)$ that are equal to $\frac{\epsilon_0}{\alpha_0}$ and $\frac{\epsilon_1}{h(1)}$, respectively. Gautschi (1967) discussed the effectiveness of second order recurrence equations and gave several examples. This case is also developed in Olver (1964), which gives an analysis of the error of Miller's algorithm. The general case of linear recurrence relations of greater order is discussed in Oliver (1967) and Zahar (1977).

Let us now consider the s-th order congruent recurrence equation

$$f(x) \quad = \quad \sum_{y=1}^{s} b_y(x)c(y)f(x-y), \quad x = 1, 2, \ldots, \tag{2.3.7}$$

with $b_y(x) > 0$ for $y = 1, 2, \ldots, s$ and $x > 0$, and where $c(y)$ is a function defined on positive integers with support $\{y_1, y_2, \ldots, s\}$ where $1 \leq y_1 < y_2 < \cdots < s < \infty$ and with one being their greatest common divisor. If the initial values are nonnegative and at most one is positive, then the solution of (2.3.7) is dominant. This result is shown in Panjer and Wang (1993). All these conditions are satisfied for (2.2.5) when N follows a Poisson distribution or a negative binomial distribution. Therefore, the recursive evaluation using (2.2.5) is stable in these two cases. They also showed that the relative error grows linearly with a slope smaller than one with respect to the number of stages. However, when N is distributed according to a binomial distribution, the solution of (2.2.5) is subordinate since the support of the compound binomial distribution is finite and $f_S(x) = 0$ from a given point. Moreover, the coefficients $b_y(x)$ become negative from some point, which causes the instability of the recursive evaluation. The recursive evaluation of the probability function of compound binomial distributions is treated in details in Section 2.6.

Globally, there are two approaches to get rid of the round-off errors propagation. The first one consists in avoiding the use of the floating-point representation such that the variables are represented by expressions of rational numbers and are evaluated only to get their numerical value. This approach leads to an exact evaluation but is inefficient due to its computation time that is much longer compared to the use of the floating-point representation. The second approach is to use the floating-point representation but to increase its precision. Although the computation time increases with respect to the precision, the use of an efficient computational tool providing methods that allows a precision management can be very useful in order to obtain an efficient evaluation.

2.4 The GNU Multiple Precision Arithmetic Library

With standard programming languages like C, C++ or Java, the representation of a real number, which is called a floating-point number, is limited to some types. For example, `float`, `double` and `long double` are the three types that we can use in the C++ language to represent real numbers. These types allow us to work with different precisions of the floating-point representation. The accuracy of a floating-point number depends on the precision assigned to each type. When a recursive evaluation is unstable, we generally need to use a greater precision than the one that we can reach with these types. It would be also useful to be able to choose the precision that we want to assign to each floating-point number. These two properties are included in the GNU Multiple Precision Arithmetic Library (GMP), an arbitrary-precision arithmetic library that we can use in addition to the C++ language, a compiled language. The arbitrary-precision arithmetic, also called bignum arithmetic, allows us to compute with an arbitrary precision only limited by the available memory of the computer. It is very useful when we have to work with numbers that contain many digits. These numbers can be integers as well as real numbers.

The GNU Multiple Precision Arithmetic Library is a free portable library written in C allowing computations with arbitrary-precision on integers, rational numbers and floating-point numbers. Its goal is to provide an efficient basic arithmetic on these three types. In GMP, each type has a corresponding C++ class such that an object declared by a GMP class is associated to a GMP C type variable. The most interesting class for us is the class for floating-point numbers that are at the heart of our problem. It is called `mpf_class` for multiple precision floating-point and corresponds to the GMP C type `mpf_t`. The GMP floating-point type can represent numbers over the range from $2^{-68719476768}$ to $2^{68719476736}$,

but the notions of underflow and overflow are not defined in this library. Therefore, when a number cannot be represented the execution of the program stops and an error message appears. The GMP class interface offers overloaded functions like absolute value, floor or square root as well as overloaded operators which are more convenient to write code. There are several other functions which do not have a C++ class interface. However, we are able to use them since there are functions that convert a GMP class object to a corresponding GMP C type variable. If we need to use several times one of these functions, we can create an equivalent C++ function, which may assume objects as arguments and returns an object. Exponential, logarithmic and trigonometric functions are not defined in GMP, but if necessary we can implement them.

At the beginning of a program written using GMP, we can set a default precision. This means that when we declare a floating-point object, it will have at least this precision. For computational efficiency, the precision of an object can take only a multiple of 32 bits with a minimum value of 64 bits. For example, if we set the default precision to 100 bits, the precision that objects will have is actually of 128 bits. Then, we can set a greater or smaller precision to each object that we declare and we can change it easily as we go along with the program. Therefore, two objects in the same program do not have necessarily the same precision.

When we are working with different precisions in a program, the precision that is used in a calculation is the one of the destination object. Therefore, when a computation needs an intermediate object, the precision of this object is the one of the destination object. For example, if we compute `c=a*x+b*y` an intermediate variable is needed to represent one of the two products. Let it be `d=a*x`. GMP assigns the precision of `c` to the variable `d`. Now, imagine that we do the same computation but using two operations: `e=a*x` and `c=e+b*y`. We can choose to set a different precision than that of `c` to `e`. Thus, the value of `c` may differ between the computation in one operation and the computation in two operations. To save time, it is sometimes useful to split a computation into several parts and compute each part with a suitable precision since the computation time depends to a large extent on the precision of the objects that are involved.

Since we are computing with numbers and we set ourselves their precision in bits, it could be interesting to find a relation between the precision in bits that is used in the representation of a real number by a GMP floating-point object and the number of decimal digits that are exact in this representation, and conversely. This relation is a change of base from base two for the precision to base ten for the number of decimal digits. Therefore, we can imagine that this number may be obtained by multiplying the precision by $\log_{10} 2$.

We check if this relation is true by doing a small example. Let us do the following computations for different precisions using GMP. First, we set the value $\frac{10}{3} = 3.\bar{3}$ to a GMP floating-point object called `t`, which generates a round-off error. The number of times that the digit 3 appears is the number of decimal digits that the computer has used to represent this fraction. Then, we consider the relative error

$$\eta = \left| \frac{10 - 3\,t}{10} \right|, \tag{2.4.1}$$

which would be equal to 0 if there was no round-off error at the previous step. The number of decimal digits used by the computer with the chosen precision is given by

$$d = \lfloor -\log_{10} \eta \rfloor\,, \tag{2.4.2}$$

where $\lfloor \cdot \rfloor$ denotes the floor function. Remark that d is the accuracy of the representation of $\frac{10}{3}$ by a GMP floating-point object. Table 2.1 contains the values of η and d for some

Table 2.1: Accuracy versus precision with GMP floating-point objects

r	η	d
64	$5.42101 \cdot 10^{-21}$	20
128	$2.93874 \cdot 10^{-40}$	39
256	$8.63617 \cdot 10^{-79}$	78
512	$7.45834 \cdot 10^{-156}$	155
1024	$5.56268 \cdot 10^{-310}$	309
8192	$9.16802 \cdot 10^{-2468}$	2467
32 768	$7.06484 \cdot 10^{-9866}$	9865

precisions. We observe that for this example the relative error is given by

$$\eta = \frac{2^{-r}}{10}, \tag{2.4.3}$$

for every precision r. From (2.4.2) and (2.4.3), it follows that

$$d \;=\; \lfloor r \cdot \log_{10} 2 \rfloor + 1. \tag{2.4.4}$$

Note that the floor function is used in order to obtain an integer for the number of exact decimal digits. With GMP, in the case where a real number can be exactly represented using only a few bits, only these bits are used in computations even if we wanted to

use a larger precision. This arises, for example, when a power of $\frac{1}{2}$ is represented by a floating-point number. Therefore, (2.4.4) gives the minimum accuracy for a given precision. Conversely, the minimum precision that has to be set in order to guarantee a desired accuracy of the floating-point representation is given by

$$r = \lceil (d-1)\log_2 10 \rceil , \tag{2.4.5}$$

where $\lceil \cdot \rceil$ denotes the ceiling function. The precision that is actually assigned to GMP objects is

$$r_{GMP} = \max\left(32\left\lceil \frac{(d-1)\log_2 10}{32} \right\rceil, 64\right). \tag{2.4.6}$$

From (2.4.5), we can express the difference between two precisions from the difference of their respective accuracies by

$$r_2 = r_1 + \lceil (d_2 - d_1)\log_2 10 \rceil . \tag{2.4.7}$$

This relation can be used to determine the precision that we have to use when we want to change the accuracy of the floating-point representation from d_1 to d_2 decimal digits.

The main problem with GMP is that we may obtain different results on two different computers because the rounding of floating-point numbers depends on the computer word size. Nevertheless, its computation speed and the fact that we can easily control the precision of each object give us an efficient computational tool compared to software like Maple, Mathematica or Matlab. Therefore, we use GMP, and especially its classes in addition to the C++ language, to obtain efficient computational methods for the recursive evaluation of the probability function of a compound distribution.

2.5 Efficiency with Panjer's recursion

To be efficient, a program has to provide results with a sufficient accuracy in the least time possible. Due to the calculation speed of GMP, we can use it to evaluate recursively the probability function of a compound distribution with (2.2.5) even for the cases where it is stable. The idea is to compute using an adequate precision. Panjer and Wang (1993, p. 248) gave a formula for the minimum accuracy that we can obtain from the number of stages and the number of decimal digits used to represent real numbers. The lower bound, $v(x)$, of the accuracy obtained at the x-th stage of the recursive evaluation is given by

$$v(x) \geq d + \log_{10} 2 - \log_{10}(x+1), \quad x = 1, 2, \ldots . \tag{2.5.1}$$

Therefore, in order to determine the precision in bits, $r(x)$, that is necessary to guarantee a required accuracy v at the x-th stage, we combine (2.4.5) and (2.5.1) to get

$$r(x) \quad = \quad \left\lfloor \log_2 \left((x+1) \cdot 10^{(v-1)} \right) \right\rfloor, \quad x = 1, 2, \ldots. \tag{2.5.2}$$

We observe that this needed precision will rarely be a large number since each additional digit in the required accuracy or each multiplication of the number of stages by 10 increases it by $\log_2 10 \approx 3.32$. Thus, it is generally smaller than 64 bits, which is the minimum precision that GMP assigns to objects. Nevertheless, the use of this library is recommended because its speed is sufficiently beneficial compared to other software solutions. Time comparisons between the use of GMP and Maple are made at the end of this section.

When the expected value of N, $\mathrm{E}[N]$, is large, two other constraints come up: the ability to represent the values with floating-point numbers and the available memory of the computer.

If $\mathrm{E}[N]$ is large, the values of the probability function of the compound distribution are small and an underflow may happen especially for p_0. If the initial value is too small to be represented by a floating-point number its representation is zero, which is not appropriate for an initial value of such recursive evaluations. The scaling functions (see e.g. Panjer and Willmot (1986) and Waldmann (1994)) give a way to avoid underflows or overflows in recursive evaluations. However, the range of numbers that can be represented with GMP is generally sufficient to represent the values of the probability function of a compound distribution. Thus, GMP avoids the use of scaling functions.

Computers have a limited memory, so we cannot declare as many variables as we wish. A problem arises when we store the values of the probability function in an array, such that they can be used for further computations. If we store them in an array, each element of this array takes a part of the computer memory with the consequence that we are limited in the size of the array. Furthermore, with Maple, the time needed by the computer for each operation grows as we go along with the recursive evaluation since it has to swap pages of memory to the disk from some stage on. This process consumes a lot of time because Maple has to move this memory to the disk and has to take it back to be able to use the variables inside. To avoid this problem and to be able to compare the computation time between Maple and GMP, we opt for another method that allows to declare a smaller number of variables.

Looking at (2.2.5) we observe that we need at least the s previous values to calculate each new value. Therefore, we can create an array with only s elements in which we

store only the values that are needed to evaluate the next stage. First, we evaluate $f_S(0), f_S(1), \ldots, f_S(s-1)$ that we store to their corresponding element of the array. Then, we evaluate $f_S(s)$ which depends on the first s values which are stored in the array and assigns it to the element 0 of the array. We use the modulo operation $s \bmod s$ in order to determine the element of the array where we store this value. Then, for the evaluation of $f_S(s+1)$ we use the same modulo operation to take the value of $f_S(s)$ in the right element of the array. We go on with the recursive evaluation using modulo operations to store or to take a value in the array. The value of $f_S(x)$, $x = 0, 1, \ldots$, is actually stored in the element $x \bmod s$ of the array as long as we need it for the evaluation of further stages. This method allows us to use a smaller number of variables but the values of the probability function are no longer stored in the computer memory at the end of the evaluation and cannot be used to compute the desired quantities.

In order to be able to calculate the required quantities using the values of the probability function there are essentially two ways: the first one is to calculate the required quantities as we go along with the recursion. For example, if we want to calculate a stop-loss premium with deductible d, we will create a variable which will be accumulated by $(x-d)_+ f_S(x)$ at each stage x of the recursive evaluation. The second way is to store elsewhere the values of the probability function. We can write these values in a file such that they are read back when we want to use them or store them in an array declared with a smaller precision. The use of an array is efficient only with GMP as long as computer memory is available while writing in a file can be done with all software and programming languages. In the file we can write the numerical values in decimal or their binary representations, both with the desired accuracy. The latter is more efficient since it avoids the conversions from bits to decimal digits and from decimal digits to bits when we write in the file and when we read it back, respectively. The chosen method will depend on which quantity we want to calculate with the probability function. The advantage to store elsewhere the values or their representations is that we can keep only the digits that fulfill the required accuracy. Remark that it is useless to keep the values with a too large accuracy knowing that the values of the last evaluation points will have a smaller accuracy than the first ones. The use of a sufficient precision is only useful to guarantee accurate values at the last evaluation points.

This procedure is summarized in Algorithm 1 for the computation of the probability function of a compound Poisson distribution with parameters $(\lambda > 0, f_X)$. Notice that at step 6 we choose to stop the recursion at a given quantile that fulfills some condition on the magnitude of the probability function. We can also choose to evaluate up to a

Algorithm 1: Recursive evaluation for compound Poisson distributions

1. Declare a table f with elements from 0 to $s-1$
2. Define a table g with elements from 1 to s for f_X
3. Define λ
4. Set $f[0]^a = e^{-\lambda}$ and $F = f[0]$
5. Store $f[0]$ or F
6. For $i = 1$ while $F < 1 - 10^{-v}$
7. $\quad sum = 0$
8. $\quad$ For $k = 1$ to $\min(i, s)$
9. $\quad\quad sum = sum + k \cdot g[k] \cdot f[i - k \bmod s]$
10. $\quad f[i \bmod s] = \frac{sum \cdot \lambda}{i}, \quad F = F + f[i \bmod s]$
11. $\quad$ Store $f[i \bmod s]$ or F

[a] $h[j]$ represents the element number j of the array h

given point. Finally, steps 5 and 11 consist in storing the probability function or the distribution function in an array or in a file.

Some comparisons[2] between the computation times using Maple XI and GMP[3] are made for the evaluation of several compound Poisson distributions according to Algorithm 1. We compare only the computation times, this is why steps 5 and 11 of Algorithm 1 are not executed here. In these comparisons, we choose f_X such that

$$f_X(x) = \begin{cases} \frac{1}{s+1}, & x = 1, \ldots, s-1 \\ \frac{2}{s+1}, & x = s \end{cases} . \tag{2.5.3}$$

Its shape has an influence on the computation time only by the number of stages that are evaluated until the stop condition is reached, which is $F < 1 - 10^{-7}$. We choose to assign a precision of 64 bits (equivalent to a representation with 20 decimal digits) to GMP objects and to compute using 14 decimal digits with Maple XI. For the first comparison, we set $s = 200$ and consider several values of λ. Table 2.2 gives the computation times in seconds and the stages where the stop condition (last stage) is reached for some λ's. Figure 2.1 shows these computation times as functions of λ. For the second comparison, we set $\lambda = 1000$ and consider several values of s. Table 2.3 gives the computation times in

[2] Computations are made on an HP Compaq computer with an Intel Pentium CPU of 3.40 GHz and 0.99 GB of RAM.

[3] The version 4.2.2 of GMP is used for computations.

seconds and the stages where the stop condition (last stage) is reached for some values of s. Figure 2.2 shows these computation times as functions of s. For the last comparison, we decide to stop the recursion at 200 000 for each evaluation. We set $\lambda = 1000$ and consider several values of s. Table 2.4 gives the computation times in seconds for some values of s and Figure 2.3 shows these computation times as functions of s. We observe that the computation times of evaluations using GMP are much smaller than the ones using Maple even if we use more decimal digits in the representation of real numbers with GMP.

Table 2.2: Computation times and last stages for the first comparison

λ	Last stage		Computation time	
	GMP	Maple	GMP	Maple
50	9952	9952	3	17
100	16 785	16 785	6	30
500	64 682	64 682	24	117
1000	120 792	120 792	45	220
5000	548 447	548 455	204	1006
10 000	1 071 160	1 071 183	399	1966

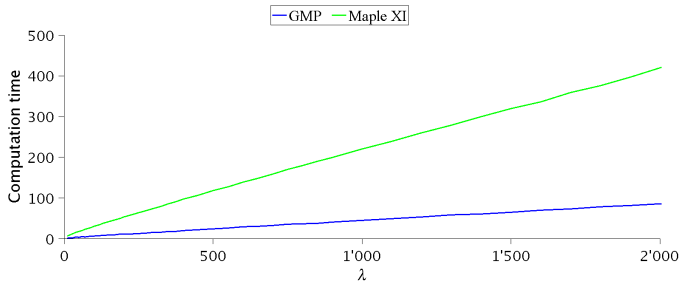


Figure 2.1: Computation times as functions of λ for the first comparison

Table 2.3: Computation times and last stages for the second comparison

s	Last stage		Computation time	
	GMP	Maple	GMP	Maple
100	60 972	60 972	11	56
200	120 792	120 792	44	222
300	180 607	180 607	98	497
400	240 422	240 417	174	885
500	300 236	300 236	271	1386
1000	599 305	599 304	1081	5734

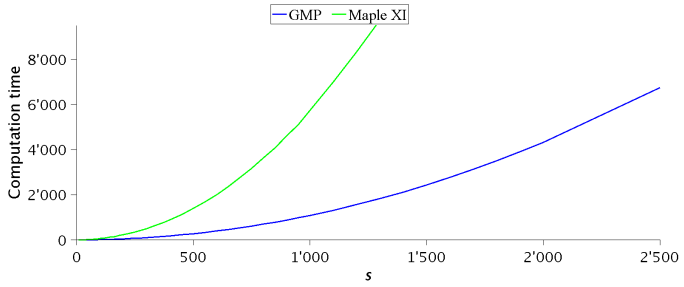


Figure 2.2: Computation times as functions of s for the second comparison

Table 2.4: Computation times for the third comparison

s	Computation time	
	GMP	Maple
50	18	91
100	36	183
200	72	366
300	109	555

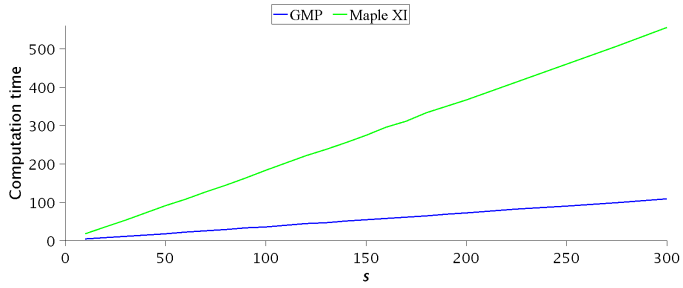


Figure 2.3: Computation times as functions of s for the third comparison

2.6 Compound binomial distributions

2.6.1 Definitions and examples

We focus now on the recursive evaluation of probability functions of compound binomial distributions. Such distributions occur in the individual risk model. In this model, the aggregate claims amount random variable S is defined as

$$S = X_1 + X_2 + \cdots + X_m, \tag{2.6.1}$$

where m is the number of policies in the portfolio and X_k is the claim amount random variable of the policy number k. We can model X_k as $X_k = I_k B_k$ where I_k is an indicator random variable and B_k is the claim amount random variable given that a claim occurs. We assume that I_k and B_k are independent and that the I_k's are mutually independent and identically distributed according to a Bernoulli distribution with parameter $0 \leq q \leq 1$ such that $\Pr[I_k = 1] = q$, $k = 1, \ldots, m$. If the random variables B_k, $k = 1, \ldots, m$, are also assumed to be independent and identically distributed, S follows a compound binomial distribution. Therefore, we can write

$$S = Y_1 + Y_2 + \cdots + Y_N, \tag{2.6.2}$$

where $Y_1, Y_2, \ldots$ are independent and identically distributed according to the same distribution as B_k and where $N = \sum_{k=1}^{m} I_k$ follows a binomial distribution with parameters m and q such that

$$\Pr[N = n] = \binom{m}{n} q^n (1-q)^{m-n}, \quad n = 0, 1, \ldots, m, \tag{2.6.3}$$

where m is a positive integer.

From (2.6.1), we can see that the recursive formula for compound binomial distributions can also be used to compute the n-fold convolution of a discrete probability function

with a positive probability mass at zero. In this case, the parameters of the binomial distribution are $m = n$ and $q = 1 - f_X(0)$. This result can be found in De Pril (1985). Finally, the computation of the probability function of a compound binomial distribution is equivalent to determining the coefficient of a polynomial (the probability generating function) raised to the power m.

To illustrate our computations in this section, we consider three examples, each of which has an individual claim amount distribution on the integers from 1 to 10.

Example 1: We consider the distribution of Example 8 of Panjer and Wang (1993, p. 249). Its probability function f_{Z_1} is given in Table 2.5. Its expected value, variance and skewness are 3.7, 5.36 and 1.007, respectively.

Example 2: We consider the random variable obtained by $Z_2 = 11 - Z_1$, where Z_1 is distributed according to the distribution of Example 1. Its probability function f_{Z_2} is given in Table 2.5. Its expected value, variance and skewness are 7.3, 5.36 and -1.007, respectively.

Example 3: For this example, we consider a skewness free distribution. Its probability function f_{Z_3} is given in Table 2.5. Its expected value and variance are 5.5 and 3.95, respectively.

Table 2.5: Individual claim amount probability functions of the three examples

x	1	2	3	4	5	6	7	8	9	10
$f_{Z_1}(x)$	0.150	0.200	0.250	0.125	0.075	0.050	0.050	0.050	0.025	0.025
$f_{Z_2}(x)$	0.025	0.025	0.050	0.050	0.050	0.075	0.125	0.250	0.200	0.150
$f_{Z_3}(x)$	0.025	0.050	0.075	0.150	0.200	0.200	0.150	0.075	0.050	0.025

2.6.2 Recurrence relations

For a binomial distribution with probability function (2.6.3) the coefficients a and b defined in (2.2.3) are

$$a = -\frac{q}{p} \quad \text{and} \quad b = (m+1)\frac{q}{p}, \tag{2.6.4}$$

where $p = 1 - q$. The recursive formula for compound binomial distributions is obtained by substituting (2.6.4) into (2.2.5). It is

$$f_S(x) = \frac{q}{px} \sum_{y=1}^{x \wedge s} ((m+1)\, y - x)\, f_X(y) f_S(x-y), \quad x = 1, 2, \ldots, ms, \tag{2.6.5}$$

with initial value $f_S(0) = p^m$. Notice that we can stop the recursion at stage ms which is the maximum value of the support of such a compound binomial distribution. As mentioned in Section 2.3, Panjer and Wang (1993) showed that (2.6.5) is unstable and could be ineffective in the recursive evaluation of f_S. The instability starts when at least one coefficient $(m+1)\, y - x$ in the sum of (2.6.5) is negative for some y. The first point where it happens is $x = m+2$, which means that (2.6.5) is stable over the range $[0, m+1]$ as pointed out by Panjer and Wang (1993).

For compound binomial distributions, we know that the event $S = ms$ is reached when we have m claims of amount s. Thus, the probability of this event is given by

$$f_S(ms) = \Pr[N = m] \prod_{i=1}^{m} \Pr[X_i = s] = (q\, f_X(s))^m. \tag{2.6.6}$$

Therefore, in contrast to the other Panjer's recursive formulas, we can calculate the relative error at ms after an evaluation using (2.6.5). This relative error can be considered as an accuracy measure since the accuracy at a given point is obtained by taking the integer part of the negative logarithm to the base ten of the relative error at this point. Observe that if the accuracy is negative the evaluation is ineffective since the related relative error is greater than one.

For a given precision of the floating-point representation, the propagation of round-off errors depends on the parameters m and q and on the shape of f_X. The relative error at ms is an increasing function of m since the number of stages increases with respect to m. We can show that it also increases with respect to q. Figure 2.4 below illustrates this property.

Another consequence of knowing the final value is that we can evaluate recursively in the backward direction i.e. from $f_S(ms)$ to $f_S(0)$. By rearranging (2.6.5), we get

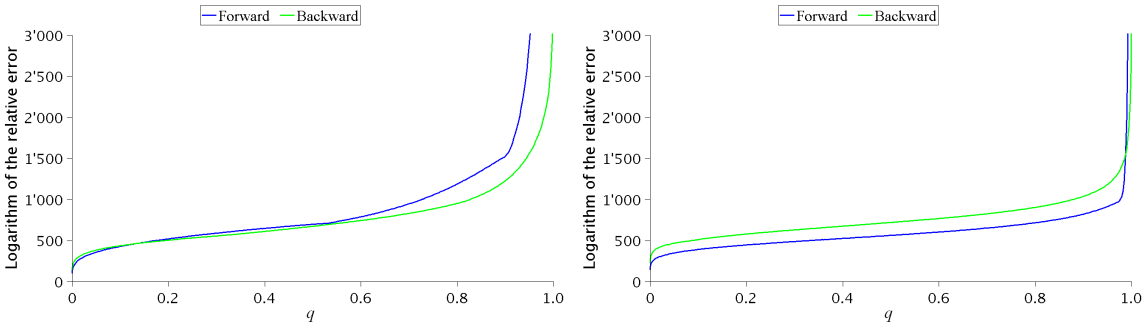
$$\begin{aligned} f_S(x) &= \frac{p}{q}\,(x+s)\, u(x) f_S(x+s) \\ &\quad + u(x) \sum_{y=1}^{s-1} (x + s - y\,(m+1))\, f_X(y) f_S(x+s-y), \end{aligned} \tag{2.6.7}$$

for $x = ms-1, ms-2, \ldots, 0$, where $u(x) = ((ms-x)\, f_X(s))^{-1}$. The initial values of (2.6.7) are (2.6.6) and $f_S(z) = 0$, $z = ms+1, ms+2, \ldots, ms+s-1$. Notice that these

initial values lead to a subordinate solution of (2.6.7) because $f_S(x) = 0$ for $x < 0$. For (2.6.7), at least one coefficient $x + s - y\,(m+1)$ is negative for some y from the point $x = ms - m - 2$ on. Therefore, it is stable over the range $[ms - m - 1, ms]$. We can measure the accuracy of this recursive evaluation from the relative error at zero. This relative error increases with respect to m and q like in the forward direction. We illustrate the behavior of this relative error with respect to q in the next subsection.

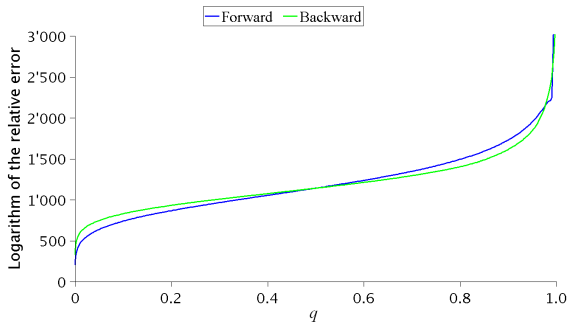
2.6.3 Forward vs backward directions

The idea is now to determine which direction is preferable for the recursive evaluation of the probability function of a given compound binomial distribution over its whole support. Figure 2.4 shows the logarithms to the base ten of the relative errors at the last evaluation



(a) Example 1

(b) Example 2



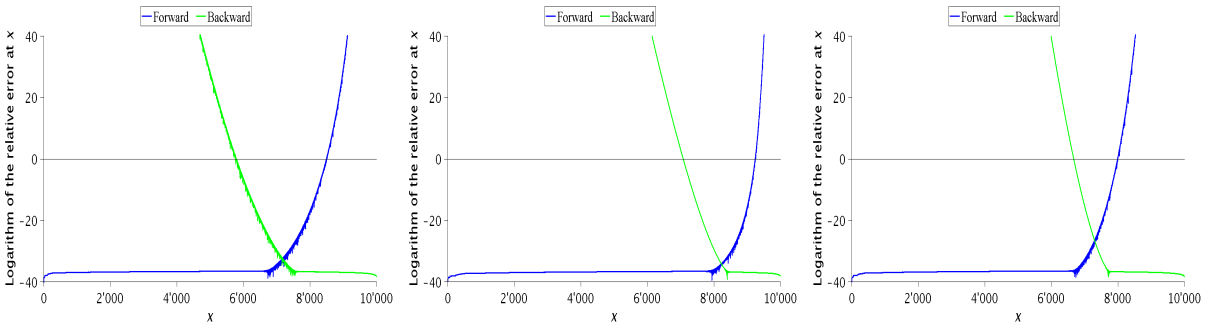
(c) Example 3

Figure 2.4: Logarithms of the relative errors at the last evaluation points as functions of q for both directions with $m = 1000$ and a precision of 128 bits

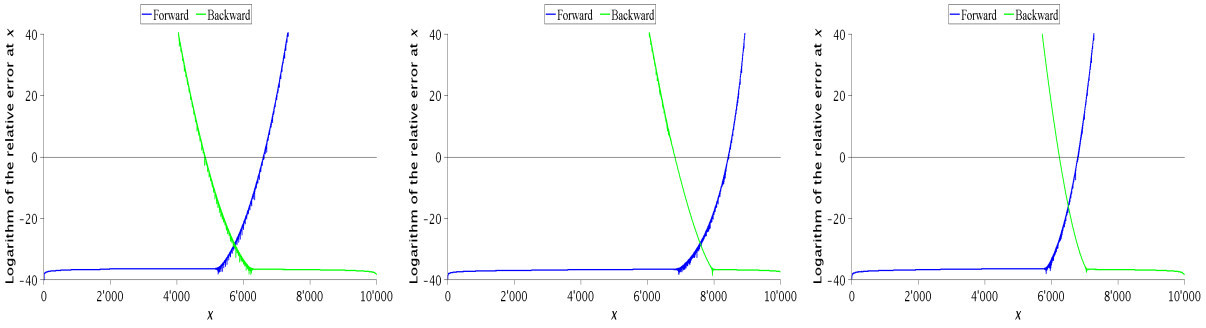
points in the forward and backward recursive evaluations as functions of q and for the three examples. In order to make these three graphs easier to understand, we choose

$m = 1000$ and the precision (128 bits) such that the recursive evaluations are ineffective. We observe that the relative error at the last evaluation point increases with respect to q for each direction. We also remark that the evaluation in the forward direction is preferable for small values of q, while the evaluation in the backward direction is preferable when the value of q is near one. This property generally holds for every distribution of X. We denote by $\hat{q}$, the value of q where the two curves cross. This value depends to a large extent on the characteristics of the individual claim amount distribution. When f_X has a positive skewness, $\hat{q}$ is usually smaller than 0.5, while it is generally greater than 0.5 if f_X has a negative skewness. Moreover, for some distributions, one direction is preferable for almost all q. Unfortunately, there is no simple rule to determine $\hat{q}$ from f_X. If it were the case, it would be recommended to evaluate in the forward direction if $q < \hat{q}$ and in the backward direction if $q > \hat{q}$. We can conclude that the forward evaluation is better for small values of q and if f_X has a negative skewness, while the backward evaluation is preferable for large values of q and if f_X has a positive skewness.

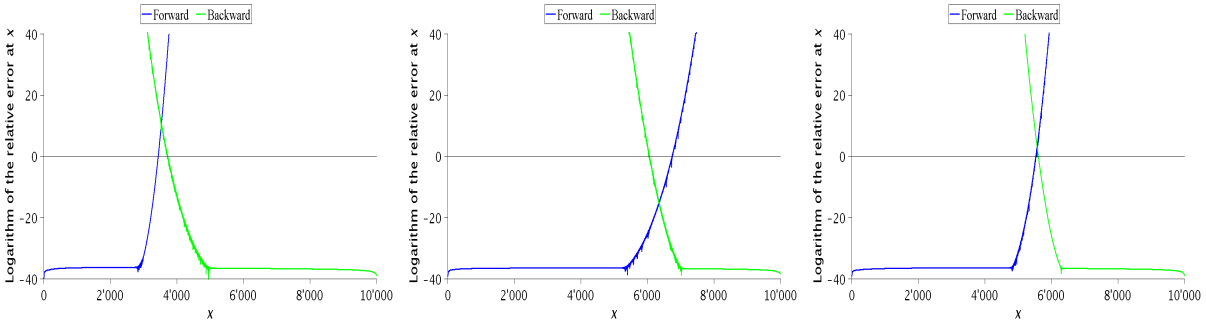
Panjer and Wang (1993) proposed a combined usage of both directions. It consists in evaluating f_S in both directions for the most part of the support. If we use a sufficient precision the first digits of both evaluations will be the same for some points in an interval. If such an interval exists an accurate probability function of S is given by the forward evaluation over the left side of the support and by the backward evaluation over its right side. This method has the advantage that the precision which is necessary to get an accurate evaluation is smaller than the one for an evaluation in only one direction. However, we do not know beforehand the location of such an interval and if it exists for the precision used in the evaluation. Moreover, if we do not obtain the same first digits for any point, we will not be able to know how many bits that we have to add in order to guarantee an accurate evaluation since we do not have any accuracy measure. This means that we may evaluate several times before having an adequate accuracy of the considered probability function. Figure 2.5 shows the evolution of the relative errors as we go along with the recursions in both directions for the three examples and three values of q. A precision of 128 bits is assigned to floating-point objects and we choose $m = 1000$. In order to calculate these relative errors, we need to know the exact values. They are evaluated recursively with a precision of 10 016 bits which can be considered as the "infinite" precision. This precision which corresponds to a representation of real numbers with more than 3000 decimal digits is chosen such that we obtain an evaluation with a more than sufficient accuracy. The interval over which the first digits of the evaluations are equal in both directions is the set of points where both curves are below



(a) Example 1 with q=0.05 (b) Example 2 with q=0.05 (c) Example 3 with q=0.05



(d) Example 1 with q=0.5 (e) Example 2 with q=0.5 (f) Example 3 with q=0.5



(g) Example 1 with q=0.95 (h) Example 2 with q=0.95 (i) Example 3 with q=0.95

Figure 2.5: Evolution of the logarithms of the relative errors as we go along with the recursions in both directions with $m = 1000$ and a precision of 128 bits

zero. We can observe that it becomes smaller as q increases and it even does not exist for Examples 1 and 3 with $q = 0.95$ (see Figures 2.5g and 2.5i). For these two cases, we have to use a greater precision in order to obtain such an interval. Furthermore, if we increase m we will have more cases where this interval does not exist. From Figure 2.5, we also remark that the forward and backward evaluations are accurate over a smaller and greater range as q increases, respectively. Thus, the interval described above moves to the left as q increases.

2.6.4 Which precision is necessary?

From now on, we consider only evaluations in the forward direction and similar results hold for evaluations in the backward direction. We start to use the computational properties of GMP, especially the fact that we can change easily the precision of an object in a program. We come back to the illustration that we did in Section 2.3 and show numerically that the recursive evaluation is ineffective if we compute the first stage with a "finite" precision. Table 2.6 gives, for the three examples, the relative error at ms of an evaluation using a

Table 2.6: Relative error at ms according to an evaluation using a "finite" precision for the first k stages

k	Relative error at ms		
	Example 1	Example 2	Example 3
Never	$6.73871 \cdot 10^{-1838}$	$3.92434 \cdot 10^{-1940}$	$2.34821 \cdot 10^{-1461}$
0	$8.64004 \cdot 10^{-22}$	$8.64004 \cdot 10^{-22}$	$8.64004 \cdot 10^{-22}$
1	$3.44691 \cdot 10^{218}$	$4.48025 \cdot 10^{360}$	$3.24581 \cdot 10^{590}$
10	$3.11429 \cdot 10^{230}$	$2.63760 \cdot 10^{362}$	$1.14339 \cdot 10^{593}$
100	$7.20225 \cdot 10^{280}$	$1.16377 \cdot 10^{375}$	$8.88992 \cdot 10^{614}$
1000	$1.07758 \cdot 10^{447}$	$8.12733 \cdot 10^{418}$	$4.79148 \cdot 10^{734}$
10 000	$2.58066 \cdot 10^{609}$	$4.95376 \cdot 10^{507}$	$4.51330 \cdot 10^{985}$

precision of 64 bits up to the k-th stage and using a precision of 8192 bits for the following stages. We choose $m = 1000$ and $q = 0.3$, which gives $ms = 10\,000$. The precision of 8192 bits is considered to be the "infinite" precision since it is much greater than the one needed to have an accurate evaluation. From (2.4.4), we know that the precisions of 64 and 8192 bits represent real numbers with 20 and 2467 decimal digits, respectively.

When $k = 0$, we represent the initial value $f_S(0)$ using a precision of 64 bits and then we evaluate recursively each stage with a precision of 8192 bits. We observe that this evaluation leads to accurate values since the accuracy of the last point is of 21 decimal digits for each of the three examples. The magnitude of this relative error comes from the round-off error made in the representation of $f_S(0)$ with a "finite" precision. Precisions of 8192 and 64 bits are assigned to each object involved in recursive evaluations for the "never" case and when $k = 10\,000$, respectively. The evaluation takes approximatively 15 seconds for each case except for $k = 10\,000$ for which it takes only one second.

We can notice that there is a relation between the relative errors obtained in the "never" case and when $k = 10\,000$. We observe that the difference between the exponents of the relative errors for each example is approximatively equal to the difference between the numbers of decimal digits used in the floating-point representation for both precisions. In Example 1, the computation with a precision of 8192 bits leads to an accuracy of 1837 digits at ms. If we want to obtain an accuracy of only 10 digits at this point we can set a smaller precision in order to have a faster evaluation. The difference between the obtained and the desired accuracies is 1827 digits. Thus, what happens if we reduce the floating-point representation by 1827 decimal digits? The consequence is that the relative errors of the first points of the perturbed solution $\tilde{h}$, discussed in Section 2.3, are approximatively multiplied by 10^{-1827}. Then, the relative errors of further points are also approximatively multiplied by 10^{-1827} including the one at ms which should be of the magnitude of 10^{-11}. From (2.4.7), this corresponds to assign a precision of 2123 bits to the objects. Due to code optimization, setting a precision of 2123 bits with GMP assigns actually a precision of 2144 bits. If this precision is set to each object we obtain a relative error at ms of $1.90147 \cdot 10^{-17}$ for a computation time of 2 seconds. However, we can also start from the relative error obtained in the "never" case of Table 2.6. For Example 1, this relative error is greater than one and its logarithm, 609.41, can be considered as a lack of decimal digits used in the floating-point representation. If we add the 10 desired digits of accuracy, it follows from (2.4.7) that we should use a precision of 2122 bits. It is one bit smaller than the precision obtained starting from the evaluation using an original precision of 8192. This relation holds for every distribution of X.

We have now a method that guarantees an accurate recursive evaluation. It consists in evaluating recursively f_S using the lowest precision possible ($r = 64$ bits) and calculating the relative error at ms denoted by η. This evaluation is accurate if η is smaller than $10^{-(v+1)}$, where v is the desired accuracy. If it is not the case, we determine the precision,

$\hat{r}$, which is necessary to obtain an accurate evaluation by

$$\hat{r} \quad = \quad r + \lceil \log_2 (\eta \cdot 10^v) \rceil \,, \tag{2.6.8}$$

which follows from (2.4.7). The GMP objects have actually a precision of

$$\hat{r}_{GMP} = 32 \left\lceil \frac{\hat{r}}{32} \right\rceil . \tag{2.6.9}$$

Table 2.7 contains the values of η, $\hat{r}$ and $\hat{r}_{GMP}$ in addition to the relative error ($\hat{\eta}$) obtained

Table 2.7: Values of η, $\hat{r}$, $\hat{r}_{GMP}$ and $\hat{\eta}$ if $m = 1000$ and $q = 0.3$

Variable	Example 1	Example 2	Example 3
η	$2.58066 \cdot 10^{609}$	$4.95376 \cdot 10^{507}$	$4.51330 \cdot 10^{985}$
$\hat{r}$	2122	1784	3372
$\hat{r}_{GMP}$	2144	1792	3392
$\hat{\eta}$	$1.90147 \cdot 10^{-17}$	$1.13409 \cdot 10^{-13}$	$1.57129 \cdot 10^{-16}$

at ms for a recursive evaluation with a precision of $\hat{r}_{GMP}$ calculated for $v = 10$. This procedure is summarized in Algorithm 2. Figure 2.6 shows the evolution of the relative errors as we go along with the recursive evaluations using precisions of 64 and $\hat{r}_{GMP}$ bits for the three examples. We observe that the difference between the logarithms of the relative errors obtained using both precisions is equal to $\hat{d}_{GMP} - 20$ for each point, where $\hat{d}_{GMP}$ is the number of decimal digits used in the floating-point representation with a precision of $\hat{r}_{GMP}$. This means that the relative error is multiplied by $10^{20-\hat{d}_{GMP}}$ at each point between the evaluations using 64 and $\hat{r}_{GMP}$ bits. These two recursive evaluations are parts of Algorithm 2.

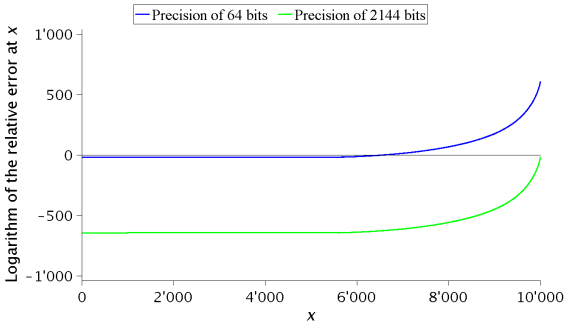
In Section 2.4, we wrote that there is no logarithm function defined in GMP. However, we can notice that at step 15 of Algorithm 2, a logarithm to the base two has to be computed in order to determine the precision needed to get an accurate evaluation. We can also observe that we need the smallest integer greater than or equal to this logarithm. Nevertheless, there exists a GMP function that we can adjust to find this integer. This function, called `mpf_class_get_d_2exp`, returns a `double` variable `d` and admits two arguments: a pointer to a `signed long int` variable `exp` and a floating-point number variable `op`. The value of `d` is the solution of $op = d \cdot 2^{exp}$ with $0.5 \leq d < 1$. The value of the exponent is stored to `exp`. This value corresponds to the ceiling function of the

Algorithm 2: Recursive evaluation of a compound binomial distribution by finding the needed precision

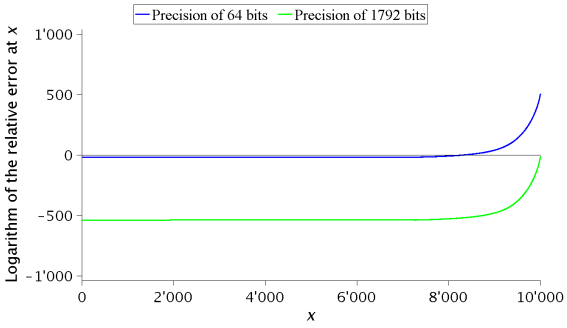
1. Set the default precision r to 64 bits
2. Declare a table f with elements from 0 to $s-1$ and an intermediate variable sum
3. Define a table g with elements from 1 to s for f_X
4. Define m and q and set $p = 1 - q$
5. Set $f[0] = p^m$ and $h = (q \cdot g[s])^m$
6. Store $f[0]$ with v decimal digits
7. For $i = 1$ to $m * s$
8. $\quad sum = 0$
9. $\quad$ For $k = 1$ to $\min(i, s)$
10. $\quad\quad sum = sum + ((m+1)\,k - i) \cdot g[k] \cdot f[i - k \bmod s]$
11. $\quad f[i \bmod s] = \frac{sum \cdot q}{p \cdot i}$
12. $\quad$ Store $f[i \bmod s]$ with v decimal digits
13. Calculate $\eta = \left|\frac{h - f[0]}{h}\right|$
14. If $\eta < 10^{-(v+1)}$ then stop
15. Set a precision of $r + \lceil \log_2(\eta \cdot 10^v) \rceil$ to each element of f and g, to p, q and sum
16. Redefine p, q and g
17. Redo steps 5 to 12

logarithm to the base two of *op* if $d \neq 0.5$. If $d = 0.5$, the ceiling function of the logarithm to the base two of *op* is equal to $exp - 1$. Therefore, in order to obtain a function that computes the ceiling function of the logarithm to the base two of a number, we can create a function, which uses the function `mpf_class_get_d_2exp` and add a condition on the value of `d` to adapt the value of `exp`. This function can be:

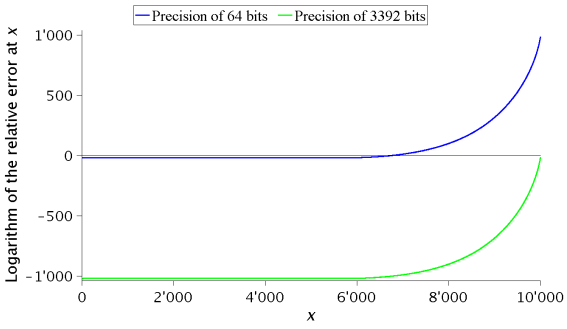
```
signed long int ceil_log_2(mpf_class number){
    double d; signed long int exp;
    d = mpf_get_d_2exp (&exp, number.get_mpf_t());
    if (d == 0.5)
        exp -= 1;
    return exp;
}
```



(a) Example 1



(b) Example 2



(c) Example 3

Figure 2.6: Evolution of the relative errors of the recursive evaluation using two different precisions

2.6.5 Precision management

Looking at Algorithm 2, we observe that we have to set a new precision to each object at step 15 and then to redefine each one according to this new precision at step 16. What happens if we assign a precision $\overline{r}$ with $\overline{r} < \hat{r}$ to the objects representing p, q and f_X? Although we represent their value with a smaller precision, their representation is still close to their exact value. Therefore, we have a new compound binomial distribution whose probability function is a very good approximation of the exact one. We have to choose $\overline{r}$ in function of the desired accuracy, in fact it must be greater than $\lceil (v-1)\log_2 10 \rceil$.

Such an approximation is interesting since time can be saved due to the use of a smaller precision. The computation time of the recursive evaluation and the corresponding relative error obtained in five cases described below and for each of the three examples can be found in Table 2.8. In order to be easier to compare the computation times, we increase them by using a greater parameter m chosen to be 10 000. The parameter q remains

Table 2.8: Relative errors and computation times in five cases

Case	Example 1		Example 2		Example 3	
	Relative error	Time	Relative error	Time	Relative error	Time
(1)	$9.64475 \cdot 10^{-21}$	78	$9.64475 \cdot 10^{-21}$	79	$9.64475 \cdot 10^{-21}$	78
(2)	$4.70198 \cdot 10^{-34}$	884	$7.83663 \cdot 10^{-35}$	883	$4.70198 \cdot 10^{-34}$	884
(3)	$9.64475 \cdot 10^{-21}$	2625	$9.64475 \cdot 10^{-21}$	2623	$9.64475 \cdot 10^{-21}$	2619
(4)	$1.54837 \cdot 10^{-13415}$	3432	$6.76541 \cdot 10^{-14435}$	3431	$2.66930 \cdot 10^{-9648}$	3427
(5)	$1.14991 \cdot 10^{6295}$	4	$1.82767 \cdot 10^{5275}$	4	$1.68812 \cdot 10^{10061}$	4

equal to 0.3. In case (1), a precision of 64 bits is assigned to the objects representing p, q and f_X. In case (2), a precision of 64 bits is set only to the objects representing f_X. In case (3), a precision of 64 bits is set only to the objects representing p and q. In case (4), each object has a precision of 65 536 bits, the "infinite" precision which was assigned to the other objects in the three previous cases. In case (5), each object has a precision of 64 bits. This last case corresponds to the first evaluation of Algorithm 2 that is used to determine $\hat{r}$. From Table 2.8, we observe that we can save a lot of time by setting a smaller precision to the "non-recursive" objects while keeping an adequate accuracy.

To save much more time, we can even split step 10 of Algorithm 2 in two parts. The first one consists in doing the multiplication of the objects which accept a smaller precision and store the product in an intermediate object `w` declared with a precision of $\overline{r}$. The second part concerns the accumulation of the products of the value of `w` and the probability function of S at the right point in the object `sum`. Algorithm 3 is obtained from Algorithm 2 with the use of a precision of $\overline{r}$ set to the objects representing p, q and f_X and with the inclusion of the split of multiplications. Notice that in most of the applications, steps 17 and 18 can be removed because $\overline{r} = 64$ is generally sufficient to guarantee the desired accuracy. The cases (1) and (2) of Table 2.8 are reproduced in Table 2.9 with the inclusion of the split of multiplications. Table 2.10 contains the values of η, $\hat{r}$ and $\hat{r}_{GMP}$ with $v = 10$ in addition to the relative error ($\hat{\eta}$) obtained at ms with a recursive evaluation using Algorithm 3. The computation time (Time 1) of step 19 of Algorithm 3 can also be found in Table 2.10 as well as the computation time (Time 2) of the evaluation using a precision of $\hat{r}_{GMP}$ assigned to each object. The difference between these two computation times is the time that we save by using Algorithm 3 instead of Algorithm 2 for each example.

Algorithm 3: Recursive evaluation of a compound binomial distribution by finding the needed precision with the inclusion of the split of multiplications

1. Set the default precision r to 64 bits
2. Declare a table f with elements from 0 to $s-1$ and the variables w and sum
3. Define a table g with elements from 1 to s for f_X
4. Define m and q and set $p = 1 - q$
5. Set $f[0] = p^m$ and $h = (q \cdot g[s])^m$
6. Store $f[0]$ with v decimal digits
7. For $i = 1$ to $m * s$
8. $\quad sum = 0$
9. $\quad$ For $k = 1$ to $\min(i, s)$
10. $\quad\quad w = ((m+1)\,k - i) \cdot g[k]$
11. $\quad\quad sum = sum + w \cdot f[i - k \bmod s]$
12. $\quad f[i \bmod s] = \frac{sum \cdot q}{p \cdot i}$
13. $\quad$ Store $f[i \bmod s]$ with v decimal digits
14. Calculate $\eta = \left|\frac{h - f[0]}{h}\right|$
15. If $\eta < 10^{-(v+1)}$ then stop
16. Set a precision of $r + \lceil \log_2(\eta \cdot 10^v) \rceil$ to each element of f and to sum
17. Set a precision of $\max(\lceil v \log_2 10 \rceil, 64)$ to each element of g, to p, q and w
18. Redefine p, q and g
19. Redo steps 5 to 13

Table 2.9: Relative errors and computation times in two cases with the inclusion of the split of multiplications

Case	Example 1		Example 2		Example 3	
	Relative error	Time	Relative error	Time	Relative error	Time
(1)	$9.64677 \cdot 10^{-21}$	68	$9.64509 \cdot 10^{-21}$	67	$9.64677 \cdot 10^{-21}$	67
(2)	$2.01948 \cdot 10^{-24}$	874	$3.36581 \cdot 10^{-25}$	874	$2.01948 \cdot 10^{-24}$	873

Table 2.10: Values of η, $\hat{r}$, $\hat{r}_{GMP}$ and $\hat{\eta}$ with computation times if $m = 10\,000$ and $q = 0.3$

Variable	Example 1	Example 2	Example 3
η	$1.14991 \cdot 10^{6295}$	$1.82767 \cdot 10^{5275}$	$1.68812 \cdot 10^{10061}$
$\hat{r}$	21 009	17 622	33 520
$\hat{r}_{GMP}$	21 024	17 632	33 536
$\hat{\eta}$	$1.01105 \cdot 10^{-14}$	$9.94587 \cdot 10^{-15}$	$1.68333 \cdot 10^{-15}$
Time 1	25	21	37
Time 2	622	468	1235

We might put here a figure similar to Figure 2.6 on the evolution of the relative error of the two recursive evaluations of Algorithm 3. However, it would not be so nice since the relative error is always on the same magnitude during the computation of step 19 of Algorithm 3.

2.6.6 Effects of m on stability

There is a last parameter for which we did not discuss its influences on the stability of recursive evaluation yet. It is the parameter m. From Tables 2.7 and 2.10, we observe that the needed precision given a parameter m, $\hat{r}\,(m)$, is approximatively 10 times greater in the case $m = 10\,000$ than in the case $m = 1000$ for each of the three examples. In both tables the desired accuracy is 10 digits, but to compare these precisions it is better to use $v = 0$. Table 2.11 gives the values of $\hat{r}\,(m)$ for $v = 0$ for the same distributions as in

Table 2.11: Values of $\hat{r}\,(m)$ for two values of m and $v = 0$

m	$\hat{r}\,(m)$		
	Example 1	Example 2	Example 3
1000	2089	1751	3339
10 000	20 976	17 589	33 487

Tables 2.7 and 2.10. We remark that the ratio is actually a bit greater than 10, the ratio between the two values of m. In reality, this is true for almost every distribution of X. This can be interpreted because multiplying the parameter $m = l$ by a factor c amounts to

the same as finding the coefficient of a polynomial raised to the power $c\,l$. Moreover, the recursive evaluation requires c times more stages and is stable over the range $[0, c\,l + 1]$, approximatively c times the one in the case $m = l$.

In order to find a relation between m and the needed precision, we do a multiple linear regression based on the needed precisions for nine values of m and 21 distributions of X. These precisions follow from (2.6.8) with $r = 64$ and $v = 0$ but without taking the ceiling function. The two explanatory variables used in the regression are c and the product of c and $\hat{r}(l)$, while the dependent variable is $\hat{r}(cl)$. We obtain the following relation

$$\hat{r}\,(c\,l) \;=\; 9.99029c + c\hat{r}\,(l) - 9.75509. \tag{2.6.10}$$

Notice that this relation is better for small values of c and that $\hat{r}(cl)$ may be insufficient for some distributions of X for some values of c. In order to increase the probability that this precision is sufficient, we add a margin proportional to $c - 1$ such that (2.6.10) becomes

$$\hat{r}\,(c\,l) \;=\; c \cdot \hat{r}\,(l) + 13\,(c-1)\,. \tag{2.6.11}$$

We have now a more efficient method which consists in evaluating recursively with a parameter $m = l$ and a precision of 64 bits and calculating η, the relative error at ls. It follows from (2.6.8) and (2.6.11) that

$$\hat{r}\,(c\,l) \;=\; \lceil c\,(r + \log_2(\eta)) + 13\,(c-1) + v\log_2 10 \rceil\,, \tag{2.6.12}$$

which leads to

$$\hat{r}_{GMP}\,(c\,l) \;=\; 32\left\lceil \frac{\hat{r}\,(c\,l)}{32} \right\rceil. \tag{2.6.13}$$

Then, we evaluate f_S with the parameter $m = c\,l$ and a precision of $\hat{r}_{GMP}\,(c\,l)$ in order to obtain accurate values. This procedure is summarized in Algorithm 4. For a very small number of distributions the margin chosen in (2.6.11) is still not sufficient. However, if this happens we can use the other direction to evaluate the remaining points until we obtain a stage where the required accuracy is reached. Remark that the number of additional evaluation points is very small since $\hat{r}_{GMP}\,(c\,l)$ is close to the precision really needed.

Table 2.12 shows the values of $\hat{r}\,(c\,m)$ for $c = 2$, 5 and 10, an original parameter $m = 1000$ and a desired accuracy of 10 digits. The value of q remains equal to 0.3. This table also contains the needed precision $\hat{r}$ calculated using (2.6.8) where η is the relative error obtained at the last evaluation point with a precision of 64 bits and $m = 1000c$. Notice

Algorithm 4: Recursive evaluation of a compound binomial distribution by finding the needed precision from a first evaluation with a smaller value of m

1. Set the default precision r to 64 bits
2. Declare a table f with elements from 0 to $s-1$ and the variables w and sum
3. Define a table g with elements from 1 to s for f_X
4. Define m and q and set $p = 1 - q$
5. Set $f[0] = p^m$ and $h = (q \cdot g[s])^m$
6. For $i = 1$ to $m * s$
7. $\quad sum = 0$
8. $\quad$ For $k = 1$ to $\min(i, s)$
9. $\quad\quad w = ((m+1)\,k - i) \cdot g[k]$
10. $\quad\quad sum = sum + w \cdot f[i - k \bmod s]$
11. $\quad f[i \bmod s] = \frac{sum \cdot q}{p \cdot i}$
12. Calculate $\eta = \left|\frac{h - f[0]}{h}\right|$
13. Set a precision of $\lceil c\,(r + \log_2 \eta) + 13\,(c-1) + v \log_2 10 \rceil$ to each element of f and to sum
14. Set a precision of $\max\left(\lceil v \log_2 10 \rceil, 64\right)$ to each element of g, to p, q and w
15. Redefine p, q and g
16. Redo steps 5 to 11 by setting $m = cm$ and storing values with v decimal digits

Table 2.12: Comparisons between $\hat{r}\,(1000c)$ and $\hat{r}$

c	Example 1		Example 2		Example 3	
	$\hat{r}\,(c\,m)$	$\hat{r}$	$\hat{r}\,(c\,m)$	$\hat{r}$	$\hat{r}\,(c\,m)$	$\hat{r}$
2	4224	4221	3548	3544	6723	6724
5	10 528	10 518	8838	8821	16 777	16 772
10	21 035	21 012	17 656	17 623	33 533	33 522

that the real precision set to GMP objects is the same for six cases over the nine considered in Table 2.12. For the three other cases, the precision assigned to GMP objects according to (2.6.13) is 32 bits greater than the one obtained by (2.6.9). Example 1 with $c = 10$ and Example 2 with $c = 5$ and $c = 10$ are these three cases for which the difference comes from the margin chosen in (2.6.11). Figure 2.7 shows the evolution of the relative error in the evaluation according to Algorithm 4 for the three examples with $m = 1000$ and $c = 2$. In order to obtain nice curves, the second evaluation of Algorithm 4 is done by setting a precision of $\hat{r}\,(c\,m)$ to each object. If the precision of objects representing p, q and f_X was 64 bits, the relative error would be always on the same magnitude as we go along the recursive evaluation.

2.7 Further remarks

Some extensions of Panjer's recursion can be found e.g. in Sundt (1992) and Sundt and Jewell (1981). The latter derived a recursive formula when (2.2.3) holds from an integer k on such that

$$p_n = \left(a + \frac{b}{n}\right) p_{n-1}, \quad n = k+1, k+2, \ldots, \tag{2.7.1}$$

with $p_n \geq 0$ for $n = 0, 1, \ldots, k-1$ and $p_k > 0$, which gives

$$\begin{aligned} f_S(x) &= p_1 f_X(x) + \sum_{n=2}^{k} \left(p_n - \left(a + \frac{b}{n}\right) p_{n-1}\right) f_X^{*n}(x) \\ &\quad + \sum_{y=1}^{x-1} \left(a + b\,\frac{y}{x}\right) f_X(y) f_S(x-y), \quad x = 1, 2, \ldots, \end{aligned} \tag{2.7.2}$$

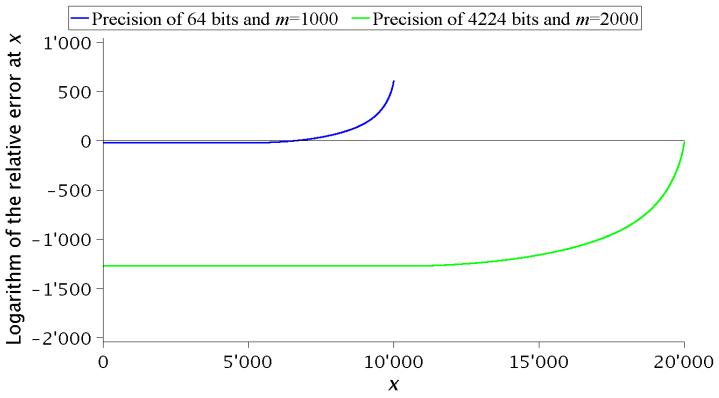
with $f_S(0) = p_0$. Sundt (1992) extends Panjer's recursion for the distributions of N that satisfy

$$p_n = \sum_{j=1}^{k} \left(a_j + \frac{b_j}{n}\right) p_{n-j}, \quad n = 1, 2, \ldots, \tag{2.7.3}$$

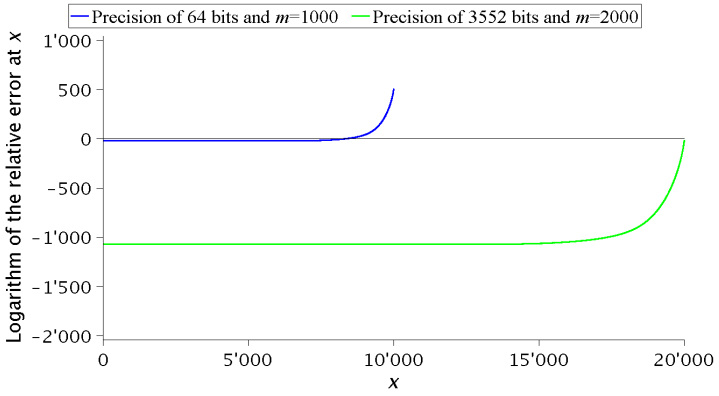
with $p_n = 0$ for $n < 0$ and $p_0 > 0$, which leads to

$$f_S(x) = \sum_{y=1}^{x} f_S(x-y) \sum_{j=1}^{k} \left(a_j + \frac{b_j}{j}\frac{y}{x}\right) f_X^{*j}(y), \quad x = 1, 2, \ldots, \tag{2.7.4}$$

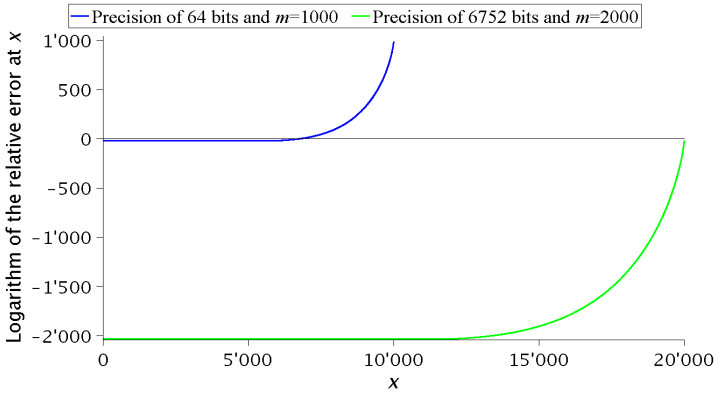
with initial value $f_S(0) = p_0$. The use of a smaller precision for the floating-point representation of the parameters of N and for the representation of f_X also gives an accurate



(a) Example 1



(b) Example 2



(c) Example 3

Figure 2.7: Evolution of the relative error evaluating according to Algorithm 4

evaluation using (2.7.2) or (2.7.4). This method can be generalized to any recursive evaluation. However, we still need to know an exact value for f_S in order to be able to determine the needed precision after a first evaluation.

Another method to compute the probability function of S is to use the fast Fourier transform (FFT) that can be applied for any random variable N with probability generating function P_N. It consists in evaluating $f_S(0), f_S(1) \ldots, f_S(n-1)$ using the inverse discrete Fourier transform by

$$f_S(x) = \frac{1}{n} \sum_{k=0}^{n-1} \phi_S \left(\frac{2\pi k}{n} \right) e^{\frac{-i2\pi xk}{n}}, \quad x = 0, 1, 2, \ldots, n-1, \tag{2.7.5}$$

where $i = \sqrt{-1}$ and ϕ_S is the characteristic function of S which is defined by

$$\phi_S(z) \;\; = \;\; \mathrm{E}\left[e^{izS}\right] = P_N\left(\phi_X(z)\right). \tag{2.7.6}$$

The terms $\phi_S\left(\frac{2\pi k}{n}\right)$ of (2.7.5) can be determined by the substitution of the discrete Fourier transform

$$\phi_X(z) = \sum_{k=0}^{n-1} f_X(k) e^{\frac{i2\pi zk}{n}}, \tag{2.7.7}$$

into (2.7.6). If n is chosen to be a power of two, (2.7.5) and (2.7.7) can be computed efficiently using an FFT algorithm. However, this method introduces an aliasing error which is equal to

$$f_S(x) - \Pr\left[S = x\right] = \sum_{j=1}^{\infty} \Pr\left[S = x + jn\right], \quad x = 0, 1, 2, \ldots, n-1, \tag{2.7.8}$$

where $\Pr\left[S = s\right]$ denotes the exact value of the probability function of S at s. We refer to Bühlmann (1984) for a comparison between evaluations using Panjer's recursion and a method based on the fast Fourier transform algorithm. An exponential tilting procedure was proposed by Grübel and Hermesmeier (1999) in order to reduce the aliasing error (see also Embrechts and Frei (2009)). Nevertheless, using the floating-point representation, this error is of the magnitude of 10^{-d+1} for each point over the interval $[0, n-1]$, where d is the number of exact decimal digits used in the floating-point representation. Therefore, when the exact probability function is smaller than 10^{-d+1} at a point, the absolute value of its evaluation is approximatively equal to 10^{-d+1}. The use of an exponential tilting decreases the accuracy of f_S since the aliasing error occurs on the tilted probability function. Finally, the use of a smaller precision for the floating-point representation of the parameters of N and for the representation of f_X in the implementation of this method

gives aliasing errors of the same magnitude as when all variables are represented using this smaller precision. Therefore, the gain of time is useless since the accuracy of the results decreases.

2.8 Conclusion

The use of an efficient computational tool like GMP is essential for evaluations involving a large number of arithmetic operations. GMP has to be used to evaluate recursively the probability function of compound distributions especially when their expected value is large. It also offers several useful functions which allow us to work with different precisions assigned to objects and it avoids the use of scaling functions. The gain of time resulting from the use of a smaller precision of the floating-point representation of "non-recursive" objects is considerable without loosing any useful accuracy. This is why we have to use GMP and its precision management properties in order to obtain an efficient evaluation of a subordinate solution of a recurrence relation like the one for the probability function of compound binomial distributions.

Chapter 3

From approximations of De Pril transforms to approximations of t-th order cumulative distribution functions

3.1 Introduction

The term *De Pril transform* was introduced by Sundt (1995) as a reference to a function derived in De Pril (1989, p. 11). Given a value of a probability function, the De Pril transform defines uniquely this probability function and there exist recursive formulas from one function to the other. The main result in relation to De Pril transforms is that the De Pril transform of a convolution of functions is the sum of the De Pril transforms of these functions. Therefore, they are useful when an efficient evaluation of the probability function of a convolution is required. Further results on De Pril transforms can be found in Sundt (1998) and Sundt and Ekuma (1999).

Convolutions appear frequently in actuarial applications, for example to model the aggregate claims amount of a portfolio of insurance policies like in the collective risk model or in the individual risk model. In the former model, the aggregate claims amount is modelled by the sum of the amounts of a random number of claims. In this case, recursions for the evaluation of the aggregate claims amount probability function can be found in Panjer (1981). In the individual risk model, the aggregate claims amount is modelled by the sum of independent random variables where one random variable represents the claim amount

of one policy. This model can be generalized to the evaluation of the probability function of a convolution of probability functions of independent random variables. Several exact or approximative recursions for the probability function of this model are discussed in the actuarial literature (see e.g. De Pril (1986b, 1988, 1989), Dhaene and Vandebroek (1995), Hipp (1985, 1986), Kornya (1983) and Waldmann (1994)). All these recursive evaluations are reviewed in Sundt (2002) and Sundt and Vernic (2009). In comparison to an evaluation involving convolutions, recursive evaluations are good strategies to save time since they reduce significantly the number of operations. The main approximations generally decrease again this number to save much more time while keeping a required accuracy in evaluations. We refer to Dhaene et al. (2006) for comparisons between the numbers of operations of the different exact evaluations and approximations.

Dhaene and Sundt (1998) developed error bounds for the distribution function and the stop-loss transform of several classes of distributions resulting from an approximation of their De Pril transforms. Other results on error bounds in connection with De Pril transforms can be found in De Pril (1989), Dhaene and De Pril (1994), Dhaene and Sundt (1997) and Sundt et al. (1998). Their approach consists in evaluating the distribution function and the stop-loss transform from an approximation of the probability function that is computed from an approximative De Pril transform. Their more accurate error bounds depend on the evaluations, which means that we know their value only after having found the approximation of the desired quantities. Moreover, for some of these error bounds we have to evaluate the probability function over its whole support which is numerically impossible in most cases.

A recursive formula for the evaluation of the t-th order cumulative distribution function based on the De Pril transform of the probability function is derived in Dhaene et al. (1999). They pointed out that frequently calculated quantities like distribution functions or stop-loss transforms are obtained directly from a given order of such functions. Sundt (1999b) expressed this recursion for discrete uniform distributions. Recursive formulas for the t-th order cumulative distribution function and the t-th order cumulative tail function of compound Poisson distributions are developed in Antzoulakos and Chadjiconstantinidis (2004) and improved in Chadjiconstantinidis and Pitselis (2009).

In Section 3.2, we define the notion of De Pril transforms and discuss the utility of recursive evaluations of t-th order cumulative distribution functions. In Section 3.3, we express the exact error of the approximation of the t-th order cumulative distribution function resulting from a De Pril transform approximation. Theoretical error bounds for such approximations are derived in Section 3.4. In Section 3.5, we expose error bounds

in relation to stop-loss contracts. Some comparisons between these error bounds and the ones developed by Dhaene and De Pril (1994) are made in Section 3.6. Finally, we apply the results to some frequently used approximations and do numerical applications.

In the sequel, we assume that the functions are defined on the nonnegative integers. The notations $a \geq b$ and $a > b$ are used to simplify the facts that $a = b, b+1, b+2, \ldots$ and $a = b+1, b+2, \ldots$, respectively. We also assume that $\sum_{k=a}^{b} h(k) = 0$ whenever $b < a$.

3.2 De Pril transforms and recursions for t-th order cumulative distribution functions

3.2.1 Definitions and review

The t-th order cumulative operator Γ^t of a function h is defined by

$$\Gamma^t h(x) = \sum_{y=0}^{x} \Gamma^{t-1} h(y), \quad x \geq 0,\ t \geq 1, \tag{3.2.1}$$

with $\Gamma^0 h(x) = h(x)$ and $\Gamma \equiv \Gamma^1$. It is shown in Antzoulakos and Chadjiconstantinidis (2004, p. 165) that this operator is equivalent to

$$\Gamma^t h(x) = \sum_{y=0}^{x} C_{x-y}^{t-1} h(y), \quad x \geq 0,\ t \geq 0, \tag{3.2.2}$$

where $C_b^a = \binom{a+b}{a}$[1].

Dhaene et al. (1999) showed that if the recursive formula for a probability function

$$f(s) = \frac{1}{s} \sum_{x=1}^{s} \varphi(x) f(s-x), \quad s \geq 1, \tag{3.2.3}$$

with initial value $f(0)$ holds then the t-th order cumulative distribution function can be evaluated by

$$\Gamma^t f(s) = \frac{1}{s} \sum_{x=1}^{s} (\varphi(x) + t) \Gamma^t f(s-x), \quad s \geq 1,\ t \geq 0, \tag{3.2.4}$$

[1]This is not the common notation for the binomial coefficient but our definition is used here in order to simplify expressions

with initial value $\Gamma^t f(0) = f(0)$. In the actuarial literature, the function φ is known under the name of De Pril transform (see Sundt (1995)). The De Pril transform of a function f is determined by

$$\varphi(x) = \frac{1}{f(0)}\left[xf(x) - \sum_{y=1}^{x-1}\varphi(x-y)f(y)\right], \quad x \geq 1. \tag{3.2.5}$$

The most useful property of De Pril transforms is that the De Pril transform of a convolution of probability functions of independent random variables is the sum of the De Pril transforms of these probability functions. This property was proved by De Pril (1989).

Some additional properties of De Pril transforms that are discussed in Dhaene and De Pril (1994) may be useful. Let $P(u)$ and φ be the probability generating function and the De Pril transform of a random variable Y, respectively. They showed that under some convergence condition, we have

$$\ln P(u) = \ln f(0) + \sum_{x=1}^{\infty}\frac{\varphi(x)}{x}u^x, \tag{3.2.6}$$

where f is the probability function of Y and

$$P(u) = \sum_{y=0}^{\infty} f(y)u^y. \tag{3.2.7}$$

Therefore, by combining (3.2.6) and (3.2.7) and setting $u = 1$ we get

$$\sum_{x=1}^{\infty}\frac{\varphi(x)}{x} = -\ln f(0). \tag{3.2.8}$$

We obtain

$$\frac{P'(u)}{P(u)} = \frac{\sum_{y=1}^{\infty} yf(y)u^{y-1}}{\sum_{y=0}^{\infty} f(y)u^y} = \sum_{x=1}^{\infty}\varphi(x)u^{x-1}, \tag{3.2.9}$$

by taking the first derivative with respect to u in (3.2.6) and (3.2.7). It follows that

$$\mathrm{E}[Y] = \sum_{x=1}^{\infty}\varphi(x), \tag{3.2.10}$$

by setting $u = 1$ in (3.2.9). If we take higher order derivatives of (3.2.6) and set $u = 1$ for each order we obtain Theorem 4.1 of Sundt et al. (1998) that gives an expression to calculate the cumulants of Y from φ.

3.2.2 A more efficient recursive evaluation

We develop now a more efficient way for the evaluation of t-th order cumulative distribution functions. It is particularly useful when the De Pril transform converges to zero. Equation (3.2.4) can be rewritten as

$$\Gamma^t f(s) = \frac{1}{s}\left[\sum_{x=1}^{s}\varphi(x)\,\Gamma^t f(s-x) + \sum_{x=1}^{s} t\,\Gamma^t f(s-x)\right], \quad s \geq 1,\ t \geq 0. \quad (3.2.11)$$

From this last equation, an embedded recursion follows for $\Gamma^t f$ which is

$$\Gamma^t f(s) = \frac{1}{s}\left[t\,\Gamma^{t+1} f(s-1) + \sum_{x=1}^{s}\varphi(x)\,\Gamma^t f(s-x)\right], \quad s \geq 1,\ t \geq 0, \quad (3.2.12)$$

where

$$\Gamma^{t+1} f(s) = \Gamma^{t+1} f(s-1) + \Gamma^t f(s), \quad (3.2.13)$$

and with initial values $\Gamma^u f(0) = f(0),\ u = t, t+1$.

We can notice that in recursion (3.2.4) the terms $\varphi(x)+t$ for $t \neq 0$ and $x \geq 1$ are generally different from zero. Therefore, we have to do s multiplications in order to evaluate $\Gamma^t f(s)$, while time can be saved using (3.2.12) because some multiplications may be avoided if the De Pril transform is equal to zero at some points. In the case where $\varphi(y) = 0$ for $y > r$, the embedded recursion (3.2.12) becomes

$$\Gamma^t f(s) = \frac{1}{s}\left[t\,\Gamma^{t+1} f(s-1) + \sum_{x=1}^{\min(s,r)}\varphi(x)\,\Gamma^t f(s-x)\right], \quad s \geq 1,\ t \geq 0, \quad (3.2.14)$$

using (3.2.13) and with initial values $\Gamma^u f(0) = f(0),\ u = t, t+1$. This is the case for the De Pril transform of a compound Poisson distribution with parameters $(\lambda, g(x))$, $x \geq 1$, where

$$\varphi(y) = \begin{cases} \lambda y g(y), & y = 1, \ldots, \omega \\ 0, & y > \omega \end{cases}, \quad (3.2.15)$$

where $\omega = \sup\{y : g(y) > 0\}$ (see Sundt (1995, p. 25)).

When the De Pril transform satisfies

$$\lim_{x \to +\infty} \varphi(x) = 0, \quad (3.2.16)$$

we can truncate it by setting it to zero from a given point $r+1$. The gain of time is double since we need to evaluate the De Pril transform for a smaller number of points and since we can use (3.2.14) instead of (3.2.4). Compound negative binomial distributions and some cases of compound binomial distributions satisfy (3.2.16) (see Theorem 6.1 in Dhaene and Sundt (1998)).

3.2.3 Some applications

One could say that it is displeasing to need the $(t+1)$-th order cumulative distribution function to evaluate the t-th order one. However, its evaluation using (3.2.13) is very simple and the time that can be saved using (3.2.14) is substantial compared to the evaluation using (3.2.4). Moreover, it gives more information about the distribution of the random variable that may be useful as we will see in the following applications.

Let $I_d = (S-d)_+$ be the amount paid by the reinsurer for a stop-loss contract with deductible $d \geq 0$ given a random loss S. From the second order cumulative distribution function, the stop-loss premium is determined by

$$\mathrm{E}[I_d] \;=\; \Gamma^2 f(d-1) + \mathrm{E}[S] - d, \quad d \geq 0, \tag{3.2.17}$$

given that we know the expected value of the loss (see Dhaene et al. (1999)). The following theorem gives a new expression for the variance of the stop-loss reinsurer's payment.

Theorem 1 *Given a random loss S, the variance of the stop-loss reinsurer's payment with deductible d can be determined from the second and the third order cumulative distribution functions by*

$$\mathrm{Var}[I_d] \;=\; \mathrm{E}\big[(S-d)^2\big] - 2\,\Gamma^3 f(d-1) + \Gamma^2 f(d-1) - \left(\mathrm{E}[I_d]\right)^2, \quad d \geq 0. \tag{3.2.18}$$

Proof. By definition, we have

$$\mathrm{Var}[I_d] \;=\; \mathrm{E}\big[I_d^2\big] - \left(\mathrm{E}[I_d]\right)^2. \tag{3.2.19}$$

From Antzoulakos and Chadjiconstantinidis (2004, p. 181), we know that

$$\mathrm{E}\big[I_d^2\big] \;=\; 2\,\Lambda^3 f(d-1) + \Lambda^2 f(d-1), \quad d \geq 0, \tag{3.2.20}$$

where Λ^t is the t-th order tail operator and is defined by

$$\Lambda^t h(x) \;=\; \sum_{y=x+1}^{\infty} \Lambda^{t-1} h(y), \quad x \geq 0,\ t \geq 1, \tag{3.2.21}$$

with $\Lambda^0 h(x) = h(x)$. From their Lemma 3.1 (d), it follows that

$$\Lambda^2 f(d-1) \;=\; \Gamma^2 f(d-1) + \mathrm{E}[S] - d, \quad d \geq 0, \tag{3.2.22a}$$

$$\Lambda^3 f(d-1) \;=\; -\Gamma^3 f(d-1) + \frac{1}{2}\,\mathrm{E}\big[(S-d)^2\big] - \frac{1}{2}\,\mathrm{E}[S] + \frac{d}{2}, \quad d \geq 0. \tag{3.2.22b}$$

The substitution of (3.2.22) into (3.2.20) leads to

$$\mathrm{E}\big[I_d^2\big] \;=\; \mathrm{E}\big[(S-d)^2\big] - 2\,\Gamma^3 f(d-1) + \Gamma^2 f(d-1). \tag{3.2.23}$$

Equation (3.2.18) follows by substituting (3.2.23) into (3.2.19) and the proof of Theorem 1 is complete. ■

Another expression for the variance of the stop-loss reinsurer's payment which contains only cumulative distribution functions and central moments of S follows from the substitution of (3.2.17) into (3.2.18). We get

$$\mathrm{Var}[I_d] \;=\; \mathrm{Var}[S] - 2\,\Gamma^3 f(d-1) + \Gamma^2 f(d-1)\left(2d+1-2\mathrm{E}[S] - \Gamma^2 f(d-1)\right), \tag{3.2.24}$$

for $d \geq 0$.

An application of the embedded recursion (3.2.14) for $t = 1$ is the computation of the expected shortfall (ES) of a random variable S at a given probability level α, which is defined by

$$\begin{aligned}\mathrm{ES}_\alpha[S] &= \frac{\mathrm{E}[S; S > \mathrm{VaR}_\alpha[S]] + (\Pr[S \leq \mathrm{VaR}_\alpha[S]] - \alpha)\,\mathrm{VaR}_\alpha[S]}{1-\alpha} \\ &= \mathrm{VaR}_\alpha[S] + \frac{1}{1-\alpha}\,\mathrm{E}\big[(S - \mathrm{VaR}_\alpha[S])_+\big],\end{aligned} \tag{3.2.25}$$

where $\mathrm{VaR}_\alpha[S]$ is the Value-at-Risk at level α. The idea consists in evaluating the distribution function using the embedded recursion (3.2.14) with $t = 1$ until $\Gamma f(s) > \alpha$, so that $\mathrm{VaR}_\alpha[S] = s$. The expected shortfall is then determined by

$$\mathrm{ES}_\alpha[S] \;=\; \frac{\mathrm{E}[S] + \Gamma^2 f(s-1) - \alpha\, s}{1-\alpha}, \tag{3.2.26}$$

where $\Gamma^2 f(s-1)$ is immediately given by (3.2.13). The evaluation of the second order cumulative distribution function using (3.2.14) is useful if we want to calculate the stop-loss premium and the variance of the stop-loss reinsurer's payment for a given deductible using (3.2.17) and Theorem 1, respectively. One can find similar applications for higher orders.

3.3 Approximations of t-th order cumulative distribution functions

Using the embedded recursion (3.2.14), an approximation of the De Pril transform leads to an approximation of the t-th order cumulative distribution function. For such approximations, bounds for the absolute error of the distribution function and stop-loss premiums

have been developed in the actuarial literature (see e.g. Dhaene and De Pril (1994) and Dhaene and Sundt (1997, 1998)). In this section, we determine the exact error of the t-th order cumulative distribution function occurring in its evaluation using an approximation of the De Pril transform.

Let $\widetilde{\varphi}$ be an approximation of the De Pril transform of a given probability function f. Let $\widetilde{f}$ be the approximation of this probability function evaluated using (3.2.3) with $\widetilde{\varphi}$ and initial value $\widetilde{f}(0)$. We define the function ψ by

$$\psi(x) \;=\; \begin{cases} 0, & x = 0 \\ \frac{\widetilde{\varphi}(x)-\varphi(x)}{x}, & x \geq 1 \end{cases} . \tag{3.3.1}$$

It follows from (3.2.8) that

$$\sum_{x=1}^{\infty} \psi(x) \;=\; \ln f(0) + \sum_{x=1}^{\infty} \frac{\widetilde{\varphi}(x)}{x}. \tag{3.3.2}$$

Equation (3.2.9) also holds for approximations. Therefore, the expected value of the approximation is given by

$$\sum_{y=1}^{\infty} y\widetilde{f}(y) \;=\; \sum_{x=1}^{\infty} \widetilde{\varphi}(x) \cdot \sum_{y=0}^{\infty} \widetilde{f}(y). \tag{3.3.3}$$

The cumulants of the approximation can be determined similarly to Theorem 4.1 in Sundt et al. (1998).

The following theorem gives an expression for the exact error of the probability function resulting from an evaluation using an approximation of its De Pril transform.

Theorem 2 *Let $\zeta(s) = \widetilde{f}(s) - f(s)$, $s \geq 0$, be the error of the probability function at s when it is evaluated using (3.2.3) with the approximative De Pril transform $\widetilde{\varphi}$ and initial value $\widetilde{f}(0) = f(0)$. If ψ is defined by (3.3.1) we have*

$$\zeta(s) \;=\; \sum_{y=1}^{s} \sum_{n=1}^{y} \frac{1}{n!}\, \psi^{*n}(y) f(s-y), \quad s \geq 1, \tag{3.3.4}$$

where ψ^{*n} is the n-fold convolution of ψ.

Proof. From Dhaene and De Pril (1994, p. 185), we know that if $\widetilde{f}(0) = f(0)$ we have

$$\zeta(s) \;=\; \sum_{y=1}^{s} a(y) f(s-y), \quad s \geq 1, \tag{3.3.5}$$

where the function a is the solution of the recurrence equation

$$xa(x) \;=\; \sum_{z=1}^{x} z\, \psi(z) a(x-z), \quad x \geq 1, \tag{3.3.6}$$

with initial value $a(0) = 1$. Therefore, to prove (3.3.4), it is sufficient to prove by induction

on y that

$$a(y) = \sum_{n=1}^{y} \frac{1}{n!}\, \psi^{*n}(y), \quad y \geq 1, \tag{3.3.7}$$

is the solution of (3.3.6) with initial value $a(0) = 1$. For $y = 1$, it holds straightforwardly that (3.3.7) is equivalent to (3.3.6) with $x = 1$. Now let us assume that (3.3.7) holds for $y = 1, \ldots, w$, we obtain from (3.3.6) that

$$\begin{aligned}(w+1)a(w+1) &= \sum_{z=1}^{w+1} z\,\psi(z)a(w+1-z) \\ &= (w+1)\,\psi(w+1) + \sum_{n=1}^{w} \sum_{z=1}^{w+1-n} \frac{z}{n!}\,\psi^{*n}(w+1-z)\,\psi(z). \end{aligned} \tag{3.3.8}$$

We know that the n-fold convolution $\psi^{*n}(s)$ is the coefficient of x^s of the polynomial

$$\left(\sum_{y=1}^{\infty} \psi(y)x^y\right)^n, \tag{3.3.9}$$

(see e.g. Knuth (1992)). Remark that $\psi^{*n}(s) = 0$ for $s < n$ and that the sum in (3.3.9) can be taken only over $y = 1, 2, \ldots, s+1-n$ in order to determine the coefficient of x^s, $s \geq n$. Therefore, we have

$$\psi^{*n}(s) = \sum_{\substack{\sum u_i = n \\ \sum i u_i = s}} \binom{n}{u_1, \ldots, u_{s+1-n}} \prod_{i=1}^{s+1-n} \psi(i)^{u_i}, \quad s \geq 1,\ n \geq 1, \tag{3.3.10}$$

where $u_i,\ i = 1, \ldots, s+1-n$, are nonnegative integers. By substituting (3.3.10) into the inner sum of (3.3.8) and letting $k = w+1-n$, it follows that

$$\begin{aligned}\sum_{z=1}^{k} \frac{z}{n!}\,\psi^{*n}(w+1-z)\psi(z) &= \sum_{z=1}^{k} \sum_{\substack{\sum u_i = n \\ \sum i u_i = w+1-z}} \frac{z}{n!}\binom{n}{u_1, \ldots, u_{k+1-z}}\psi(z) \prod_{i=1}^{k+1-z} \psi(i)^{u_i} \\ &= \sum_{z=1}^{k} \sum_{\substack{\sum u_i = n \\ \sum i u_i = w+1-z}} \frac{z}{n!}\binom{n}{u_1, \ldots, u_k}\psi(z) \prod_{i=1}^{k} \psi(i)^{u_i} \\ &= \sum_{z=1}^{k} \sum_{\substack{\sum u_i = n+1 \\ \sum i u_i = w+1}} \frac{z u_z}{(n+1)!}\binom{n+1}{u_1, \ldots, u_k} \prod_{i=1}^{k} \psi(i)^{u_i} \\ &= \sum_{\substack{\sum u_i = n+1 \\ \sum i u_i = w+1}} \frac{\sum_{z=1}^{k} z u_z}{(n+1)!}\binom{n+1}{u_1, \ldots, u_k} \prod_{i=1}^{k} \psi(i)^{u_i} \\ &= \frac{w+1}{(n+1)!}\,\psi^{*(n+1)}(w+1). \end{aligned} \tag{3.3.11}$$

The substitution of (3.3.11) into (3.3.8) and the division of the latter by $w+1$ on both sides lead to

$$a(w+1) \;=\; \psi(w+1) + \sum_{n=1}^{w} \frac{1}{(n+1)!}\,\psi^{*(n+1)}(w+1) = \sum_{n=1}^{w+1} \frac{1}{n!}\,\psi^{*n}(w+1). \quad (3.3.12)$$

This proves that (3.3.7) holds for $y = w+1$. Thus, (3.3.7) is the solution of (3.3.6) with initial value $a(0) = 1$ and the proof of Theorem 2 is complete. ■

Corollary 3 *Let $\widetilde{\Gamma^t f}$, $t \geq 0$, be the approximation of the t-th order cumulative distribution function evaluated using (3.2.4) or (3.2.12) with $\widetilde{\varphi}$ and initial values $\widetilde{\Gamma^u f}(0) = \widetilde{f}(0)$, $u = t, t+1$. The error of $\widetilde{\Gamma^t f}$ is given by*

$$\widetilde{\Gamma^t f}(s) - \Gamma^t f(s) \;=\; \Gamma^t \zeta(s), \quad s \geq 0,\ t \geq 0. \quad (3.3.13)$$

Proof. For $t = 0$, (3.3.13) is Theorem 2. Let us now assume that (3.3.13) holds for $t = u$, we obtain

$$\begin{aligned} \widetilde{\Gamma^{u+1} f}(s) - \Gamma^{u+1} f(s) \;&=\; \sum_{x=0}^{s} \left(\widetilde{\Gamma^u f}(x) - \Gamma^u f(x) \right) \\ &=\; \sum_{x=0}^{s} \Gamma^u \zeta(x) = \Gamma^{u+1} \zeta(s), \quad s \geq 0. \end{aligned} \quad (3.3.14)$$

Therefore, (3.3.13) holds for $t = u+1$ and the proof of Corollary 3 is complete by induction on t. ■

In particular, when $t = 1$ Corollary 3 is

$$\widetilde{F}(s) - F(s) \;=\; \sum_{y=1}^{s} \sum_{n=1}^{y} \frac{1}{n!}\,\psi^{*n}(y) F(s-y), \quad s \geq 0, \quad (3.3.15)$$

where F and $\widetilde{F}$ denote the distribution function and its approximation, respectively. It follows in the limit $s \to \infty$ that

$$\begin{aligned} \lim_{s\to\infty} \left(\widetilde{F}(s) - F(s) \right) \;&=\; \sum_{y=1}^{\infty} \sum_{n=1}^{y} \frac{1}{n!}\,\psi^{*n}(y) = \sum_{n=1}^{\infty} \frac{1}{n!} \left(\sum_{y=1}^{\infty} \psi(y) \right)^n \\ &=\; \exp\left\{ \sum_{y=1}^{\infty} \psi(y) \right\} - 1. \end{aligned} \quad (3.3.16)$$

In opposition to the results in Dhaene and De Pril (1994), we also consider the sign of the error in Theorem 2 and Corollary 3, which leads to an expression for the exact error

instead of giving error bounds. If we take the absolute value of addends on both sides of (3.3.16) we arrive to an error bound that they derived, that is

$$\lim_{s\to\infty}\left|\widetilde{F}(s)-F(s)\right| \;\le\; \sum_{x=0}^{\infty}\left|\widetilde{f}(x)-f(x)\right| \;\le\; \exp\left\{\sum_{y=1}^{\infty}|\psi(y)|\right\}-1. \tag{3.3.17}$$

The latter inequality of (3.3.17) will be proved later.

Remark 1: We get from Theorem 2 and Corollary 3 that

$$\widetilde{\Gamma^t f}(s)-\Gamma^t f(s) \;=\; \sum_{y=1}^{s}\sum_{n=1}^{y}\frac{1}{n!}\,\psi^{*n}(y)\Gamma^t f(s-y), \quad s\ge 0,\ t\ge 0. \tag{3.3.18}$$

However, this error depends on the exact values of the t-th order cumulative distribution function that should not be known if an approximation is made.

Remark 2: From the substitution of (3.3.2) into (3.3.16), it follows that

$$\lim_{s\to\infty}\left(\widetilde{F}(s)-F(s)\right) \;=\; f(0)\cdot\exp\left\{\sum_{x=1}^{\infty}\frac{\widetilde{\varphi}(x)}{x}\right\}-1, \tag{3.3.19}$$

which leads to

$$\lim_{s\to\infty}\widetilde{F}(s) \;=\; f(0)\cdot\exp\left\{\sum_{x=1}^{\infty}\frac{\widetilde{\varphi}(x)}{x}\right\}. \tag{3.3.20}$$

Remark 3: If the assumption $\widetilde{f}(0)=f(0)$ is released, the initial value of recursion (3.3.6) is $a(0)=\frac{\widetilde{f}(0)}{f(0)}$ (see Dhaene and De Pril (1994, p. 185)). Equations (3.3.4), (3.3.15), (3.3.16), (3.3.17), (3.3.19) and (3.3.20) become respectively

$$\zeta(s) \;=\; (a(0)-1)f(s)+a(0)\sum_{y=1}^{s}\sum_{n=1}^{y}\frac{1}{n!}\,\psi^{*n}(y)f(s-y), \quad s\ge 0; \tag{3.3.21}$$

$$\widetilde{F}(s)-F(s) \;=\; (a(0)-1)F(s)+a(0)\sum_{y=1}^{s}\sum_{n=1}^{y}\frac{1}{n!}\,\psi^{*n}(y)F(s-y), \quad s\ge 0; \tag{3.3.22}$$

$$\lim_{s\to\infty}\left(\widetilde{F}(s)-F(s)\right) \;=\; a(0)\cdot\exp\left\{\sum_{y=1}^{\infty}\psi(y)\right\}-1; \tag{3.3.23}$$

$$\lim_{s\to\infty}\left|\widetilde{F}(s)-F(s)\right| \le \sum_{k=0}^{\infty}\left|\widetilde{f}(k)-f(k)\right| \le \exp\left\{\left|\ln\left(\frac{\widetilde{f}(0)}{f(0)}\right)\right|+\sum_{y=1}^{\infty}|\psi(y)|\right\}-1; \tag{3.3.24}$$

$$\lim_{s\to\infty}\left(\widetilde{F}(s)-F(s)\right) \;=\; \widetilde{f}(0)\cdot\exp\left\{\sum_{x=1}^{\infty}\frac{\widetilde{\varphi}(x)}{x}\right\}-1; \tag{3.3.25}$$

$$\lim_{s\to\infty}\widetilde{F}(s) \;=\; \widetilde{f}(0)\cdot\exp\left\{\sum_{x=1}^{\infty}\frac{\widetilde{\varphi}(x)}{x}\right\}. \tag{3.3.26}$$

However, if we consider the t-th order cumulative distribution function, the error of $\widetilde{f}(0)$ will have a large effect on this function because it is accumulated at each point for t orders. Therefore, it is always better to start recursions with the exact value, which is easy in practice.

Remark 4: The substitution of (3.3.26) into (3.3.3) leads to

$$\sum_{y=1}^{\infty}y\widetilde{f}(y) \;=\; \widetilde{f}(0)\cdot\exp\left\{\sum_{x=1}^{\infty}\frac{\widetilde{\varphi}(x)}{x}\right\}\cdot\sum_{x=1}^{\infty}\widetilde{\varphi}(x), \tag{3.3.27}$$

which gives an expression for the expected value of the approximation in the general case.

3.4 Error bounds for t-th order cumulative distribution functions

In the previous section, the exact error of $\widetilde{\Gamma^t f}$ is expressed as a function of ψ but depends on exact values. Hence, we use it to built a bound for such an error. This error bound is given in the following theorem. From now on, we will generally consider cumulative distribution functions with an order greater than or equal to one. Therefore, in the sequel the expressions in relation to $\Gamma^t f$ will hold for $t \geq 1$ unless stated.

Theorem 4 *Let $\epsilon(s)=\sum_{k=1}^{s}|\psi(k)|$, $s\geq 0$. If we assume that $\epsilon(s-1)\neq 0$, a bound for the absolute error of $\widetilde{\Gamma^t f}$ evaluated using (3.2.4) or (3.2.12) according to the approximation (3.3.1) is given by*

$$\left|\widetilde{\Gamma^t f}(s)-\Gamma^t f(s)\right| \;\leq\; \eta_t(s), \quad s\geq 0, \tag{3.4.1}$$

where $\eta_t(s)$ is defined by

$$\eta_t(s)=\begin{cases}0, & s=0\\ \epsilon(1)f(0), & s=1\\ \frac{e^{\epsilon(s-1)}-1}{\epsilon(s-1)}\,\Gamma^{t-1}\epsilon(s), & s>1\end{cases}. \tag{3.4.2}$$

Proof. For $s = 1$ the prove of (3.4.1) is straightforward from Theorem 2 and Corollary 3. We even have the equality

$$\widetilde{\Gamma^t f}(1) - \Gamma^t f(1) \;=\; \psi(1)f(0), \quad t \geq 0. \tag{3.4.3}$$

Let us now prove (3.4.1) for $t = 1$ and $s > 1$. First, we rewrite (3.3.15) as

$$\widetilde{F}(s) - F(s) \;=\; \sum_{n=1}^{s}\sum_{y=n}^{s} \frac{1}{n!}\,\psi^{*n}(y)F(s-y), \quad s \geq 0. \tag{3.4.4}$$

It follows that

$$\begin{aligned}\left|\widetilde{F}(s) - F(s)\right| \;&\leq\; \sum_{n=1}^{s}\sum_{y=n}^{s} \frac{1}{n!}\,|\psi^{*n}(y)| \leq \sum_{n=1}^{s} \frac{1}{n!}\,\epsilon(s+1-n)^n \leq \epsilon(s) + \sum_{n=2}^{\infty} \frac{1}{n!}\,\epsilon(s-1)^n \\ &=\; \epsilon(s) + e^{\epsilon(s-1)} - 1 - \epsilon(s-1), \quad s \geq 1. \end{aligned} \tag{3.4.5}$$

We assume now that $\epsilon(s-1) \neq 0$. Since $\epsilon(s) = \epsilon(s-1) + |\psi(s)|$, we have $\epsilon(s) \neq 0$ and we obtain from (3.4.5) that

$$\begin{aligned}\left|\widetilde{F}(s) - F(s)\right| \;&\leq\; \epsilon(s)\left[1 + \frac{e^{\epsilon(s-1)} - 1 - \epsilon(s-1)}{\epsilon(s)}\right] \leq \epsilon(s)\left[1 + \frac{e^{\epsilon(s-1)} - 1 - \epsilon(s-1)}{\epsilon(s-1)}\right] \\ &=\; \frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s) = \eta_1(s), \quad s > 1. \end{aligned} \tag{3.4.6}$$

This proves that Theorem 4 holds for $t = 1$ and $s > 1$. In the general case, we have

$$\left|\widetilde{\Gamma^{t-1}F}(s) - \Gamma^{t-1}F(s)\right| \leq \Gamma^{t-1}\left|\widetilde{F}(s) - F(s)\right| \leq \Gamma^{t-1}\left[\frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s)\right], \quad s > 1. \tag{3.4.7}$$

Since $\epsilon(s)$ is nondecreasing with respect to s and $\frac{e^x-1}{x}$ is a positive increasing function for any real number x, we arrive to

$$\Gamma^{t-1}\left[\frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s)\right] \;\leq\; \frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\Gamma^{t-1}\epsilon(s) = \eta_t(s), \quad s > 1. \tag{3.4.8}$$

The proof of Theorem 4 is complete by substituting (3.4.8) into (3.4.7). ■

Remark 5: The assumption $\epsilon(s-1) \neq 0$ is not restricting, otherwise there would not be any error until $s-1$ and we would have $\widetilde{\Gamma^t f}(s) - \Gamma^t f(s) = \psi(s)f(0)$. This equality follows immediately from Theorem 2 and Corollary 3.

Remark 6: The term $\frac{e^{\epsilon(s-1)}-1}{\epsilon(s-1)}$ tends to one if $\epsilon(s-1)$ tends to zero. It is generally close to one in reasonable applications, which means that $\eta_t(s)$ grows similarly to $\Gamma^{t-1}\epsilon(s)$.

Corollary 5 *In the limit $s \to \infty$, the error bound defined in Theorem 4 is also a bound for the distance between the approximative and the exact $(t-1)$-th order cumulative distribution functions. We have*

$$\sum_{x=0}^{\infty} \left| \widetilde{\Gamma^{t-1} f}(x) - \Gamma^{t-1} f(x) \right| \leq \lim_{s \to \infty} \eta_t(s). \tag{3.4.9}$$

Proof. For $t = 1$, (3.4.9) is proved using successively Theorem 2, (3.4.5) and (3.4.6) in

$$\begin{aligned} \sum_{x=0}^{\infty} \left| \widetilde{f}(x) - f(x) \right| &\leq \sum_{x=0}^{\infty} \sum_{y=1}^{x} \sum_{n=1}^{y} \frac{1}{n!} \left| \psi^{*n}(y) \right| f(x-y) \\ &\leq \sum_{y=1}^{x} \sum_{n=1}^{y} \frac{1}{n!} \left| \psi^{*n}(y) \right| \leq \lim_{s \to \infty} \eta_1(s). \end{aligned} \tag{3.4.10}$$

We assume now that (3.4.9) holds for $t = u$. It follows for the same reasons as for (3.4.8) that

$$\begin{aligned} \sum_{x=0}^{\infty} \left| \widetilde{\Gamma^{u} f}(x) - \Gamma^{u} f(x) \right| &\leq \sum_{x=0}^{\infty} \sum_{y=0}^{x} \left| \widetilde{\Gamma^{u-1} f}(y) - \Gamma^{u-1} f(y) \right| \\ &\leq \sum_{x=0}^{\infty} \eta_u(x) \leq \lim_{s \to \infty} \eta_{u+1}(s). \end{aligned} \tag{3.4.11}$$

Therefore, (3.4.9) holds for $t = u+1$ and the proof of Corollary 5 is complete by induction on t. ■

We consider now the case where the De Pril transform is approximated according to

$$\widetilde{\varphi}(y) = \begin{cases} \varphi(y), & y = 1, \ldots, r \\ 0, & y > r \end{cases}, \ r \geq 1, \tag{3.4.12}$$

such that the evaluation using the embedded recursion (3.2.14) is more efficient than the evaluation using (3.2.4). Let

$$\epsilon^{(r)}(s) = \sum_{k=r+1}^{s} \frac{|\varphi(k)|}{k}, \quad r \geq 0, \ s > r, \tag{3.4.13}$$

denote the equivalent to $\epsilon(s)$ for such an approximation by truncating the De Pril transform. Notice that (3.4.13) is nonincreasing with respect to r. Thus, from the error bound defined in Theorem 4, a truncation point can be determined for any required accuracy for the approximation of the t-th order cumulative distribution function. From (3.2.2) and (3.4.13), it follows that

$$\Gamma^{t} \epsilon^{(r)}(s) = \Gamma^{t} \epsilon^{(r-1)}(s) - C_{s-r}^{t-1} \frac{|\varphi(r)|}{r}, \quad r \geq 1, \ s > r, \tag{3.4.14}$$

from which we can obtain a better guaranteed accuracy by setting the De Pril transform to zero from $r+1$ instead of r. We can also use it in the other direction if we want to have a faster evaluation of $\widetilde{\Gamma^t f}$ by reducing its guaranteed accuracy. Observe that the guaranteed accuracy of $\widetilde{\Gamma^t f}(s)$ is given by

$$\eta_t^{(r)}(s) \;=\; \frac{e^{\epsilon^{(r)}(s-1)}-1}{\epsilon^{(r)}(s-1)}\,\Gamma^{t-1}\epsilon^{(r)}(s), \quad r\geq 1,\ s>r+1, \tag{3.4.15}$$

which behaves similarly to $\Gamma^{t-1}\epsilon^{(r)}(s)$.

3.5 Error bounds for stop-loss contracts

In this section, we combine the results of Sections 3.2.3 and 3.4. We derive intervals for the approximation of the stop-loss premium and for the approximation of the variance of the stop-loss reinsurer's payment. Unless specified, the expressions in this section hold for any deductible $d\geq 0$.

Let $\widetilde{\mathrm{E}[I_d]}$ be the approximation of the stop-loss premium resulting from the approximative De Pril transform $\widetilde{\varphi}$. It is determined by

$$\widetilde{\mathrm{E}[I_d]} \;=\; \widetilde{\Gamma^2 f}(d-1)+\mathrm{E}[S]-d. \tag{3.5.1}$$

We also introduce the approximations associated to (3.2.23) and (3.2.18), which are obtained by

$$\widetilde{\mathrm{E}[I_d^2]} \;=\; \mathrm{E}\big[(S-d)^2\big]-2\,\widetilde{\Gamma^3 f}(d-1)+\widetilde{\Gamma^2 f}(d-1) \tag{3.5.2}$$

and

$$\widetilde{\mathrm{Var}[I_d]} \;=\; \widetilde{\mathrm{E}[I_d^2]}-\Big(\widetilde{\mathrm{E}[I_d]}\Big)^2, \tag{3.5.3}$$

respectively.

From (3.2.17), (3.5.1) and Theorem 4, an error bound for the stop-loss premium with deductible d is given by

$$\Big|\widetilde{\mathrm{E}[I_d]}-\mathrm{E}[I_d]\Big| \;\leq\; \eta_2(d-1). \tag{3.5.4}$$

From (3.2.23), (3.5.2) and Theorem 4, it follows that

$$\Big|\widetilde{\mathrm{E}[I_d^2]}-\mathrm{E}\big[I_d^2\big]\Big| \;\leq\; 2\,\eta_3(d-1)+\eta_2(d-1). \tag{3.5.5}$$

Moreover, if $\epsilon(d-2) \neq 0$ we have

$$\left|\widetilde{\mathrm{E}[I_d^2]} - \mathrm{E}\big[I_d^2\big]\right| \;\leq\; \frac{e^{\epsilon(d-2)}-1}{\epsilon(d-2)}\left[2\,\Gamma\epsilon(d-1)+\epsilon(d-1)\right], \quad d>2. \tag{3.5.6}$$

Since $\mathrm{E}[I_d] \geq 0$, we obtain from (3.5.4) that

$$\left[\left(\widetilde{\mathrm{E}[I_d]} - \eta_2(d-1)\right)_+\right]^2 \;\leq\; \mathrm{E}[I_d]^2 \;\leq\; \left(\widetilde{\mathrm{E}[I_d]} + \eta_2(d-1)\right)^2. \tag{3.5.7}$$

An interval for the variance of the stop-loss reinsurer's payment follows from (3.5.5) and (3.5.7). It is given by

$$\mathrm{Var}[I_d] \geq \left(\widetilde{\mathrm{Var}[I_d]} - 2\,\eta_3(d-1) - \eta_2(d-1) - 2\,\widetilde{\mathrm{E}[I_d]}\,\eta_2(d-1) - [\eta_2(d-1)]^2\right)_+ \tag{3.5.8a}$$

$$\mathrm{Var}[I_d] \leq \widetilde{\mathrm{E}[I_d^2]} + 2\,\eta_3(d-1) + \eta_2(d-1) - \left[\left(\widetilde{\mathrm{E}[I_d]} - \eta_2(d-1)\right)_+\right]^2. \tag{3.5.8b}$$

In some cases, we are able to determine if the approximation made in the De Pril transform will lead to an underestimation or an overestimation of $\Gamma^t f$. In such cases, the intervals defined in (3.5.4), (3.5.5) and (3.5.8) can be reduced.

On the one hand, if we know that we underestimate $\Gamma^t f$ for at least $t \geq 2$, we get from Theorem 4 that

$$\widetilde{\Gamma^t f}(s) \;\leq\; \Gamma^t f(s) \;\leq\; \widetilde{\Gamma^t f}(s) + \eta_t(s), \quad s \geq 0,\; t \geq 2. \tag{3.5.9}$$

It follows from (3.2.17) and (3.5.1) that

$$\left(\widetilde{\mathrm{E}[I_d]}\right)_+ \;\leq\; \mathrm{E}[I_d] \;\leq\; \widetilde{\mathrm{E}[I_d]} + \eta_2(d-1), \tag{3.5.10}$$

and squaring it leads to

$$\left[\left(\widetilde{\mathrm{E}[I_d]}\right)_+\right]^2 \;\leq\; \mathrm{E}[I_d]^2 \;\leq\; \left(\widetilde{\mathrm{E}[I_d]} + \eta_2(d-1)\right)^2. \tag{3.5.11}$$

Let us subtract (3.5.2) from (3.2.23), we get

$$\mathrm{E}\big[I_d^2\big] - \widetilde{\mathrm{E}[I_d^2]} \;=\; \Gamma^2 f(d-1) - \widetilde{\Gamma^2 f}(d-1) - \left(2\,\Gamma^3 f(d-1) - 2\,\widetilde{\Gamma^3 f}(d-1)\right). \tag{3.5.12}$$

Since $\widetilde{\Gamma^2 f}(d-1)$ and $\widetilde{\Gamma^3 f}(d-1)$ are underestimation of $\Gamma^2 f(d-1)$ and $\Gamma^3 f(d-1)$, respectively, both errors partly compensate in (3.5.12). Moreover, from Corollary 3 and (3.5.9) we know that

$$\Gamma^{t+1} f(s) - \widetilde{\Gamma^{t+1} f}(s) \;\geq\; \Gamma^t f(s) - \widetilde{\Gamma^t f}(s), \quad s \geq 0,\; t \geq 2, \tag{3.5.13}$$

which leads to

$$-2\eta_3(d-1) \;\leq\; \Gamma^2 f(d-1) - \widetilde{\Gamma^2 f}(d-1) - \Big(2\,\Gamma^3 f(d-1) - 2\,\widetilde{\Gamma^3 f}(d-1)\Big) \;\leq\; 0. \quad (3.5.14)$$

Therefore, we obtain

$$\Big(\widetilde{\mathrm{E}[I_d^2]} - 2\eta_3(d-1)\Big)_+ \;\leq\; \mathrm{E}\big[I_d^2\big] \;\leq\; \widetilde{\mathrm{E}[I_d^2]}, \quad (3.5.15)$$

by combining (3.5.12) and (3.5.14). An interval for the variance of the stop-loss reinsurer's payment follows from (3.5.11) and (3.5.15). It is given by

$$\Big(\widetilde{\mathrm{Var}[I_d]} - 2\,\eta_3(d-1) - 2\,\widetilde{\mathrm{E}[I_d]}\,\eta_2(d-1) - [\eta_2(d-1)]^2\Big)_+ \;\leq\; \mathrm{Var}[I_d] \;\leq\; \widetilde{\mathrm{Var}[I_d]}. \quad (3.5.16)$$

On the other hand, if we know that we overestimate $\Gamma^t f$ for at least $t \geq 2$, the intervals (3.5.9), (3.5.10), (3.5.11), (3.5.15) and (3.5.16) become respectively:

$$\Big(\widetilde{\Gamma^t f}(s) - \eta_t(s)\Big)_+ \;\leq\; \Gamma^t f(s) \;\leq\; \widetilde{\Gamma^t f}(s), \quad s \geq 0,\ t \geq 2; \quad (3.5.17)$$

$$\Big(\widetilde{\mathrm{E}[I_d]} - \eta_2(d-1)\Big)_+ \;\leq\; \mathrm{E}[I_d] \;\leq\; \widetilde{\mathrm{E}[I_d]}; \quad (3.5.18)$$

$$\left[\Big(\widetilde{\mathrm{E}[I_d]} - \eta_2(d-1)\Big)_+\right]^2 \;\leq\; \mathrm{E}[I_d]^2 \;\leq\; \left[\widetilde{\mathrm{E}[I_d]}\right]^2; \quad (3.5.19)$$

$$\Big(\widetilde{\mathrm{E}[I_d^2]}\Big)_+ \;\leq\; \mathrm{E}\big[I_d^2\big] \;\leq\; \widetilde{\mathrm{E}[I_d^2]} + 2\eta_3(d-1); \quad (3.5.20)$$

$$\Big(\widetilde{\mathrm{Var}[I_d]}\Big)_+ \;\leq\; \mathrm{Var}[I_d] \;\leq\; \widetilde{\mathrm{E}[I_d^2]} + 2\,\eta_3(d-1) - \left[\Big(\widetilde{\mathrm{E}[I_d]} - \eta_2(d-1)\Big)_+\right]^2. \quad (3.5.21)$$

Observe that we always take the positive part of the lower bounds because the exact values are positive and these lower bounds may be negative.

3.6 Error bounds analysis

Dhaene and De Pril (1994) developed bounds for the absolute error of the distribution function and stop-loss premiums (see also Dhaene and Sundt (1997, 1998) and Sundt and Vernic (2009, Chapter 10)). Their results are based on the computation of these two quantities from the probability function evaluated according to (3.2.3).

The error bounds defined in Corollaries 2 and 4 of Dhaene and De Pril (1994) for the distribution function and stop-loss premiums, respectively, depend on the evaluations. This means that first we have to approximate the quantities in order to be able to determine the corresponding error bound. Hence, we cannot compare them in terms of accuracy to the error bounds defined in Theorem 4 and (3.5.4) that can be calculated before the evaluation and hold for any De Pril transform approximation. They also developed error bounds for approximations of stop-loss premiums that depend on the tail of the probability function. However, they involve the evaluation of the whole probability function, which is numerically impossible in some cases and is generally more demanding in terms of computing time in comparison to a recursive evaluation of $\Gamma^2 f$ using the embedded recursion (3.2.14).

The only comparison that we can make is between the error bound defined in Corollary 1 of Dhaene and De Pril (1994) and the one defined in Theorem 4 with $t = 1$. We can show that the latter is more accurate than the other one, that is

$$\frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s) \;\leq\; \lim_{s\to\infty} e^{\epsilon(s)} - 1, \quad s > 1. \tag{3.6.1}$$

Since $\epsilon(s)$ is nondecreasing with respect to s, (3.6.1) follows by taking the limit $s \to \infty$ on the right-hand side of

$$\frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s) \;\leq\; e^{\epsilon(s)} - 1, \quad s > 1. \tag{3.6.2}$$

This inequality holds because

$$\begin{aligned} e^{\epsilon(s)} - 1 - \frac{e^{\epsilon(s-1)} - 1}{\epsilon(s-1)}\,\epsilon(s) &= \sum_{k=1}^{\infty} \frac{1}{k!}\,\epsilon(s)^k - \frac{\epsilon(s)}{\epsilon(s-1)} \sum_{k=1}^{\infty} \frac{1}{k!}\,\epsilon(s-1)^k \\ &= \epsilon(s) \sum_{k=1}^{\infty} \frac{1}{k!} \left(\epsilon(s)^{k-1} - \epsilon(s-1)^{k-1}\right) \geq 0, \quad s > 1. \end{aligned} \tag{3.6.3}$$

Notice that we obtain equality in (3.6.1) if we take the limit $s \to \infty$ on its left-hand side.

3.7 Applications

3.7.1 Error bounds for compound Poisson distributions

We consider a compound Poisson distribution with parameters $(\lambda, g(x))$, $x = 1, 2, \ldots, \omega$. We let G be the distribution function related to the probability function g. The De Pril

transform of such a compound Poisson distribution is given by (3.2.15) where ω may be infinity. Therefore, the approximation by truncating g such that

$$\widetilde{\varphi}(y) = \begin{cases} \lambda y g(y), & y = 1, \ldots, r \\ 0, & y > r \end{cases}, \quad r \geq 1, \tag{3.7.1}$$

leads to

$$\psi(y) = \begin{cases} 0, & y = 1, \ldots, r \\ -\lambda g(y), & y > r \end{cases}, \quad r \geq 1. \tag{3.7.2}$$

Since we eliminate the probabilities in the tail of g, the truncation according to (3.7.1) gives an underestimation of $\Gamma^t f$ for $t \geq 0$. Thus, the intervals (3.5.9), (3.5.10), (3.5.11), (3.5.15) and (3.5.16) hold for this approximation. In this case, we can also use (3.4.14) to adapt the guaranteed accuracy if necessary.

For such an approximation we can show that

$$\Gamma^{t-1}\epsilon(s) = \lambda \sum_{k=r+1}^{s} C_{s-k}^{t-1}\, g(k) = \lambda \left[\Gamma^t g(s) - \sum_{k=1}^{t} C_{s-r-1}^{t-k} \Gamma^k g(r) \right], \quad s > r. \tag{3.7.3}$$

The first equality of (3.7.3) follows immediately from (3.2.2) and (3.7.2). In order to prove the second equality of (3.7.3), it is sufficient to prove that

$$\sum_{k=1}^{r} C_{s-k}^{t-1}\, g(k) = \sum_{k=1}^{t} C_{s-r-1}^{t-k} \Gamma^k g(r), \quad s > r, \tag{3.7.4}$$

since we have

$$\Gamma^t g(s) = \sum_{k=1}^{s} C_{s-k}^{t-1}\, g(k), \quad s > 1, \tag{3.7.5}$$

from (3.2.2). The substitution of (3.2.2) into the right-hand side of (3.7.4) leads to

$$\begin{aligned} \sum_{k=1}^{t} C_{s-r-1}^{t-k} \Gamma^k g(r) &= \sum_{k=1}^{t} C_{s-r-1}^{t-k} \sum_{y=1}^{r} C_{r-y}^{k-1}\, g(y) = \sum_{y=1}^{r} g(y) \sum_{k=1}^{t} C_{s-r-1}^{t-k} C_{r-y}^{k-1} \\ &= \sum_{y=1}^{r} C_{s-y}^{t-1}\, g(y), \quad s > r, \end{aligned} \tag{3.7.6}$$

which proves (3.7.3).

Remark that we can obtain an expression for $\Gamma^{t-1}\epsilon(s)$, which only depends on the t-th order cumulative distribution function of g. Let ∇_x^n be the n-th order backward difference operator with respect to x, which is defined by

$$\nabla_x^n h(x) = \nabla_x^{n-1} h(x) - \nabla_x^{n-1} h(x-1), \quad x \geq 1,\ n \geq 1, \tag{3.7.7}$$

with $\nabla_x^0 h(x) = h(x)$. Since $\nabla_x^1 \Gamma^t h(x) = \Gamma^{t-1} h(x)$, we have

$$\Gamma^k g(r) = \nabla_r^{t-k} \Gamma^t g(r) = \sum_{j=0}^{t-k} (-1)^j C_j^{t-k-j} \Gamma^t g(r-j), \quad r \geq 1. \tag{3.7.8}$$

By substituting (3.7.8) into (3.7.3), we arrive to

$$\begin{aligned}
\Gamma^{t-1}\epsilon(s) &= \lambda \left[\Gamma^t g(s) - \sum_{k=1}^{t} \sum_{j=0}^{t-k} (-1)^j C_{s-r-1}^{t-k} C_j^{t-k-j} \, \Gamma^t g(r-j) \right] \\
&= \lambda \left[\Gamma^t g(s) - \sum_{j=0}^{t-1} (-1)^j \, \Gamma^t g(r-j) \sum_{k=1}^{t-j} C_{s-r-1}^{t-k} C_j^{t-k-j} \right] \\
&= \lambda \left[\Gamma^t g(s) - C_{s-r}^{t-1} \sum_{j=0}^{t-1} (-1)^j \frac{s-r}{s-r+j} \, C_j^{t-1-j} \, \Gamma^t g(r-j) \right], \quad s > r. \qquad (3.7.9)
\end{aligned}$$

For $t = 1$, (3.7.3) or (3.7.9) is

$$\epsilon(s) = \lambda \sum_{y=r+1}^{s} g(y) = \lambda \left(G(s) - G(r) \right), \quad s > r, \tag{3.7.10}$$

which leads to

$$\eta_1(s) = \frac{e^{\lambda(G(s-1)-G(r))} - 1}{G(s-1) - G(r)} \left(G(s) - G(r) \right), \quad s > r+1. \tag{3.7.11}$$

In particular, we obtain

$$\lim_{s \to \infty} \left(\widetilde{F}(s) - F(s) \right) = e^{\lambda(G(r)-1)} - 1, \tag{3.7.12}$$

from (3.3.16). This expression shows the effect on the compound Poisson distribution function of the probabilities that are neglected in the tail of g.

In the case where $t = 2$, (3.7.3) is

$$\begin{aligned}
\Gamma\epsilon(s) &= \lambda \left(\Gamma^2 g(s) - \Gamma^2 g(r) - (s-r) G(r) \right) \\
&= \lambda \left(\mathrm{E}[(X-s-1)_+] - \mathrm{E}[(X-r-1)_+] + (s-r)(1-G(r)) \right), \quad s > r, \qquad (3.7.13)
\end{aligned}$$

which leads to

$$\eta_2(s) = \frac{e^{\lambda(G(s-1)-G(r))} - 1}{G(s-1) - G(r)} \left(\Gamma^2 g(s) - \Gamma^2 g(r) - (s-r) G(r) \right), \quad s > r+1. \tag{3.7.14}$$

In this case, (3.7.9) is

$$\Gamma\epsilon(s) = \lambda \left(\Gamma^2 g(s) - (s-r+1) \Gamma^2 g(r) + (s-r) \Gamma^2 g(r-1) \right), \quad s > r, \tag{3.7.15}$$

which gives another expression for the evaluation of $\eta_2(s)$.

The approximation defined in (3.7.1) does not correspond to the De Pril transform of a probability function with initial value $\widetilde{f}(0) = f(0) = e^{-\lambda}$ because the right-hand side of (3.7.12) is negative. In order to obtain a De Pril transform of a probability function with an exact initial value, Dhaene and Sundt (1997) proposed the approximation

$$\widetilde{\varphi}(y) = \begin{cases} \lambda y g(y), & y = 1, \ldots, r-1 \\ \lambda r(1 - G(r-1)), & y = r \\ 0, & y > r \end{cases}, \quad r \geq 1, \tag{3.7.16}$$

which gives

$$\psi(y) = \begin{cases} 0, & y = 1, \ldots, r-1 \\ \lambda(1 - G(r)), & y = r \\ -\lambda g(y), & y > r \end{cases}. \tag{3.7.17}$$

Such an approximation leads to an overestimation of $\Gamma^t f$ for $t \geq 1$ since we accumulate the probabilities of the tail of g at r. Thus, the intervals (3.5.17), (3.5.18), (3.5.19), (3.5.20) and (3.5.21) hold for this approximation.

We can observe that (3.7.2) and (3.7.17) are equal at each point except r. Therefore, we only need to include this error into the expressions derived for the approximation according to (3.7.1). Let $\delta_{a,b}$ be the Kronecker delta defined by

$$\delta_{a,b} = \begin{cases} 1, \text{ if } a = b \\ 0, \text{ if } a \neq b \end{cases}. \tag{3.7.18}$$

From (3.2.2), it follows that

$$\Gamma^t \delta_{s,b} = \sum_{a=0}^{s} \Gamma^{t-1} \delta_{a,b} = C_{s-b}^{t-1} = \binom{s-b+t-1}{s-b}, \quad b \geq 0,\ s \geq b,\ t \geq 0, \tag{3.7.19}$$

where $\Gamma^0 \delta_{a,b} \equiv \delta_{a,b}$. The inclusion of $|\psi(r)|$ into (3.7.3) using (3.7.19) gives

$$\Gamma^{t-1} \epsilon(s) = \lambda \left[C_{s-r}^{t-1} (1 - G(r)) + \Gamma^t g(s) - \sum_{k=1}^{t} C_{s-r-1}^{t-k} \Gamma^k g(r) \right], \quad s > r, \tag{3.7.20}$$

from which we can derive an expression similar to (3.7.9) by following the same way.

In the case where $t = 1$, we have

$$\epsilon(s) = \lambda \left(1 + G(s) - 2G(r)\right), \quad s > r. \tag{3.7.21}$$

An upper bound for the distance between the approximative and the exact probability functions is given by

$$\sum_{x=0}^{\infty} \left| \widetilde{f}(x) - f(x) \right| \quad \leq \quad e^{2\lambda(1-G(r))} - 1. \tag{3.7.22}$$

This upper bound is greater than the one obtained in the approximation according to (3.7.1) which is equal to the absolute value of the right-hand side of (3.7.12). From (3.3.16), we can verify that (3.7.16) is the De Pril transform of a probability function with an exact initial value since we have $\sum_{y=1}^{\infty} \psi(y) = 0$, which leads to

$$\lim_{s \to \infty} \left(\widetilde{F}(s) - F(s) \right) \quad = \quad 0. \tag{3.7.23}$$

3.7.2 Error bounds for the individual risk model

We consider now an insurance policies portfolio. The policies are grouped in different classes according to their probability that a claim occurs and the severity of this claim given that it occurs. We assume that there are n_{ij} policies in class (i, j) where the probability that a claim occurs is q_j, $j = 1, \ldots, b$, and where the probability function of the claim amount given it occurs is $g_i(x)$ for $x \geq 1$ and $i = 1, \ldots, a$. Dhaene et al. (2006) compared different methods for exact evaluations or approximations of the probability function of the aggregate claims amount of such a portfolio (see also Dhaene and Vandebroek (1995)).

The De Pril transform of the aggregate claims amount probability function is given by

$$\varphi(y) \quad = \quad \sum_{i=1}^{a} \sum_{j=1}^{b} n_{ij} \varphi_{ij}(y) = \sum_{i=1}^{a} \sum_{j=1}^{b} n_{ij} \, y \sum_{k=1}^{y} \frac{(-1)^{k+1}}{k} \left(\frac{q_j}{p_j} \right)^k g_i^{*k}(y), \quad y \geq 1, \tag{3.7.24}$$

where $p_j = 1 - q_j$, g_i^{*k} is the k-fold convolution of g_i and φ_{ij} is the De Pril transform of the probability function of the claim amount of a policy in class (i, j) (see De Pril (1989)). The De Pril transform φ_{ij} can be evaluated recursively using (3.2.5) with initial value $f_{ij}(0) = p_j$ for $i = 1, \ldots, a$ and $j = 1, \ldots, b$. If we assume that $q_j < \frac{1}{2}$, $j = 1, \ldots, b$, (3.7.24) converges to zero when y tends to infinity and an accurate approximation can be obtained by truncating the De Pril transform. The truncation of (3.7.24) from the point $r + 1$ leads to

$$\psi(y) \quad = \quad \begin{cases} 0, & y = 1, \ldots, r \\ \sum\limits_{i=1}^{a} \sum\limits_{j=1}^{b} n_{ij} \sum\limits_{k=1}^{y} \frac{(-1)^k}{k} \left(\frac{q_j}{p_j} \right)^k g_i^{*k}(y), & y > r \end{cases}, \quad r \geq 1. \tag{3.7.25}$$

De Pril (1989) and Dhaene and Sundt (1998, Application 8D) developed error bounds for an approximation such that the terms $\left(\frac{q_j}{p_j}\right)^k$ in (3.7.24) are set to zero for $k > z$. It is known under the name of De Pril's z-th order approximation.

Theorem 6 *Let* $\omega_i = \sup\{x : g_i(x) > 0\}$ *and* $r_i = \left\lfloor \frac{r}{\omega_i} \right\rfloor$ *for* $i = 1, \ldots, a$. *If we approximate the De Pril transform of the aggregate claims amount probability function according to (3.7.25) we obtain*

$$\begin{aligned} \Gamma^{t-1}\epsilon(s) \;\leq\; & \sum_{i=1}^{a}\sum_{j=1}^{b} \frac{n_{ij}}{r_i+1}\left[C_{s-r-1}^{t-1} \frac{q_j}{p_j-q_j}\left(\frac{q_j}{p_j}\right)^{r_i} + \left(\frac{q_j}{q_j-p_j}\right)^{t}\left(\frac{q_j}{p_j}\right)^{s} \right. \\ & \left. - \sum_{u=2}^{t} C_{s-r-1}^{t-u}\left(\frac{q_j}{q_j-p_j}\right)^{u}\left(\frac{q_j}{p_j}\right)^{r} \right], \quad s > r. \end{aligned} \tag{3.7.26}$$

The substitution of this expression into (3.4.2) gives an error bound for the approximation of the t-th order cumulative distribution function.

Proof. First, we prove (3.7.26) for $t = 1$. By following a similar way to the one in Dhaene et al. (2006, p. 552) for each policy in class (i, j), $i = 1, \ldots, a$, and $j = 1, \ldots, b$, we get

$$\begin{aligned} \epsilon_{ij}(s) \;&=\; \sum_{y=r+1}^{s} |\psi_{ij}(y)| = \sum_{y=r+1}^{s}\sum_{k=r_i+1}^{s} \frac{1}{k}\left(\frac{q_j}{p_j}\right)^k g_i^{*k}(y) \\ &\leq\; \frac{1}{r_i+1}\frac{q_j}{p_j-q_j}\left[\left(\frac{q_j}{p_j}\right)^{r_i} - \left(\frac{q_j}{p_j}\right)^{s}\right], \quad s > r, \end{aligned} \tag{3.7.27}$$

where $r_i = \left\lfloor \frac{r}{\omega_i} \right\rfloor$ and $\lfloor \cdot \rfloor$ is the floor function. If we sum (3.7.27) over all policies we obtain

$$\epsilon(s) \;\leq\; \sum_{i=1}^{a}\sum_{j=1}^{b} \frac{n_{ij}}{r_i+1}\frac{q_j}{p_j-q_j}\left[\left(\frac{q_j}{p_j}\right)^{r_i} - \left(\frac{q_j}{p_j}\right)^{s}\right], \quad s > r, \tag{3.7.28}$$

which is (3.7.26) with $t = 1$. We assume now that (3.7.26) holds for $t = v$. Given that

$$\sum_{y=r+1}^{s} C_{y-r-1}^{v-u} \;=\; C_{s-r-1}^{v-u+1}, \quad s > r,\ v \geq u, \tag{3.7.29}$$

and that

$$\sum_{y=r+1}^{s}\left(\frac{q_j}{p_j}\right)^{y} \;=\; \frac{q_j}{p_j-q_j}\left[\left(\frac{q_j}{p_j}\right)^{r} - \left(\frac{q_j}{p_j}\right)^{s}\right], \quad s > r, \tag{3.7.30}$$

we arrive to

$$\begin{aligned}
\Gamma^v\epsilon(s) &= \sum_{y=r+1}^{s}\Gamma^{v-1}\epsilon(y)\\
&\leq \sum_{y=r+1}^{s}\sum_{i=1}^{a}\sum_{j=1}^{b}\frac{n_{ij}}{r_i+1}\Bigg[C_{y-r-1}^{v-1}\frac{q_j}{p_j-q_j}\left(\frac{q_j}{p_j}\right)^{r_i}+\left(\frac{q_j}{q_j-p_j}\right)^{v}\left(\frac{q_j}{p_j}\right)^{y}\\
&\qquad -\sum_{u=2}^{v}C_{y-r-1}^{v-u}\left(\frac{q_j}{q_j-p_j}\right)^{u}\left(\frac{q_j}{p_j}\right)^{r}\Bigg]\\
&= \sum_{i=1}^{a}\sum_{j=1}^{b}\frac{n_{ij}}{r_i+1}\Bigg[C_{s-r-1}^{v}\frac{q_j}{p_j-q_j}\left(\frac{q_j}{p_j}\right)^{r_i}+\left(\frac{q_j}{q_j-p_j}\right)^{v+1}\left[\left(\frac{q_j}{p_j}\right)^{s}-\left(\frac{q_j}{p_j}\right)^{r}\right]\\
&\qquad -\sum_{u=2}^{v}C_{s-r-1}^{v+1-u}\left(\frac{q_j}{q_j-p_j}\right)^{u}\left(\frac{q_j}{p_j}\right)^{r}\Bigg],\quad s>r. \qquad (3.7.31)
\end{aligned}$$

Therefore, (3.7.26) holds for $t=v+1$ and the proof of (3.7.26) is complete by induction on t. Finally, it is sufficient to refer to Theorem 4 to complete the proof of Theorem 6. ■

Remark 7: One can say that the error bound of Theorem 6 grows too much with respect to t, but this is a bound for the absolute error of $\widetilde{\Gamma^t f}$ and the values of $\Gamma^t f$ also grow very fast with respect to t.

Remark 8: An upper bound for the expression (3.7.26) is given by

$$\Gamma^{t-1}\epsilon(s) \leq C_{s-r-1}^{t-1}\sum_{i=1}^{a}\sum_{j=1}^{b}\frac{n_{ij}}{r_i+1}\frac{q_j}{p_j-q_j}\left[\left(\frac{q_j}{p_j}\right)^{r_i}-\left(\frac{q_j}{p_j}\right)^{s}\right],\quad s>r. \quad (3.7.32)$$

It holds from (3.7.28) and (3.7.29) since $\epsilon(s)\geq\epsilon(y)$ for $s\geq y$. Notice that (3.7.26) and (3.7.32) hold for all $0<\left|q_j-\frac{1}{2}\right|<\frac{1}{2}$, $j=1,\ldots,b$. However, if at least one q_j is greater than $\frac{1}{2}$, a truncation may give very bad results because the De Pril transform may diverge. If we assume $q_j<\frac{1}{2}$, $j=1,\ldots,b$, we obtain from (3.7.28) that

$$\epsilon(s) \leq \lim_{s\to\infty}\epsilon(s) \leq \sum_{j=1}^{b}\frac{q_j}{p_j-q_j}\sum_{i=1}^{a}\frac{n_{ij}}{r_i+1}\left(\frac{q_j}{p_j}\right)^{r_i},\quad s>r, \qquad (3.7.33)$$

which leads to

$$\Gamma^{t-1}\epsilon(s) \leq C_{s-r-1}^{t-1}\sum_{j=1}^{b}\frac{q_j}{p_j-q_j}\sum_{i=1}^{a}\frac{n_{ij}}{r_i+1}\left(\frac{q_j}{p_j}\right)^{r_i},\quad s>r. \qquad (3.7.34)$$

Expressions (3.7.26), (3.7.32) and (3.7.34) are numerically close to each other if the values of the q_j's are small or when $s\gg r$.

Remark 9: It should be noted that the approximation defined in (3.7.25) is more accurate than the De Pril's z-th order approximation where $z = \min(\{r_i\}_{i=1}^a) \leq r$. Some terms $\left(\frac{q_j}{p_j}\right)^k$ with $k > z$ are taken into account such that the first r values are exact. In the De Pril's z-th order approximation, only the first z values are exact.

Remark 10: If we set $g_i(1) = 1$ such that $\omega_i = 1$ for $i = 1, \ldots, a$, (3.7.26) becomes

$$\Gamma^{t-1}\epsilon(s) \leq \sum_{i=1}^{a}\sum_{j=1}^{b}\frac{n_{ij}}{r+1}\left[\left(\frac{-q_j}{p_j-q_j}\right)^t\left(\frac{q_j}{p_j}\right)^s - \sum_{u=1}^{t}C_{s-r-1}^{t-u}\left(\frac{q_j}{q_j-p_j}\right)^u\left(\frac{q_j}{p_j}\right)^r\right], \quad (3.7.35)$$

for $s > r$. The substitution of (3.7.35) into (3.4.2) gives an error bound for the t-th order cumulative distribution function of the number of claims for which White and Greville (1959) derived an algorithm to calculate its probability function. In the limit $s \to \infty$ and for $t = 1$, we obtain the same error bound as De Pril (1989) if $q_j < \frac{1}{2}$, $j = 1, \ldots, b$.

3.7.3 Error bounds for approximations of the individual life model by compound Poisson distributions

In the actuarial literature, error bounds for approximations of the aggregate claims amount distribution in the individual risk model by compound Poisson distributions have been largely discussed (see e.g. De Pril and Dhaene (1992), Gerber (1984) and Hipp (1985, 1986)). We consider here the individual life model, a special case of the individual risk model described in Section 3.7.2. We assume that we have a portfolio of n life insurance policies where the policy number i has an amount at risk of b_i, $i = 1, \ldots, n$, such that

$$g_i(x) = \delta_{x,b_i}, \quad x \geq 1,\ i = 1, \ldots, n. \quad (3.7.36)$$

The De Pril transform of the claim amount of this policy is given by

$$\varphi_i(x) = \begin{cases} -b_i\left(-\frac{q_i}{p_i}\right)^{\frac{x}{b_i}}, & x = b_i, 2b_i, \ldots \\ 0, & \text{elsewhere} \end{cases}, \ i = 1, \ldots, n, \quad (3.7.37)$$

where q_i is the probability that a claim occurs for this policy and $p_i = 1 - q_i$. We derive now error bounds considering two methods of approximating the aggregate claims amount distribution of such a portfolio by a compound Poisson distribution.

In the first method, we approximate the aggregate claims amount by a compound Poisson such that $\lambda_i = |\ln p_i|$, $i = 1 \ldots, n$, in order to satisfy $\widetilde{f}(0) = f(0) = e^{-\lambda}$, where

$\lambda = \sum_{i=1}^{n} \lambda_i$. We have a compound Poisson distribution with parameters (λ, g) where g is given by

$$g(x) = \sum_{i=1}^{n} \frac{1}{\lambda} \delta_{x,b_i} \left|\ln p_i\right|, \quad x \geq 1. \tag{3.7.38}$$

The probability function of the aggregate claims amount is underestimated since more than one claim per policy may occur with compound Poisson distributions and since we have $\widetilde{f}(0) = f(0)$. Thus, $\Gamma^t f$ is also underestimated and the intervals (3.5.9), (3.5.10), (3.5.11), (3.5.15) and (3.5.16) hold for this approximation.

As a simplification we consider a portfolio which contains only the policy number i to derive expressions for this policy. The expressions for the portfolio with n policies will follow by adding them over all policies since the De Pril transform of a convolution is the sum of the De Pril transforms of the individual policies. The approximative De Pril transform of the claim amount of policy number i is given by

$$\widetilde{\varphi}_i(x) = b_i \delta_{x,b_i} \left|\ln p_i\right|, \quad x \geq 1. \tag{3.7.39}$$

From (3.7.37) and (3.7.39), it follows that

$$\psi_i(y) = \begin{cases} \left|\ln p_i\right| - \frac{q_i}{p_i}, & y = b_i \\ \frac{b_i}{y} \left(-\frac{q_i}{p_i}\right)^{\frac{y}{b_i}}, & y = 2b_i, 3b_i, \ldots \\ 0, & \text{elsewhere} \end{cases}. \tag{3.7.40}$$

If we assume that $q_i < \frac{1}{2}$, we have

$$\left|\psi_i(b_i)\right| = \ln p_i + \frac{q_i}{p_i}, \tag{3.7.41}$$

which leads to

$$\begin{aligned} \lim_{s\to\infty} \epsilon_i(s) &= \ln p_i + \sum_{y=1}^{\infty} \delta_{y \bmod b_i,0} \frac{b_i}{y} \left(\frac{q_i}{p_i}\right)^{\frac{y}{b_i}} = \ln p_i + \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{q_i}{p_i}\right)^{k} \\ &= \ln p_i - \ln\left(1 - \frac{q_i}{p_i}\right) = \ln\left(\frac{p_i^2}{p_i - q_i}\right), \end{aligned} \tag{3.7.42}$$

where $a \bmod b$ is the modulo operation. It follows that

$$\epsilon(s) = \sum_{i=1}^{n} \epsilon_i(s), \quad s \geq 1, \tag{3.7.43}$$

and that

$$\sum_{x=0}^{\infty} \left|\widetilde{f}(x) - f(x)\right| \leq \prod_{i=1}^{n} \frac{p_i^2}{p_i - q_i} - 1, \tag{3.7.44}$$

from Corollary 5. In this case, we can also develop an expression for $\epsilon(s)$ following the same approach as in the previous application. It is obtained by (3.7.43) with

$$\begin{aligned}\epsilon_i(s) &= \ln p_i + \sum_{y=1}^{s} \delta_{y \bmod b_i,0} \frac{b_i}{y} \left(\frac{q_i}{p_i}\right)^{\frac{y}{b_i}} \leq \ln p_i + \sum_{k=1}^{\left\lfloor \frac{s}{b_i} \right\rfloor} \left(\frac{q_i}{p_i}\right)^k \\ &= \ln p_i + \frac{q_i}{p_i - q_i}\left(1 - \left(\frac{q_i}{p_i}\right)^{\left\lfloor \frac{s}{b_i} \right\rfloor}\right), \quad s \geq 1,\ i = 1, \ldots, n. \end{aligned} \tag{3.7.45}$$

An expression for $\Gamma^{t-1}\epsilon(s)$ can be derived using the same way as in Theorem 6 such that we are able to determine an error bound for the t-th order cumulative distribution function using Theorem 4.

The second approximation concerns the common approximation where $\lambda_i = q_i$ such that the expected value of the approximation is exact. The condition $\widetilde{f}(0) = f(0)$ is no more fulfilled. However, we are able to determine a bound for the distance between the approximative and the exact probability functions from (3.3.24). In this case, the approximative De Pril transform of the claim amount of policy number i is given by

$$\widetilde{\varphi}_i(x) = b_i \delta_{x,b_i} q_i, \quad x \geq 1. \tag{3.7.46}$$

From (3.7.37) and (3.7.46), it follows that

$$\psi_i(y) = \begin{cases} q_i - \frac{q_i}{p_i}, & y = b_i \\ \frac{b_i}{y}\left(-\frac{q_i}{p_i}\right)^{\frac{y}{b_i}}, & y = 2b_i, 3b_i, \ldots \\ 0, & \text{elsewhere} \end{cases}. \tag{3.7.47}$$

If we assume that $q_i < \frac{1}{2}$, $i = 1, \ldots, n$, we have

$$|\psi_i(b_i)| = \frac{q_i}{p_i} - q_i, \tag{3.7.48}$$

which leads to

$$\begin{aligned}\lim_{s\to\infty} \epsilon_i(s) &= -q_i + \sum_{y=1}^{\infty} \delta_{y \bmod b_i,0} \frac{b_i}{y} \left(\frac{q_i}{p_i}\right)^{\frac{y}{b_i}} = -q_i + \sum_{k=1}^{\infty} \frac{1}{k} \left(\frac{q_i}{p_i}\right)^k \\ &= -q_i - \ln\left(1 - \frac{q_i}{p_i}\right). \end{aligned} \tag{3.7.49}$$

Since we have

$$\left|\ln \widetilde{f}(0) - \ln f(0)\right| = -q_i - \ln p_i, \tag{3.7.50}$$

we obtain from (3.3.24) and (3.7.43) that

$$\sum_{x=0}^{\infty}\left|\widetilde{f}(x)-f(x)\right| \;\leq\; \frac{\exp\left\{-2\sum_{i=1}^{n}q_i\right\}}{\prod_{i=1}^{n}(p_i-q_i)}-1. \tag{3.7.51}$$

This bound is close to the one derived in Gerber (1984), but it is always greater than it. Finally, the expected value of this approximation is exact since

$$\sum_{x=1}^{\infty}\varphi(x) \;=\; \sum_{x=1}^{\infty}\widetilde{\varphi}(x). \tag{3.7.52}$$

3.8 Numerical applications

In this section we implement numerically the results obtained in the applications of Sections 3.7.2 and 3.7.3. We consider the example of Gerber (1979) where we have a portfolio of life insurance policies that are grouped according to Table 3.1 and where the claim

Table 3.1: Number of policies in each class for the example of Gerber (1979)

j	q_j	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
1	0.03	2	3	1	2	-
2	0.04	-	1	2	2	1
3	0.05	-	2	4	2	2
4	0.06	-	2	2	2	1

amount distribution is $g_i(x)=\delta_{x,i}$ for $x\geq 1$ and $i=1,\ldots,5$.

First, we consider the application of Section 3.7.2. In this case, the numerical values of (3.7.26), (3.7.32) and (3.7.34) are extremely close to each other since the values of the q_j's are small. Table 3.2 contains exact values, approximations and error bounds for the t-th order cumulative distribution function for two points and for two truncation points. Notice that error bounds in the case where $s=97$ (Tables 3.2c and 3.2d) are also error bounds for the distance between the exact and the approximative $(t-1)$-th order cumulative distribution functions since 97 is the maximum point of the support of the aggregate claims amount distribution of this portfolio.

We consider now the application of Section 3.7.3. For this application, the portfolio contains 31 policies with parameters q_i and $b_i=i$, $i=1,\ldots,31$, given in Table 3.1. For

Table 3.2: Approximations of $\Gamma^t f(s)$ by setting $\varphi(y)$ for $y > r$

(a) $r = 5$ and $s = 20$

t	$\Gamma^t f(20)$	$\widetilde{\Gamma^t f}(20)$	$\Gamma^t \zeta(20)$	$\eta_t(20)$
1	0.99890	1.02402	0.02511	0.02935
2	16.5116	16.7483	0.2368	0.4403
3	152.193	153.594	1.400	3.522

(b) $r = 12$ and $s = 20$

t	$\Gamma^t f(20)$	$\widetilde{\Gamma^t f}(20)$	$\Gamma^t \zeta(20)$	$\eta_t(20)$
1	0.99890	0.99878	-0.000128	0.000236
2	16.5116	16.5111	-0.000496	0.001890
3	152.193	152.192	-0.001447	0.008503

(c) $r = 5$ and $s = 97$

t	$\Gamma^t f(97)$	$\widetilde{\Gamma^t f}(97)$	$\Gamma^t \zeta(97)$	$\eta_t(97)$
1	1	1.0261	0.02613	0.02935
2	93.51	95.757	2.2468	2.7003
3	4426.47	4524.41	97.948	125.566

(d) $r = 12$ and $s = 97$

t	$\Gamma^t f(97)$	$\widetilde{\Gamma^t f}(97)$	$\Gamma^t \zeta(97)$	$\eta_t(97)$
1	1	0.9998	-0.000187	0.000236
2	93.51	93.495	-0.01473	0.02008
3	4426.47	4425.88	-0.5888	0.8633

the approximation $\lambda_i = |\ln p_i|$, the bound (3.7.44) for the distance between the exact and the approximative probability functions is 0.07724 while the exact distance is 0.02449. For the common approximation $\lambda_i = q_i$, the bound (3.7.51) for the distance between the exact and the approximative probability functions is 0.15457 while the exact distance is 0.02629. The bound developed in Gerber (1984) is equal to 0.134 in the latter case.

3.9 Conclusion

In the recursive evaluation of t-th order cumulative distribution functions, a truncation of the De Pril transform may provide a considerable gain of time. Moreover, it gives more information about the distribution that may be useful, e.g., in relation to stop-loss contracts. When the De Pril transform converges to zero this approximation has a negligible effect on the accuracy of the evaluation. Moreover, a bound for the absolute error of the t-th order cumulative distribution function can be determined before the evaluation for any approximation of the De Pril transform as well as a bound for the distance between the exact values and the approximations of such functions. Both bounds hold for any kind of approximation if we are given the exact and approximative De Pril transforms.

Chapter 4

On the stability of recursive evaluations of t-th order cumulative distribution functions

4.1 Introduction

The aggregate claims amount of a portfolio of insurance policies can be modelled according to the so-called individual risk model. This model can be generalized to determine the distribution of a sum of independent random variables. Several exact or approximative recursive evaluations for its probability function are discussed in the actuarial literature (see e.g. De Pril (1986b, 1988, 1989), Dhaene and Vandebroek (1995), Hipp (1985, 1986), Kornya (1983) and Waldmann (1994)). These methods are reviewed in Sundt (2002) and Sundt and Vernic (2009). In comparison to an evaluation involving convolutions, recursions are good strategies to save time since they reduce the number of operations. Dhaene et al. (2006) and Dhaene and Vandebroek (1995) made some comparisons between the different methods of evaluation. It results that the Dhaene-Vandebroek algorithm is in many cases the most efficient among the exact recursive evaluations.

In the actuarial literature, recursions for the t-th order cumulative distribution function of a random variable were introduced by Dhaene et al. (1999). This function is useful to calculate some quantities like stop-loss premiums. Its recursive evaluation is based on the De Pril transform of its probability function. The recursive evaluation for convolutions of discrete uniform distributions and compound Poisson distributions are considered in Sundt (1999b) and Antzoulakos and Chadjiconstantinidis (2004), respectively. The lat-

ter paper also discusses recursive formulas for the t-th order cumulative tail function of compound Poisson distributions. Their results are improved in Chadjiconstantinidis and Pitselis (2009).

The De Pril transform is a function derived in De Pril (1989, p. 11) which leads to a two-stage recursive evaluation scheme under the individual risk model. This function was named by Sundt (1995) who expressed it for some special cases. The usefulness of De Pril transforms appears if we want to evaluate the probability function of a convolution since the De Pril transform of a convolution of functions is the sum of the De Pril transforms of these functions. Further results on De Pril transforms can be found in Sundt (1998) and Sundt and Ekuma (1999). An expression for the exact error of the t-th order cumulative distribution function resulting from an evaluation using an approximative De Pril transform is derived in Chapter 3 as well as an upper bound for this error. A more efficient recursive evaluation is also discussed that leads to accurate results when the De Pril transform converges to zero. Some additional results on error bounds in relation to De Pril transform approximations can be found in De Pril (1989), Dhaene and De Pril (1994), Dhaene and Sundt (1997, 1998), and Sundt et al. (1998).

Numerical problems may occur with recursive evaluations if we use the floating-point representation of numbers. Such a representation generates round-off errors whose propagation may lead to meaningless results. Oliver (1967) proposed a classification of error propagation based on the consideration of relative errors. In the actuarial literature, Panjer and Wang (1993) discussed the stability against round-off errors of the recursive evaluations developed in Panjer (1981) for the probability function of a family of compound distributions. The stability of the evaluations using recursive formulas discussed in Sundt (1992) is studied in Panjer and Wang (1995). An efficient method for the recursive evaluation of the probability function of a compound binomial distribution is discussed in Chapter 2. It consists in using an arbitrary-precision arithmetic library that allows the representation of real numbers by floating-point numbers with different precisions in a computer program. The precision is only limited by the available memory of the computer.

In Section 4.2, we present some exact evaluations and approximations of the t-th order cumulative distribution function and define the individual risk model with its associated recursive formulas. We extend the Dhaene-Vandebroek algorithm to the recursive evaluation of the t-th order cumulative distribution function in Section 4.3. In Section 4.4, we study the convergence of De Pril transforms and consider the case of some compound distributions. The stability against round-off errors of recursive evaluations of the t-th

order cumulative distribution function is studied in Section 4.5, where we find that the convergence or divergence rate of the De Pril transform is crucial to determine the effect of round-off errors. Finally, in Section 4.6, we illustrate numerically the results of Section 4.5.

In the sequel, we assume that the functions are defined on the nonnegative integers. The notations $a \geq b$ and $a > b$ are used to simplify the facts that $a = b, b+1, b+2, \ldots$ and $a = b+1, b+2, \ldots$, respectively, when a is used as a function argument. We also assume that $\sum_{k=a}^{b} h(k) = 0$ whenever $b < a$.

4.2 Definitions and review

4.2.1 Some exact evaluations and approximations of t-th order cumulative distribution functions

The t-th order cumulative operator Γ^t of a function h is defined by

$$\Gamma^t h(x) = \sum_{y=0}^{x} \Gamma^{t-1} h(y), \quad x \geq 0,\ t \geq 1, \tag{4.2.1}$$

with $\Gamma^0 h(x) = h(x)$. When h is a probability function of a random variable X, $\Gamma^t h$ is the t-th order cumulative distribution function of X. Some frequently used quantities like stop-loss transforms can be computed from some order of this function.

In the actuarial literature, recursive formulas for exact evaluations and approximations were developed for t-th order cumulative distribution functions. Dhaene et al. (1999) showed that this function can be evaluated by

$$\Gamma^t f(s) = \frac{1}{s} \sum_{y=1}^{s} (\varphi(y) + t)\, \Gamma^t f(s-y), \quad s \geq 1,\ t \geq 0. \tag{4.2.2}$$

where φ is the De Pril transform of f and with initial value $\Gamma^t f(0) = f(0)$. The De Pril transform is determined from the probability function by

$$\varphi(y) = \frac{1}{f(0)} \left(y f(y) - \sum_{x=1}^{y-1} \varphi(x) f(y-x) \right), \quad y \geq 1, \tag{4.2.3}$$

and defines uniquely a probability function given a value of it (e.g. $f(0)$). In Chapter 3, we showed that a lot of computing time can be saved by truncating the De Pril transform

from a given point such that

$$\widetilde{\varphi}(y) \;=\; \begin{cases} \varphi(y), & y = 1, \ldots, r \\ 0, & y > r \end{cases}, \; r \geq 1, \tag{4.2.4}$$

when $\varphi(y)$ converges to zero in the limit $y \to \infty$. It is equivalent to approximating $\Gamma^t f(s)$ by $\widetilde{\Gamma^t f}(s)$ using the embedded recursion

$$\widetilde{\Gamma^t f}(s) \;=\; \frac{1}{s}\left[t\,\widetilde{\Gamma^{t+1} f}(s-1) + \sum_{y=1}^{s \wedge r} \varphi(y) \widetilde{\Gamma^t f}(s-y) \right], \quad s \geq 1,\ t \geq 0, \tag{4.2.5}$$

where $a \wedge b = \min(a, b)$ and

$$\widetilde{\Gamma^{t+1} f}(s) \;=\; \widetilde{\Gamma^{t+1} f}(s-1) + \widetilde{\Gamma^t f}(s), \quad s \geq 1,\ t \geq 0, \tag{4.2.6}$$

with initial values $\widetilde{\Gamma^u f}(0) = f(0)$, $u = t, t+1$. Expressions for the exact error of $\widetilde{\Gamma^t f}(s)$ and a bound for its absolute value are derived in Chapter 3 for any De Pril transform approximation. Approximations of the form of (4.2.4) appear implicitly when floating-point numbers are used in evaluations. We explain why this is the case and discuss the propagation of round-off errors for such recursive evaluations in Section 4.5.

4.2.2 The individual risk model

The individual risk model is generally used to model the aggregate claims amount of a portfolio of insurance policies during a given period of time. The insurance policies are assumed to be mutually independent and grouped in different classes according to their probability that a claim occurs and their claim severity given that it occurs. We assume that there are n_{ij} policies in class (i, j) where the probability that a claim occurs is q_j, $j = 1, \ldots, b$, and the probability function of the claim amount given that it occurs is $g_i(x)$ for $x \geq 1$ and $i = 1, \ldots, a$. The aggregate claims amount random variable S is modelled as

$$S \;=\; \sum_{i=1}^{a} \sum_{j=1}^{b} \sum_{k=1}^{n_{ij}} I_{ijk} B_{ijk}, \tag{4.2.7}$$

where I_{ijk} and B_{ijk} are the indicator if a claim occurs and the conditional claim amount random variable for the policy number k in class (i, j), respectively. Notice that $I_{ijk} B_{ijk}$ follows a compound Bernouilli distribution and that $\sum_{k=1}^{n_{ij}} I_{ijk} B_{ijk}$ follows a compound binomial distribution.

Since the De Pril transform of a convolution of probability functions is the sum of the De Pril transforms of these probability functions, the t-th order cumulative distribution function can be evaluated by

$$\Gamma^t f(s) = \frac{1}{s}\left[t\,\Gamma^{t+1} f(s-1) + \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij} \sum_{y=1}^{s} \varphi_{ij}(y)\Gamma^t f(s-y)\right], \quad s \geq 1,\ t \geq 0, \tag{4.2.8}$$

where φ_{ij} is the De Pril transform of $I_{ijk}B_{ijk}$, $k = 1, \ldots, n_{ij}$ and

$$\Gamma^{t+1} f(s) = \Gamma^{t+1} f(s-1) + \Gamma^t f(s), \quad s \geq 1,\ t \geq 0. \tag{4.2.9}$$

The initial values of (4.2.8) and (4.2.9) are

$$\Gamma^u f(0) = \prod_{i=1}^{a}\prod_{j=1}^{b} (p_j)^{n_{ij}}, \quad u = t, t+1, \tag{4.2.10}$$

where $p_j = 1 - q_j$. De Pril (1989) showed that φ_{ij} can be determined either by

$$\varphi_{ij}(y) = \frac{q_j}{p_j}\left(y g_i(y) - \sum_{x=1}^{y-1} \varphi_{ij}(x) g_i(y-x)\right), \quad y \geq 1, \tag{4.2.11}$$

or by

$$\varphi_{ij}(y) = y \sum_{k=1}^{y} \frac{(-1)^{k+1}}{k} \left(\frac{q_j}{p_j}\right)^k g_i^{*k}(y), \quad y \geq 1, \tag{4.2.12}$$

where g_i^{*k} denotes the k-fold convolution of g_i.

We discuss now an important property of the individual risk model which is that we are able to know the exact value of $\Gamma^t f(\xi)$, $t \geq 0$, where ξ is the maximum value of the support of S. This result which is used later to measure the accuracy of a recursive evaluation using the floating-point representation is given in the following theorem.

Theorem 7 *Let* $\xi = \sup\{s : f(s) > 0\}$ *which is determined by*

$$\xi = \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij}\,\omega_i, \tag{4.2.13}$$

where $\omega_i = \sup\{x : g_i(x) > 0\}$. *We have*

$$\Gamma^t f(\xi) = \begin{cases} \prod_{i=1}^{a}\prod_{j=1}^{b} \left[q_j g_i(\omega_i)\right]^{n_{ij}}, & t = 0 \\ \frac{1}{(t-1)!}\,\mathrm{E}\left[(\xi - S + 1)^{\overline{t-1}}\right], & t \geq 1 \end{cases}, \tag{4.2.14}$$

where $\mathrm{E}\left[X^{\overline{k}}\right]$ *is the* k*-th rising factorial moment with* $\mathrm{E}\left[X^{\overline{0}}\right] = 1$.

Proof. For $t = 0$, the event $S = \xi$ takes place when a claim of amount ω_i occurs for each policy. Thus, (4.2.14) holds because the claim amounts of policies are assumed to be mutually independent.

From Antzoulakos and Chadjiconstantinidis (2004, p. 173), we know that

$$\Gamma^t f(s) \;=\; (-1)^t \Lambda^t f(s) + \sum_{j=0}^{t-1} (-1)^j \, C_x^{t-j-1} \, \frac{\mathrm{E}\big[S^{\underline{j}}\big]}{j!}, \quad s \geq 0, \; t \geq 0, \tag{4.2.15}$$

where $\mathrm{E}\big[X^{\underline{k}}\big]$ is the k-th falling factorial moment, $C_b^a = \binom{a+b}{a}$[1] and Λ^t is the t-th order tail operator, which is defined by

$$\Lambda^t h(x) \;=\; \sum_{y=x+1}^{\infty} \Lambda^{t-1} h(y), \quad x \geq 0, \; t \geq 1, \tag{4.2.16}$$

with $\Lambda^0 h(x) = h(x)$. Since $f(s) = 0$ for $s > \xi$, we have $\Lambda^t f(\xi) = 0$ for $t \geq 1$. It follows that

$$\begin{aligned} \Gamma^t f(\xi) \;&=\; \mathrm{E}\left[\sum_{j=0}^{t-1} (-1)^j \, C_\xi^{t-j-1} \, \frac{S^{\underline{j}}}{j!}\right] = \mathrm{E}\left[\sum_{j=0}^{t-1} (-1)^j \, C_\xi^{t-j-1} \, C_j^{S-j}\right] = \mathrm{E}\big[(-1)^{t-1} \, C_{S-\xi-t}^{t-1}\big] \\ &=\; \mathrm{E}\big[C_{\xi-S}^{t-1}\big] = \frac{1}{(t-1)!} \, \mathrm{E}\Big[(\xi - S + 1)^{\overline{t-1}}\Big], \quad t \geq 1, \end{aligned} \tag{4.2.17}$$

and the proof of Theorem 7 is complete. ■

4.3 Dhaene-Vandebroek algorithm for t-th order cumulative distribution functions

The Dhaene-Vandebroek algorithm derived in Dhaene and Vandebroek (1995) is another exact recursive evaluation for the probability function of S under the individual risk model. This algorithm is in many cases more efficient than the evaluation using the embedded recursion (4.2.8) with (4.2.11) and $t = 0$ (see Dhaene et al. (2006) and Dhaene and Vandebroek (1995)). In the following theorem we extend this algorithm to the recursive evaluation of t-th order cumulative distribution functions.

Theorem 8 *The t-th order cumulative distribution function of the aggregate claims amount under the individual risk model is obtained by*

$$\Gamma^t f(s) \;=\; \frac{1}{s} \left[t \, \Gamma^{t+1} f(s-1) + \sum_{i=1}^{a} \sum_{j=1}^{b} n_{ij} \Gamma^t v_{ij}(s) \right], \quad s \geq 1, \; t \geq 0, \tag{4.3.1}$$

[1] This is not the common notation for the binomial coefficient but our definition is used here in order to simplify expressions

where $\Gamma^{t+1}f(s-1)$ is given by (4.2.9) and the coefficients $\Gamma^t v_{ij}(s)$ are determined by

$$\Gamma^t v_{ij}(s) \;=\; \frac{q_j}{p_j}\sum_{k=1}^{s} g_i(k)\left(k\,\Gamma^t f(s-k) - \Gamma^t v_{ij}(s-k)\right). \tag{4.3.2}$$

The initial values for this algorithm are given by (4.2.10) and $\Gamma^t v_{ij}(0) = 0$, $t \geq 0$.

Proof. For $t = 0$, Theorem 8 is the Dhaene-Vandebroek algorithm.
We prove now recurrence equations (4.3.1) and (4.3.2) by induction on t. First, let us assume that (4.3.2) holds for $t = u$, we have

$$\begin{aligned}
\Gamma^{u+1}v_{ij}(s) &= \sum_{x=1}^{s}\Gamma^u v_{ij}(x) = \sum_{x=1}^{s}\frac{q_j}{p_j}\sum_{k=1}^{x} g_i(k)\left(k\,\Gamma^u f(x-k) - \Gamma^u v_{ij}(x-k)\right)\\
&= \frac{q_j}{p_j}\sum_{k=1}^{s} g_i(k)\left[k\sum_{x=k}^{s}\Gamma^u f(x-k) - \sum_{x=k}^{s}\Gamma^u v_{ij}(x-k)\right]\\
&= \frac{q_j}{p_j}\sum_{k=1}^{s} g_i(k)\left(k\,\Gamma^{u+1} f(s-k) - \Gamma^{u+1} v_{ij}(s-k)\right).
\end{aligned} \tag{4.3.3}$$

This proves that (4.3.2) holds for $t = u + 1$. The multiplication by s on both sides of (4.3.1) gives

$$s\,\Gamma^t f(s) \;=\; t\,\Gamma^{t+1} f(s-1) + \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij}\Gamma^t v_{ij}(s), \quad s \geq 1,\ t \geq 0. \tag{4.3.4}$$

Moreover, since $\Gamma^t h(y) = \Gamma^{t+1}h(y) - \Gamma^{t+1}h(y-1)$ for $y \geq 1$ and $t \geq 0$, we obtain using summation by parts that

$$\sum_{y=1}^{s} y\,\Gamma^t h(y) \;=\; s\,\Gamma^{t+1}h(s) - \Gamma^{t+2}h(s-1), \quad s \geq 1,\ t \geq 0. \tag{4.3.5}$$

Let us now assume that (4.3.4) holds for $t = u$. It follows successively from (4.3.5) and (4.3.4) that

$$\begin{aligned}
s\,\Gamma^{u+1}f(s) &= \Gamma^{u+2}f(s-1) + \sum_{y=1}^{s} y\,\Gamma^u f(y)\\
&= \Gamma^{u+2}f(s-1) + \sum_{y=1}^{s}\left[u\,\Gamma^{u+1}f(y-1) + \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij}\Gamma^u v_{ij}(y)\right]\\
&= (u+1)\Gamma^{u+2}f(s-1) + \sum_{i=1}^{a}\sum_{j=1}^{b} n_{ij}\Gamma^{u+1}v_{ij}(s).
\end{aligned} \tag{4.3.6}$$

This proves that (4.3.4) holds for $t = u + 1$. The equation (4.3.1) follows by dividing (4.3.6) by s on both sides, which completes the proof of Theorem 8. ■

The comparison between Theorem 8 and the recursive evaluation using (4.2.8) with (4.2.11) leads to the same conclusions as comparisons between the corresponding evaluations of the probability function that are made in Dhaene et al. (2006) and Dhaene and Vandebroek (1995). Similarly to (4.2.8) we also evaluate the $(t+1)$-th order cumulative distribution function, which gives more information about the distribution. Applications involving both orders may be developed, two of them are discussed in Chapter 3. We study the stability against round-off errors in the implementation of Theorem 8 in Section 4.5.

4.4 Convergence of De Pril transforms

4.4.1 General case

Let us now discuss the cases where the approximation by a truncation of the De Pril transform may be implemented or when it is involved in computations using the floating-point representation. The results derived in this section are important for Section 4.5 where we discuss the stability against round-off errors of the recursive evaluation of t-th order cumulative distribution functions.

Theorem 9 *Let $h(x)$, $x = 0, 1, \ldots, \omega$, be a probability function with $h(0) > 0$ and $h(\omega) > 0$. The De Pril transform of h is given by*

$$\varphi(y) = -\sum_{k=1}^{\omega} \lambda_k^y, \quad y \geq 1, \tag{4.4.1}$$

where $\lambda_1, \ldots, \lambda_\omega$ are the roots of the polynomial

$$\sum_{x=0}^{\omega} h(\omega - x)\lambda^x. \tag{4.4.2}$$

Proof. The De Pril transform of a probability function h is the solution of (4.2.3). However, it is also the solution of the ω-th order recurrence equation

$$\varphi(y) = -\frac{1}{h(0)} \sum_{x=1}^{\omega} \varphi(y-x)h(x), \quad y > \omega, \tag{4.4.3}$$

with initial values $\varphi(1), \ldots, \varphi(\omega)$ that are determined by the first ω steps of (4.2.3). The characteristic polynomial of (4.4.3) is

$$\lambda^\omega + \sum_{x=1}^{\omega} \frac{h(x)}{h(0)} \lambda^{\omega - x}, \tag{4.4.4}$$

which has the same roots as (4.4.2). We assume that (4.4.4) has the roots $\mu_1, \ldots, \mu_l$, $l \leq \omega$, with multiplicities $\alpha_1, \ldots, \alpha_l$, respectively. The solution of (4.4.3) is given by

$$\varphi(y) \;=\; \sum_{k=1}^{l} \mu_k^y \sum_{u=0}^{\alpha_k - 1} C_{ku}\, y^u, \quad y > \omega, \tag{4.4.5}$$

where the ω coefficients C_{ku}'s are uniquely determined from ω initial values. From Newton's identities, we are able to determine recursively $s_k = \sum_{i=1}^{n} r_i^k$, where r_i, $i = 1, \ldots, n$, are the roots of the polynomial $x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0$ by

$$s_k \;=\; -k a_{n-k} - \sum_{j=1}^{k-1} a_{n-j} s_{k-j}, \quad k = 1, \ldots, n. \tag{4.4.6}$$

Note the similarity of (4.4.6) with (4.2.3). Thus, by substituting $n = \omega$ and $a_{\omega - j} = \frac{h(j)}{h(0)}$ into (4.4.6) we obtain that $\varphi(y) = -s_y$, $y = 1, \ldots, \omega$. Therefore, using the initial values of (4.4.3) we find that

$$C_{ku} \;=\; \begin{cases} -\alpha_k, & u = 0 \\ 0, & \text{otherwise} \end{cases}, \; k = 1, \ldots, l. \tag{4.4.7}$$

It follows that the De Pril transform of h is given by (4.4.1) where $\lambda_1, \ldots, \lambda_\omega$ are the roots of (4.4.4) and the proof of Theorem 9 is complete. ■

Corollary 10 *Let $h(x)$, $x = 0, 1, \ldots, \omega$, be a probability function with $h(0) > 0$ and $h(\omega) > 0$. The De Pril transform of h converges to zero in absolute value if the roots of (4.4.2) satisfy $|\lambda_k| < 1$, $k = 1, \ldots, \omega$, where $|z|$ denotes the complex modulus of z.*

Proof. The De Pril transform of a probability function is given by Theorem 9 but the roots of (4.4.2) may be complex. Using the polar representation of complex numbers in (4.4.1), we get

$$\varphi(y) \;=\; -\sum_{k=1}^{\omega} |\lambda_k|^y \left(\cos(y\,\theta_k) + i \sin(y\,\theta_k)\right), \quad y \geq 1, \tag{4.4.8}$$

where θ_k is the complex argument of λ_k. Therefore, the De Pril transform converges to zero in absolute value if $|\lambda_k| < 1$ for $k = 1, \ldots, \omega$. ■

Corollary 11 *The De Pril transform of a convolution of probability functions converges to zero if the De Pril transform of each probability function satisfies Corollary 10.*

Proof. The proof is immediate since the De Pril transform of a convolution of probability functions is the sum of the De Pril transforms of the probability functions and that the sum of convergent sequences converges in absolute value to the sum of their limits which are here equal to zero. ■

Notice that if we use Corollary 10 in order to determine if the approximation of the t-th order cumulative distribution function obtained by truncating the De Pril transform is effective, we have to know its associated probability function. However, from Corollary 11 we are able to know if such an approximation for a convolution is appropriate before evaluating the De Pril transforms. We are also able to determine in advance if the De Pril transform of a compound distribution is convergent. This point is discussed in the next subsection.

4.4.2 Convergence of De Pril transforms of some compound distributions

Compound distributions are frequently used to model the aggregate claims amount distribution. Such a model is known under the name of collective risk model in the actuarial literature. The aggregate claims amount random variable is defined as

$$S = X_1 + X_2 + \cdots + X_N, \tag{4.4.9}$$

where N and X_i represent the number of claims and the amount of the i-th claim, respectively. The X_i's and N are assumed to be mutually independent. We assume that the X_i's are identically distributed with common probability function $g(x)$, $x = 1, \ldots, \omega$.

Corollary 10 holds for compound Poisson and compound negative binomial distributions. These results follow from Theorem 6.1 in Dhaene and Sundt (1998) since the De Pril transforms of the Poisson and the negative binomial distributions are bounded and convergent. For compound Poisson distributions, we even have $\varphi(y) = 0$ for $y > \omega$ (see e.g. Sundt (1995, p. 25)). Corollary 10 does not hold for all compound binomial distributions, the convergence of their De Pril transforms depends on the shape of g and on the parameters of the binomial distribution. Sundt and Ekuma (1999, p. 183) developed recursive forms for the De Pril transforms of compound negative binomial and compound binomial distributions that we use now to express these De Pril transforms as sums of polynomial roots raised to a power.

First, we consider compound negative binomial distributions. Let N be a negative bino-

mial distribution with parameters $r > 0$ and $0 < q < 1$ such that

$$\Pr[N = n] = \binom{r+n-1}{n}(1-q)^r q^n, \quad n \geq 0, \tag{4.4.10}$$

Sundt and Ekuma (1999, p. 183) showed that the De Pril transform of a compound negative binomial probability function can be evaluated by

$$\varphi(y) = qryg(y) + q\sum_{x=1}^{\omega\wedge(y-1)} g(x)\varphi(y-x), \quad y \geq 1. \tag{4.4.11}$$

This De Pril transform is the solution of

$$\varphi(y) = q\sum_{x=1}^{\omega} g(x)\varphi(y-x), \quad y > \omega, \tag{4.4.12}$$

with initial values $\varphi(1), \ldots, \varphi(\omega)$ determined by (4.4.11). The corresponding characteristic polynomial is

$$\lambda^\omega - q\sum_{x=1}^{\omega} g(x)\lambda^{\omega-x}. \tag{4.4.13}$$

By substituting $n = \omega$ and $a_{w-j} = -qg(j)$ into (4.4.6), and multiplying it by r, we obtain from (4.4.11) that

$$\varphi(y) = r\sum_{k=1}^{\omega} \lambda_k^y, \quad y \geq 1, \tag{4.4.14}$$

where the λ_k's are the roots of (4.4.13). Moreover, from Descartes' rule of signs, this polynomial has exactly one positive real root, let it be λ_1. Furthermore, $\lambda_1 < 1$ because $\sum_{x=1}^{\omega} qg(x) = q < 1$. Since Cauchy (1829, p. 122), we know that

$$|\lambda_k| \leq \lambda_1, \quad k = 2, \ldots, \omega, \tag{4.4.15}$$

which proves that the De Pril transform of a compound negative binomial distribution always converges to zero.

We consider now the case of compound binomial distributions. Let N be a binomial distribution with parameters m and $0 < q < 1$ such that

$$\Pr[N = n] = \binom{m}{n} q^n p^{m-n}, \quad n = 0, 1, \ldots, m, \tag{4.4.16}$$

where $p = 1 - q$ and m is a positive integer. Sundt and Ekuma (1999, p. 183) showed that the De Pril transform of a compound binomial probability function can be evaluated by

$$\varphi(y) = \frac{mqyg(y)}{p} - \frac{q}{p}\sum_{x=1}^{\omega\wedge(y-1)} g(x)\varphi(y-x), \quad y \geq 1. \tag{4.4.17}$$

This De Pril transform is the solution of

$$\varphi(y) \;=\; -\frac{q}{p}\sum_{x=1}^{\omega} g(x)\varphi(y-x), \quad y>\omega, \tag{4.4.18}$$

with initial values $\varphi(1),\ldots,\varphi(\omega)$ determined by (4.4.17). The characteristic polynomial of (4.4.18) is

$$\lambda^{\omega}+\frac{q}{p}\sum_{x=1}^{\omega} g(x)\lambda^{\omega-x}. \tag{4.4.19}$$

By substituting $n=\omega$ and $a_{w-j}=\frac{q}{p}g(j)$ into (4.4.6), and multiplying it by $-m$ we obtain from (4.4.17) that

$$\varphi(y) \;=\; -m\sum_{k=1}^{\omega}\lambda_k^{y}, \quad y\geq 1, \tag{4.4.20}$$

where the λ_k's are the roots of (4.4.19) that does not depend on m. From Cauchy (1829, p. 122), we have that (4.4.20) converges to zero when $\frac{q}{p}<1$, which is equivalent to $q<\frac{1}{2}$. However, in this case the corresponding upper bound of (4.4.15) is not a root of (4.4.19) since it has no positive root. This means that (4.4.20) also converges for some $q\geq\frac{1}{2}$ depending on g. The following corollary shows this property for the case where $\omega=2$ and $m=1$.

Corollary 12 *For an individual claim amount random variable with a probability that a claim occurs of q and a conditional claim amount distribution satisfying $g(1)+g(2)=1$, the De Pril transform of its probability function converges to zero if*

$$q \;<\; \min\left(\frac{1}{2-g(1)},\frac{1}{2g(1)}\right), \quad 0\leq g(1)\leq 1. \tag{4.4.21}$$

Proof. In this case, the claim amount distribution follows a compound Bernouilli distribution with parameter q and conditional claim amount probability function g such that $g(2)=1-g(1)$. The roots of (4.4.19) are

$$\lambda_1=\frac{-qg(1)-\sqrt{q^2g(1)^2-4qpg(2)}}{2p} \quad\text{and}\quad \lambda_2=\frac{-qg(1)+\sqrt{q^2g(1)^2-4qpg(2)}}{2p}. \tag{4.4.22}$$

If $q<\frac{4(1-g(1))}{(2-g(1))^2}$, both roots are complex with $|\lambda_1|=|\lambda_2|$. Therefore, we have $|\lambda_1|\geq|\lambda_2|$ for any $(g(1),q)$ in the unit square and in order to determine if the De Pril transform converges it is sufficient to find for which values of $(g(1),q)$ the inequality $|\lambda_1|<1$ holds. In the case of complex roots, one can show that $|\lambda_1|<1$ is satisfied over the range

$q < (2 - g(1))^{-1}$. Together with the condition of having complex roots, the range of convergence is

$$q < \min\left(\frac{1}{2-g(1)}, \frac{4\left(1-g\left(1\right)\right)}{\left(2-g\left(1\right)\right)^2}\right), \quad 0 \leq g(1) \leq 1. \tag{4.4.23}$$

In the case of real roots, one can show that $|\lambda_1| < 1$ holds if $q < 2 \cdot (2 + g(1))^{-1}$ and $q < (2g(1))^{-1}$. Together with the condition of having real roots, the range of convergence becomes

$$\frac{4\left(1-g\left(1\right)\right)}{\left(2-g\left(1\right)\right)^2} < q < \frac{1}{2g(1)}, \quad \frac{2}{3} \leq g(1) \leq 1. \tag{4.4.24}$$

The union of (4.4.23) and (4.4.24) leads to (4.4.21) and the proof is complete. Figure 4.1 shows the different functions encountered above and the area where the De Pril transform is convergent in this particular case. ■

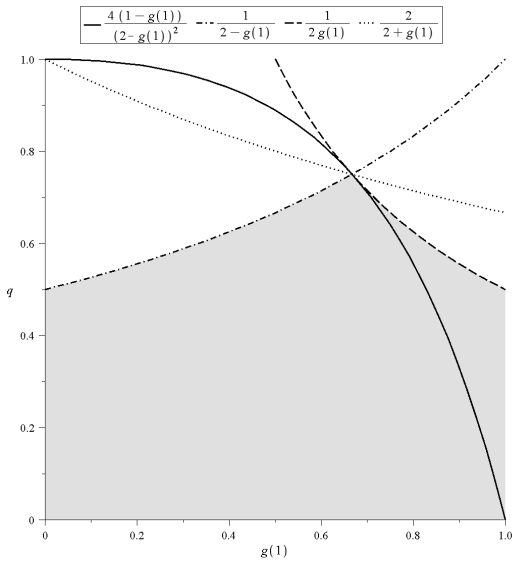


Figure 4.1: Area where the De Pril transform is convergent in the case of Corollary 12

From Corollary 11 and (4.4.20), it follows that the De Pril transform of the aggregate claims amount under the individual model is given by

$$\varphi(y) = -\sum_{i=1}^{a}\sum_{j=1}^{b}\sum_{k=1}^{\omega_i} n_{ij}\lambda_{ijk}^{y}, \quad y \geq 1, \tag{4.4.25}$$

where λ_{ijk}, $k = 1, \ldots, \omega_i$, are the roots of

$$p_j\lambda^{\omega_i} + q_j\sum_{x=1}^{\omega_i} g_i(x)\lambda^{\omega_i - x}, \tag{4.4.26}$$

for $i = 1, \ldots, a$ and $j = 1, \ldots, b$. Therefore, (4.4.25) converges to zero if the modulus of every root is smaller than one.

Remark 11: If $a = 1$ and $w_i = 1$ such that $g(1) = 1$, the roots of (4.4.26) are $\lambda_{1j1} = -\frac{p_j}{q_j}$, $j = 1, \ldots, b$. Moreover, if $n_{1j} = 1$, $j = 1, \ldots, b$, a recursive evaluation of the probability function equivalent to (4.2.2) with (4.4.25) and $t = 0$ is given by formulas (7) and (8) of White and Greville (1959).

4.5 Stability against round-off errors

4.5.1 Some definitions and general results

In this section, we study the propagation of round-off errors when floating-point numbers are used for the evaluation of $\Gamma^t f$ according to the recursive formulas considered above. The use of floating-point numbers is highly recommended to save time when a large number of computations have to be done. A floating-point number is generally represented as $\pm s_0 s_1 s_2 \ldots s_{\rho-1} \cdot \beta^e$ according to a base $\beta \geq 2$ and a precision ρ. The sequence of integers $s_0 s_1 s_2 \ldots s_{\rho-1}$ is called the *significand* or mantissa and the integer e is the exponent. The floating-point number $\pm s_0 s_1 s_2 \ldots s_{\rho-1} \cdot \beta^e$ represents the number

$$\pm \beta^e \sum_{j=0}^{\rho-1} s_j \beta^{-j}, \quad s_j \in \{0, 1, \ldots, \beta - 1\}. \tag{4.5.1}$$

We denote by ε the maximal relative error of the floating-point representation $\widetilde{x}$ of a real number x if the computer rounds to the nearest floating-point number. It is defined by

$$\left| \frac{x - \widetilde{x}}{x} \right| \leq \frac{\beta^{1-\rho}}{2} = \varepsilon. \tag{4.5.2}$$

The propagation of such round-off errors in a recursive evaluation may grow without bound and may lead to meaningless values. The relative error is used as a measure of accuracy. The accuracy is defined to be the number of decimal digits that are exact in an evaluation. We refer to Oliver (1967) for a classification of stability of recursive evaluations. Panjer and Wang (1993) showed that the evaluation of a recursion of the form

$$h(s) = b(s) + \sum_{y=1}^{s} a_y(s) h(s - y), \quad s \geq 1, \tag{4.5.3}$$

is strongly stable if $b(s)$, the coefficients $a_y(s)$ and the initial values are nonnegative.

Dhaene et al. (1999) wrote that for any random variable with probability function h, there exists an order u such that the recursive evaluation of $\Gamma^t h(z)$ using (4.2.2) is strongly stable for $t \geq u$. Due to Panjer and Wang (1993), a sufficient condition for u is that

$$u + \varphi(y) \geq 0, \quad y = 1, 2, \ldots, z. \tag{4.5.4}$$

On the one hand, if φ converges to zero but has negative values then u may be quite small. Its value will depend on the first values of the De Pril transform. On the other hand, if φ diverges and has negative values, the order u will be extremely large and close to $\varphi(z)$. However, if we desire to evaluate $\Gamma^t h(z)$ with $t < u$ the first digits of the significand of $\Gamma^u h(z)$ cancel out by taking the finite differences to obtain that order because $\Gamma^t h(z)$ is smaller than $\Gamma^u h(z)$. The evaluation obtained for $\Gamma^t h(z)$ is not better in terms of accuracy than the one resulting from a recursive evaluation for the order t, which avoids the computation of finite differences. Notice that for compound Poisson and compound negative binomial distributions (4.5.4) holds for $u = 0$ since their De Pril transforms are nonnegative. It follows that the recursive evaluation of $\Gamma^t h$ is stable for any positive order. Panjer and Wang (1993) showed this result for a recursive evaluation with $t = 0$.

Panjer and Wang (1993) defined the notions of dominant and subordinate solutions of a recurrence equation and showed that the evaluation of a dominant solution is stable while evaluating a subordinate solution is unstable (see also Chapter 2). Under the individual risk model, the recurrence equation (4.2.8) with (4.2.11) or (4.2.12) and the system of recurrence equations of Theorem 8 have the same set of solutions since they are built from the same power series (see Dhaene et al. (2006, pp. 546-547)). Therefore, their solutions with their respective initial values are equal and the propagation of round-off errors behaves similarly in both cases. The same results are obtained by considering the recursion of the form of (4.2.2) under the individual risk model. Thus, if we know that one of these recursive evaluations is unstable then the other ones are unstable and conversely. In general the evaluation according to Theorem 8 produces smaller relative errors but not significantly. Moreover, as pointed out by Dhaene and Vandebroek (1995), this algorithm needs to store a smaller number of values at each evaluation stage of the recursion. This difference is reduced if the De Pril transform can be truncated according to (4.2.4). For all these reasons we conclude that the evaluation of the t-th order cumulative distribution function according to Theorem 8 is the most efficient amongst the recursions considered in this paper. Since it is easier to study the propagation of round-off errors for recurrence equations (4.2.2) and (4.2.5) than for Theorem 8, we focus on these two recursive evaluations and the results will also hold for Theorem 8.

4.5.2 Stability with a convergent De Pril transform

When floating-point numbers are used in the evaluation of (4.2.2) or (4.2.5), the De Pril transform cannot be represented exactly and we implicitly have an approximative De Pril transform. An expression for the error of $\widetilde{\Gamma^t f}$ resulting from a recursive evaluation using any De Pril transform approximation $\widetilde{\varphi}$ is developed in Chapter 3. This expression is

$$\widetilde{\Gamma^t f}(s) - \Gamma^t f(s) \;=\; \sum_{y=1}^{s}\sum_{n=1}^{y} \frac{1}{n!}\,\psi^{*n}(y)\Gamma^t f(s-y), \quad s \geq 0,\ t \geq 0, \tag{4.5.5}$$

where we assume $\widetilde{\Gamma^t f}(0) = \Gamma^t f(0)$, $t \geq 0$, and ψ is defined by

$$\psi(x) \;=\; \begin{cases} 0, & x = 0 \\ \frac{\widetilde{\varphi}(x)-\varphi(x)}{x}, & x \geq 1 \end{cases}. \tag{4.5.6}$$

Since $\Gamma^t f$ is a nondecreasing function for $t \geq 1$, an upper bound for the relative error of $\widetilde{\Gamma^t f}(s)$ follows from (4.5.5). It is

$$\begin{aligned} \left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \;&\leq\; \sum_{y=1}^{s}\sum_{n=1}^{y} \frac{1}{n!}\,|\psi^{*n}(y)| \leq \exp\left\{\sum_{y=1}^{s}|\psi(y)|\right\} - 1 \\ &\leq\; \exp\left\{\sum_{y=1}^{\infty}|\psi(y)|\right\} - 1, \quad s \geq 0,\ t \geq 1. \end{aligned} \tag{4.5.7}$$

However, this error bound holds only if there is not any round-off errors propagation during the recursive evaluation for example by setting an "infinite" precision to the variables representing the t-th order cumulative distribution function.

If the computer uses a precision of ρ to represent the De Pril transform, we obtain from (4.5.2) that

$$|\varphi(y) - \widetilde{\varphi}(y)| \;\leq\; \varepsilon\,|\varphi(y)|, \quad y \geq 1, \tag{4.5.8}$$

where $\widetilde{\varphi}$ denotes the floating-point representation of φ. If we evaluate the t-th order cumulative distribution function, whose variables are assumed to be represented using an "infinite" precision, it follows that

$$\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \;\leq\; \exp\left\{\varepsilon\sum_{y=1}^{\infty}\frac{|\varphi(y)|}{y}\right\} - 1, \quad s \geq 0,\ t \geq 1. \tag{4.5.9}$$

From (4.4.8) we have

$$|\varphi(y)| \;\leq\; \sum_{k}|\lambda_k|^y, \quad y \geq 1, \tag{4.5.10}$$

where the λ_k's are the roots of the characteristic polynomial of (4.2.3). Under the individual risk model, we can see from (4.4.25) that the sum of (4.5.10) contains ω_i roots per policy. Assuming that Corollary 10 holds for this De Pril transform, the substitution of (4.5.10) into (4.5.9) leads to

$$\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \leq \exp\left\{\varepsilon \sum_{y=1}^{\infty} \sum_k \frac{|\lambda_k|^y}{y}\right\} - 1 = \prod_k (1 - |\lambda_k|)^{-\varepsilon} - 1, \qquad (4.5.11)$$

for $s \geq 0$ and $t \geq 1$. Using the first order multivariate Taylor series expansion we get

$$\prod_k (1 - |\lambda_k|)^{-\varepsilon} \approx 1 + \varepsilon \sum_k |\lambda_k|. \qquad (4.5.12)$$

It follows that

$$\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \leq \varepsilon \sum_k |\lambda_k|, \quad s \geq 0,\ t \geq 1. \qquad (4.5.13)$$

which shows that the recursive evaluation of $\Gamma^t f$ for $t \geq 1$ obtained from the representation of a convergent De Pril transform using a reasonable precision is accurate even though $t + \varphi(y)$ is negative for some y.

The idea of truncating a convergent De Pril transform comes from looking at (4.2.2). When floating-point numbers are used for the evaluation of $\Gamma^t f$ using (4.2.2), the value of $\varphi(y) + t$ is represented by t when $|\varphi(y)| \leq t\varepsilon$. We denote by r_t the maximal point for which the term $\varphi(y) + t$ is not represented only by t, it is determined by

$$r_t = \sup\{y \geq 1 : |\varphi(y)| > t\varepsilon\}, \quad t \geq 1. \qquad (4.5.14)$$

This is equivalent to setting $\varphi(y) = 0$ for $y > r_t$. In this case it is more efficient to evaluate $\Gamma^t f$ using (4.2.5) where r is given by (4.5.14) instead of (4.2.2) because time can be saved without losing accuracy. For such an evaluation, (4.5.9) becomes

$$\begin{aligned}
\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| &\leq \exp\left\{\varepsilon \sum_{y=1}^{r_t} \frac{|\varphi(y)|}{y} + \sum_{y=r_t+1}^{\infty} \frac{|\varphi(y)|}{y}\right\} - 1 \\
&\leq \exp\left\{\varepsilon \sum_{y=1}^{\infty} \sum_k \frac{|\lambda_k|^y}{y} + \frac{1-\varepsilon}{r_t+1} \sum_{y=r_t+1}^{\infty} \sum_k |\lambda_k|^y\right\} - 1 \\
&= \exp\left\{\frac{1-\varepsilon}{r_t+1} \sum_k \frac{|\lambda_k|^{r_t+1}}{1-|\lambda_k|}\right\} \prod_k (1 - |\lambda_k|)^{-\varepsilon} - 1, \qquad (4.5.15)
\end{aligned}$$

for $s \geq 0$ and $t \geq 1$. Using the $(r_t + 1)$-th order multivariate Taylor series expansion, we obtain

$$\exp\left\{\frac{1-\varepsilon}{r_t+1} \sum_k \frac{|\lambda_k|^{r_t+1}}{1-|\lambda_k|}\right\} \approx 1 + \frac{1-\varepsilon}{r_t+1} \sum_k |\lambda_k|^{r_t+1} \approx 1 + \frac{t\varepsilon}{r_t+1}, \quad t \geq 1, \qquad (4.5.16)$$

since $\sum_k |\lambda_k|^{r_t+1} \approx t\varepsilon$ from (4.5.14). The substitution of (4.5.12) and (4.5.16) into (4.5.15) leads to

$$\exp\left\{\frac{1-\varepsilon}{r_t+1}\sum_k \frac{|\lambda_k|^{r_t+1}}{1-|\lambda_k|}\right\}\prod_k (1-|\lambda_k|)^{-\varepsilon} - 1 \approx \varepsilon\sum_k |\lambda_k| + \frac{t\varepsilon}{r_t+1}, \quad t \geq 1. \tag{4.5.17}$$

Therefore, the recursive evaluation using (4.2.5) with r given by (4.5.14) is accurate for $t \geq 1$ if we use a reasonable precision and if the De Pril transform converges to zero.

Until now we considered only the effect of round-off errors coming from the representation of φ by floating-point numbers. However, it is important to consider also the propagation of such errors as we go along with the recursive evaluation of $\Gamma^t f$ using a "finite" precision ρ. The recursions (4.2.2) and (4.2.5) are both of the form of (4.5.3) therefore the propagation of round-off errors as we go along with these recursions is similar and does not create numerical problems. Such problems may arise with (4.2.5) when $\varphi(y)$ is negative for some y's, which is the case under the individual risk model. They happen when a difference between two numbers is significantly smaller in absolute value than both numbers, which means that the first digits of the significands of both numbers vanish and the exact significand of the difference has a smaller precision. Here, since $s\,\Gamma^t f(s) \geq t\,\Gamma^{t+1} f(s-1) \geq 0$ for $s \geq 1$ and $t \geq 0$, the two numbers that may cause such problems are the maximum and minimum values of the products $\varphi(y)\Gamma^t f(s-y)$ over $y = 1, 2, \ldots, s$. However, such problems are already taken into account in (4.5.15). Therefore, the evaluation of $\Gamma^t f$ using the floating-point representation is strongly stable for $t \geq 1$ when φ converges to zero. In this case, the values of $y_{s,t}^-$ and $y_{s,t}^+$ defined such that

$$\begin{array}{rcl} \pi_{s,t}^- = \varphi(y_{s,t}^-)\,\Gamma^t f(s-y_{s,t}^-) & = & \min\limits_{y=1,\ldots,s}(\varphi(y)\,\Gamma^t f(s-y)) \\ \pi_{s,t}^+ = \varphi(y_{s,t}^+)\,\Gamma^t f(s-y_{s,t}^+) & = & \max\limits_{y=1,\ldots,s}(\varphi(y)\,\Gamma^t f(s-y)) \end{array}, \quad s \geq 1,\ t \geq 0, \tag{4.5.18}$$

are small for any $s \geq 1$ and remain constant when s becomes large. Moreover, the products $\pi_{s,t}^-$ and $\pi_{s,t}^+$ are of the same magnitude as $s\,\Gamma^t f(s)$, for $s \geq 1$. Remark that as long as $\pi_{s,t}^-$ is positive, the recursion satisfies the condition of stability of Panjer and Wang (1993).

The last point concerns the accuracy of the evaluation of the De Pril transform. Under the individual risk model, if we have

$$\lambda_{ijk} \neq \lambda_{ijl}, \quad k \neq l, \tag{4.5.19}$$

in (4.4.25) the solution of (4.2.11) is dominant and its evaluation is strongly stable. If this holds for each class of policies, the De Pril transform of the probability function of

S is accurate. However, if it does not hold for all classes, a subordinate solution has to be evaluated but it is generally accurate up to the implicit truncation point. Another way to calculate the De Pril transform is using (4.4.25), which is more efficient than an evaluation using (4.2.12) but less efficient than the recursive evaluation with (4.2.11).

Notice that (4.5.7) does not hold for $t = 0$ since the probability function is nonincreasing. In general the probability function under the individual risk model is roughly concave and unimodal. For the points above the mode, the probability function decreases exponentially involving that $f(s)$ converges to zero at a decreasing rate with respect to s, i.e. $\frac{f(s+1)}{f(s)}$ decreases with respect to s if s is above the mode. In this case, numerical problems arise in the recursive evaluation from the point c where the convergence rate of f becomes smaller than τ, which is the convergence rate of τ^y with respect to y. The point c is then defined by

$$\frac{f(s+1)}{f(s)} \leq \tau, \quad \text{for all } s \geq c, \tag{4.5.20}$$

where

$$\tau = \max_k |\lambda_k|, \tag{4.5.21}$$

where the maximum is taken over all the roots that compose the De Pril transform such that $\varphi(y)$ behaves like τ^y. For the points $s > c$, $y^-_{s,0}$ and $y^+_{s,0}$ are both approximatively equal to $s - c$ and the values of $\left|\pi^-_{s,0}\right|$ and $\pi^+_{s,0}$ become much greater than $sf(s)$, which creates a lack of precision in $\widetilde{f}(s)$. Therefore, if the De Pril transform converges to zero, the recursive evaluation of the probability function is strongly stable up to the stage c and is unstable from this stage on. Over the unstable range we have actually

$$\widetilde{f}(s) \approx \begin{cases} f(s), & s = c+1, c+2, \ldots, d^\rho \\ \frac{d^\rho}{s}\,\tau^{s-d^\rho}\, f(d^\rho), & s > d^\rho \end{cases}, \tag{4.5.22}$$

where d^ρ is the maximal stage for which all previous evaluations have at least one exact digit in their significand. It can be approximatively determined by

$$d^\rho \approx \inf\{s > c : |\varphi(s-c)|\, f(c)\, \varepsilon > sf(s)\} - 1, \tag{4.5.23}$$

which increases with respect to ρ. This is an approximation since the relative error of $\widetilde{f}(s)$, $s > c$, depends on the relative error of $\widetilde{f}(c)$, which is generally greater than ε. We explain now how (4.5.22) is derived. For the points $s = c+1, c+2, \ldots, d^\rho$, $\widetilde{f}(s)$ evolves similarly to $f(s)$ because the loss of precision in $\pi^+_{s,t} + \pi^-_{s,t}$ is smaller than ρ. However, the

accuracy of $\widetilde{f}(s)$ decreases with respect to s but $\widetilde{f}(d^\rho)$ has at least one exact digit in its significand. For the points $s > d^\rho$, we have from (4.5.2) and (4.5.23) that

$$s\widetilde{f}(s) \approx \pi^+_{s,t}\,\varepsilon \approx \left|\pi^-_{s,t}\,\varepsilon\right| \approx |\varphi(s-c)|\, f(c)\,\varepsilon > sf(s), \quad s > d^\rho, \tag{4.5.24}$$

which leads to

$$(s+1)\widetilde{f}(s+1) \approx |\varphi(s+1-c)|\, f(c)\,\varepsilon \approx \tau\,|\varphi(s-c)|\, f(c)\,\varepsilon \approx \tau s\widetilde{f}(s), \quad s > d^\rho. \tag{4.5.25}$$

Given $f(d^\rho) \approx \widetilde{f}(d^\rho)$, it follows that

$$s\widetilde{f}(s) \;\approx\; \tau^{s-d^\rho} d^\rho \widetilde{f}(d^\rho), \quad s > d^\rho. \tag{4.5.26}$$

which is equivalent to (4.5.22). Therefore, $s\widetilde{f}(s)$ converges to zero at a rate of τ from d^ρ on and the relative error grows exponentially. The main problem is that we do not know the points c and d^ρ before the evaluation. A method to obtain an accurate probability function of a compound binomial distribution over its entire support is developed in Chapter 2. This method, which consists in increasing ρ efficiently, can be generalized for the individual risk model described in Section 4.2.2. We come back to this method in the next subsection. We illustrate these results through a numerical example in Section 4.6.

4.5.3 Stability with a divergent De Pril transform

When the De Pril transform diverges the upper bound of (4.5.9) tends to infinity. This implies that the evaluation may be inaccurate. However, we know that a stable evaluation is obtained at least from a given order defined in (4.5.4). Nevertheless, this order is not the smallest one for which the recursive evaluation is stable as we have seen above.

We consider here the stability of recursive evaluation when the De Pril transform diverges. From Theorem 9 we know that $\varphi(y)$ behaves like τ^y where $\tau \geq 1$ and is defined by (4.5.21). Like in Section 4.5.2 for the case where $t = 0$, we let c_t be the point where $\Gamma^t f$ becomes less increasing than the De Pril transform. This is equivalent to writing that

$$\frac{\Gamma^t f(s+1)}{\Gamma^t f(s)} \;\leq\; \tau, \quad \text{for all } s \geq c_t,\ t \geq 0. \tag{4.5.27}$$

Then, the values of $y^-_{s,t}$ and $y^+_{s,t}$ are both approximatively equal to $s - c_t$ for $s > c_t$ and we can define d^ρ_t as the maximal stage for which all previous evaluations have at least one exact digit in their significand. It can be approximatively determined by

$$d^\rho_t \;=\; \inf\{s > c_t : |\varphi(s-c_t)|\,\Gamma^t f(c_t)\,\varepsilon > s\,\Gamma^t f(s)\} - 1, \quad t \geq 0, \tag{4.5.28}$$

and is an increasing function of ρ. For the same reasons as mentioned above the recursive evaluation of $\Gamma^t f(s)$ is strongly stable up to the stage $s = c_t$ and is unstable from this stage on. For $s > c_t$, we obtain similarly to (4.5.22) that

$$\widetilde{\Gamma^t f}(s) \;\approx\; \begin{cases} \Gamma^t f(s), & s = c_t+1, c_t+2, \ldots, d_t^\rho \\ \frac{d_t^\rho}{s}\,\tau^{s-d_t^\rho}\,\Gamma^t f(d_t^\rho), & s > d_t^\rho \end{cases}, \quad t \geq 0, \tag{4.5.29}$$

Notice that $\widetilde{\Gamma^t f}(s) > \Gamma^t f(s)$ for $s > d_t^\rho$ and $t \geq 0$. This statement can be justified in the same way as in (4.5.24).

We can observe that $\Gamma^{t+1} f(s)$ is generally more increasing with respect to s than $\Gamma^t f(s)$ because we usually have

$$\frac{\Gamma^{t+1} f(s_1)}{\Gamma^{t+1} f(s_2)} \;\leq\; \frac{\Gamma^t f(s_1)}{\Gamma^t f(s_2)}, \quad s_2 \geq s_1 \geq 0,\ t \geq 0. \tag{4.5.30}$$

It follows that c_t and d_t^ρ increase with respect to t while the values of $y_{s,t}^-$ and $y_{s,t}^+$ decrease with respect to t. Thus, the recursive evaluation is stable over a greater interval and becomes more accurate as t increases. Moreover, from (4.5.29) we obtain that

$$\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \;<\; 1, \quad s = c_t+1, c_t+2, \ldots, d_t^\rho,\ t \geq 0, \tag{4.5.31}$$

because there is at least one exact digit in the significand of $\widetilde{\Gamma^t f}(s)$. We also get that

$$\left|\frac{\widetilde{\Gamma^t f}(s) - \Gamma^t f(s)}{\Gamma^t f(s)}\right| \approx \frac{d_t^\rho\,\tau^{s-d_t^\rho}\,\Gamma^t f(d_t^\rho)}{s\,\Gamma^t f(s)} - 1 \approx \frac{d_t^\rho\,\tau^{s-d_t^\rho}\,\Gamma^t f(d_t^\rho)}{s\,\Gamma^t f(s)}, \quad s > d_t^\rho,\ t \geq 0. \tag{4.5.32}$$

Remark that the relative errors in (4.5.31) and (4.5.32) both increase exponentially with respect to s as explained in Section 4.5.2. We can show that (4.5.32) decreases with respect to t, which is not a surprise since the recursive evaluation is more accurate as t increases. We have from (4.5.28) that

$$d_t^\rho\,\Gamma^t f(d_t^\rho) \;\approx\; |\varphi(d_t^\rho - c_t)|\,\Gamma^t f(c_t)\,\varepsilon, \quad t \geq 0. \tag{4.5.33}$$

If (4.5.30) holds, it follows that

$$\frac{d_{t+1}^\rho\,\Gamma^{t+1} f(d_{t+1}^\rho)}{s\,\Gamma^{t+1} f(s)} \approx \frac{\left|\varphi(d_{t+1}^\rho - c_{t+1})\right|\Gamma^{t+1} f(c_{t+1})\,\varepsilon}{s\,\Gamma^{t+1} f(s)} \leq \frac{\left|\varphi(d_{t+1}^\rho - c_{t+1})\right|\Gamma^t f(c_{t+1})\,\varepsilon}{s\,\Gamma^t f(s)}, \tag{4.5.34}$$

for $s > d_{t+1}^\rho$ and $t \geq 0$. Moreover, using (4.5.27) and (4.5.33), we have

$$\begin{aligned} \tau^{s-d_{t+1}^\rho}\left|\varphi(d_{t+1}^\rho - c_{t+1})\right|\Gamma^t f(c_{t+1})\,\varepsilon \;&\leq\; \tau^{s-d_{t+1}^\rho+c_{t+1}-c_t}\left|\varphi(d_{t+1}^\rho - c_{t+1})\right|\Gamma^t f(c_t)\,\varepsilon \\ &\approx\; \tau^{s-d_t^\rho}\,|\varphi(d_t^\rho - c_t)|\,\Gamma^t f(c_t)\,\varepsilon \\ &\approx\; \tau^{s-d_t^\rho}\,d_t^\rho\,\Gamma^t f(d_t^\rho), \quad s > d_{t+1}^\rho,\ t > 0. \end{aligned} \tag{4.5.35}$$

The combination of (4.5.34) and (4.5.35) shows that the relative error of $\widetilde{\Gamma^{t+1}f}(s)$ is smaller than the one of $\widetilde{\Gamma^t f}(s)$ over the range $s > d_{t+1}^{\rho}$. For the points $s = d_t^{\rho}+1, d_t^{\rho}+2, \ldots, d_{t+1}^{\rho}$, the relative error of $\widetilde{\Gamma^{t+1}f}(s)$ is smaller than one which is smaller than the relative error of $\widetilde{\Gamma^t f}(s)$. This proves that the accuracy of $\widetilde{\Gamma^t f}$ becomes better as t increases.

Hence, there exists an order u_c for which the recursive evaluation of $\Gamma^t f$ is strongly stable up to the point z for $t \geq u_c$. There also exists an order u_d^{ρ} for which the recursive evaluation of $\Gamma^t f$ using a precision ρ leads to values with at least one exact digit in their significand up to the point z for $t \geq u_d^{\rho}$. These orders are determined by

$$u_c = \inf\{t \geq 0 : c_t \geq z\}, \quad z \geq 1, \tag{4.5.36}$$

and

$$u_d^{\rho} = \inf\{t \geq 0 : d_t^{\rho} \geq z\}, \quad z \geq 1, \tag{4.5.37}$$

which are much smaller than the order defined in (4.5.4). However, we do not know the values of c_t and d_t^{ρ}, $t \geq 0$, in advance and if we want to evaluate $\Gamma^t f(z)$ with $t < u_d^{\rho}$ we lose some accuracy in taking finite differences as explained above. The relative error obtained for that value is of the same magnitude as (4.5.32). Therefore, the most efficient approach consists in the calculation of $\Gamma^t f$ using a recursive evaluation for the order t and augmenting ρ in order to increase d_t^{ρ} and to obtain an accurate value of $\Gamma^t f(z)$.

The question is now to determine the smallest precision that we can use in order to obtain a value for $\Gamma^t f(z)$, $z \geq 1$ and $t \geq 0$, with at least one exact digit in its significand. We denote this precision by $\hat{\rho}_{z,t}$. Remark that if we desire an accuracy of at least v decimal digits in $\widetilde{\Gamma^t f}(z)$ we only have to add $\log_{\beta} 10^v$ to $\hat{\rho}_{z,t}$. The evaluation of $\widetilde{\Gamma^t f}(s)$ for $s = 0, 1, \ldots, z$ using this precision will then have an accuracy of at least v digits since (4.5.31) and (4.5.32) increase with respect to s. Similarly as in Chapter 2, we can consider the base-β logarithm of the relative error at a given point as a lack of precision of the floating-point representation in order to determine $\hat{\rho}_{z,t}$.

Under the individual risk model, a first method is to evaluate $\Gamma^t f$ using a reasonable precision ρ up to the final value ξ and then to evaluate it again using a precision of

$$\hat{\rho}_{\xi,t} = \rho + \log_{\beta} \left| \frac{\widetilde{\Gamma^t f}(\xi) - \Gamma^t f(\xi)}{\Gamma^t f(\xi)} \right|, \quad t \geq 0, \tag{4.5.38}$$

where $\Gamma^t f(\xi)$ is determined by Theorem 7. It is assumed here that the evaluation of $\widetilde{\Gamma^t f}(\xi)$ with a precision of ρ is inaccurate. The second evaluation using the precision $\hat{\rho}_{\xi,t}$ gives accurate values for $\Gamma^t f$ over the entire support of S.

We develop now another method to determine $\hat{\rho}_{z,t}$ which may involve a smaller number of operations during the first evaluation. If we consider the base-β logarithm of the relative error at z, which is obtained in an evaluation using a precision ρ, as a lack of precision, it follows from (4.5.32) that

$$\hat{\rho}_{z,t} \approx \rho + \log_\beta \frac{d_t^\rho \, \tau^{z-d_t^\rho} \, \Gamma^t f(d_t^\rho)}{z \, \Gamma^t f(z)}, \quad z > d_t^\rho, \tag{4.5.39}$$

which holds for any ρ for which $\widetilde{\Gamma^t f}(z)$ is inaccurate. Notice that the precision in (4.5.39) cannot be determined before an evaluation since it depends on exact values and on d_t^ρ that are unknown. However, we get from (4.5.29) and (4.5.33) that

$$z\widetilde{\Gamma^t f}(z) \approx d_t^\rho \, \tau^{z-d_t^\rho} \, \Gamma^t f(d_t^\rho) \approx |\varphi(d_t^\rho - c_t)| \, \tau^{z-d_t^\rho} \, \Gamma^t f(c_t) \, \varepsilon \approx |\varphi(z-c_t)| \, \Gamma^t f(c_t) \, \varepsilon, \tag{4.5.40}$$

for $t \geq 0$, which leads to

$$\hat{\rho}_{z,t} \approx 1 + \log_\beta \frac{|\varphi(z-c_t)| \, \Gamma^t f(c_t)}{2z \, \Gamma^t f(z)}, \quad z > c_t, \; t \geq 0, \tag{4.5.41}$$

which no more depends on any other precision but still depends on exact values and on c_t. From (4.5.41), we are able to calculate $\hat{\rho}_{\xi,t}$ which is an upper bound of $\hat{\rho}_{z,t}$ for $z = c_t+1, c_t+2 \ldots, \xi$, if we are given $\Gamma^t f(\xi)$ and after a first evaluation using a reasonable precision up to the point c_t. This point can be determined using (4.5.27). This is possible under the individual risk model because Theorem 7 gives the final value and since the first evaluation gives an accurate value of $\Gamma^t f(c_t)$ whichever precision we use. The recursive evaluation of $\Gamma^t f$ using a precision of $\hat{\rho}_{\xi,t}$ is accurate over the whole support of S. However, we cannot determine exactly $\hat{\rho}_{z,t}$ and using a precision $\hat{\rho}_{\xi,t}$ in the evaluation of $\Gamma^t f(z)$ may consume too much time if the difference between these two precisions is significantly large.

Remark 12: The precision given by (4.5.39) or (4.5.41) corresponds to the precision for which $|\varphi(z-c_t)| \, \Gamma^t f(c_t) \, \hat{\varepsilon}_{z,t} = z\Gamma^t f(z)$, where $\hat{\varepsilon}_{z,t}$ is the maximal relative error occurring in the floating-point representation with a precision of $\hat{\rho}_{z,t}$. It is similar to write that $\hat{\rho}_{z,t}$ is the value of ρ that satisfies $z = d_t^\rho$, $t \geq 0$.

Remark 13: The recursive evaluation using (4.2.5) with a precision obtained by (4.5.41) with $z = \xi$ is accurate over the range $[0, \xi]$ for any random variable with De Pril transform φ if we know $\Gamma^t f(\xi)$ and if $\log \Gamma^t f$ is roughly concave. The latter condition is generally fulfilled for any random variable.

Remark 14: If the claim amount random variables of all policies are identically distributed, the precision $\hat{\rho}_{\xi,0}$ given by (4.5.38) or (4.5.41) with $\beta = 2$ and Theorem 7 corresponds to the needed precision discussed in Chapter 2.

Remark 15: Both methods developed above also hold for the recursive evaluation of a probability function when its De Pril transform converges absolutely to zero.

We discuss now a method to determine a precision that guarantees accurate values up to the point z. It can be used for any random variable with a divergent De Pril transform and without any other information else than the De Pril transform. Under the individual risk model, this method, which can be used only for $t \geq 1$, is more adapted if we want to evaluate only the first z values of $\Gamma^t f$. It follows from (4.5.41) that

$$\hat{\rho}_{z,t} \leq 1 + \log_\beta \frac{|\varphi(z - c_t)|}{2z}, \quad z > c_t,\ t \geq 1, \tag{4.5.42}$$

which does not depend on any exact value. Therefore, it is sufficient to represent exactly the integer part of $\varphi(z-c_t)$ to obtain an accurate value of $\Gamma^t f(z)$, $z > c_t$. Nevertheless, we still need a first evaluation in order to determine the point c_t. However, since φ diverges we have

$$\hat{\rho}_{z,t} \leq 1 + \log_\beta \frac{|\varphi(z)|}{2z}, \quad z \geq 1,\ t \geq 1, \tag{4.5.43}$$

which does no more depend on c_t and on t. Under the individual model, when y becomes large the De Pril transform is given by

$$|\varphi(y)| \approx n_{ij}^{\tau} \tau^y, \tag{4.5.44}$$

where n_{ij}^{τ} is the number of policies in the class where the largest root appears. Thus, it follows that

$$\hat{\rho}_{z,t} \leq 1 + z \log_\beta \tau + \log_\beta \frac{n_{ij}^{\tau}}{2z}, \quad z \geq 1,\ t \geq 1. \tag{4.5.45}$$

Remark 16: To be prudent we should define a reasonable minimal precision that should be used in evaluations when we calculate a sufficient precision using (4.5.41), (4.5.42), (4.5.43) or (4.5.45).

Remark 17: The truncation of a divergent De Pril transform is not possible. Therefore, in such a case the most efficient strategy consists in the evaluation of $\Gamma^t f$ according to Theorem 8 even if the precision that is necessary depends on the De Pril transform that we can approximate by (4.5.44).

Remark 18: We can derive a recursive formula for the t-th order tail probability function, $\Lambda^t f$, defined in (4.2.16), from the substitution of the terms $\lambda x f(x)$ by $\varphi(x)$ into Lemma 3.2 of Antzoulakos and Chadjiconstantinidis (2004) that holds for compound Poisson distributions. However, such a recursive evaluation requires a greater number of operations than the one of $\Gamma^t f$. Moreover, for $t \geq 1$, $\Lambda^t f(s)$ decreases with respect to s and is thus unstable for any kind of distributions. In addition, we need a large precision in order to obtain accurate results because the exact values become smaller and smaller as we go along with the recursion. The same problems arise for a backward recursive formula for $\Gamma^t f$ that can be derived from (4.2.8) in the case where a truncation of the De Pril transform is effective. An idea to obtain a stable evaluation for $\Lambda^t f$ could be to derive a backward recursive evaluation when the De Pril transform can be truncated. However, in this case such functions can be obtained by (4.2.15) from a stable evaluation of $\Gamma^t f$.

4.6 Numerical illustrations

In this section we illustrate the results of Sections 4.5.2 and 4.5.3. For the first illustration we consider an example with a convergent De Pril transform. We consider an adaptation of the example of Gerber (1979) where we have a portfolio of life insurance policies that are grouped according to Table 4.1 and where the individual claim amount distribution is

Table 4.1: Number of policies in each class

j	q_j	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
1	0.03	20	30	10	20	-
2	0.04	-	10	20	20	10
3	0.05	-	20	40	20	20
4	0.06	-	20	20	20	10

$$g_i(x) = \begin{cases} 1, \text{ if } x = i \\ 0, \text{ if } x \neq i \end{cases}, \quad x \geq 1, \ i = 1, \ldots, 5. \tag{4.6.1}$$

We have $\xi = 970$ for this portfolio. For each class (i, j), we have $|\lambda_{ijk}| = \left(\frac{q_j}{p_j}\right)^{\frac{1}{i}}$, for $k = 1, \ldots, i$, which leads to $\tau = \left(\frac{q_4}{p_4}\right)^{\frac{1}{5}} = 0.57677$.

We evaluate the probability function according to Theorem 8 using ten decimal digits in the representation of real numbers by floating-point numbers i.e. $\beta = 10$ and $\rho = 10$. It turns out that $c = 260$ and $d^\rho = 445$ when they are determined by (4.5.20) and (4.5.23), respectively. Table 4.2 gives the exact values and the evaluations using the

Table 4.2: Values of $f(s)$ and $\widetilde{f}(s)$ with relative errors for some points

s	$f(s)$	$\widetilde{f}(s)$	$\left\|\dfrac{\widetilde{f}(s) - f(s)}{f(s)}\right\|$
c	$2.9435 \cdot 10^{-34}$	$2.9435 \cdot 10^{-34}$	$1.6987 \cdot 10^{-9}$
d^ρ	$8.8074 \cdot 10^{-89}$	$8.6401 \cdot 10^{-89}$	0.0190
ξ	$4.5802 \cdot 10^{-422}$	$2.7667 \cdot 10^{-217}$	$6.0405 \cdot 10^{204}$

floating-point representation with their relative error for some points of the probability function. Figure 4.2 illustrates the fact that $s\widetilde{f}(s)$ converges to zero at the rate τ from the

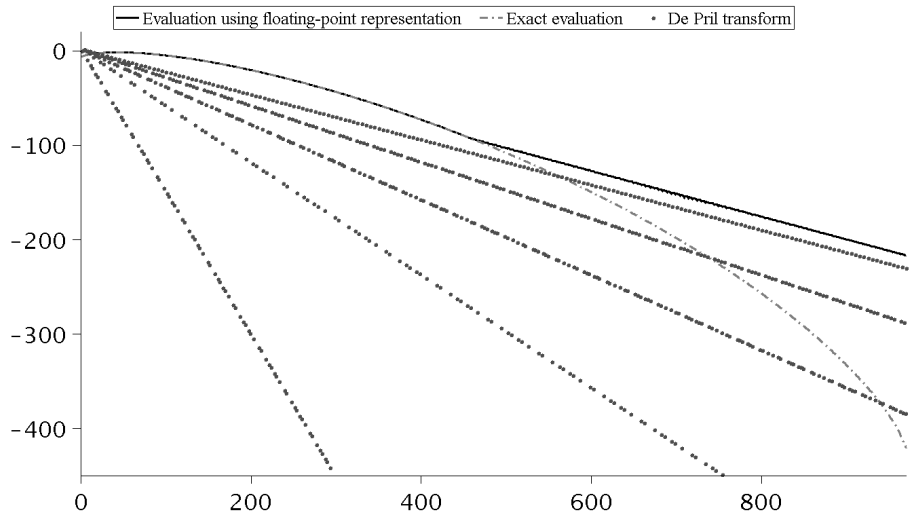


Figure 4.2: $\log_{10} \widetilde{f}(s)$, $\log_{10} f(s)$ and $\log_{10} |\varphi|$

point $d^\rho = 445$ on. It shows the logarithm of the recursive evaluation using the floating-point representation, the logarithm of f and the logarithm of the De Pril transform. We observe that the logarithm of the De Pril transform is composed by five lines. This is not surprising since here $\sum_{k=1}^{\omega_i} \lambda_{ijk}^y = 0$ if y is not a multiple of i. However, the upper line behaves similarly to $s \log_{10} \tau$ and is parallel to $\log_{10} \widetilde{f}(s)$ for $s \geq d^\rho$. The fact that these curves are parallel shows that $s\widetilde{f}(s)$ converges to zero at the same rate as τ^s for

$s \geq d^{\rho}$. Finally, the evolution of the relative error as we go along with the recursion using the floating-point representation is displayed in Figure 4.3. Notice that the recursive

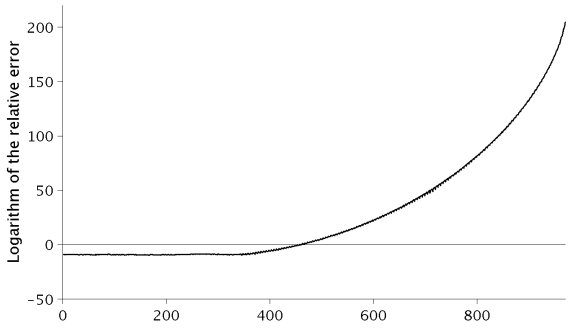


Figure 4.3: Logarithm of the relative error resulting from an evaluation using the floating-point representation

evaluation of $\Gamma^t f$ for $t \geq 1$ is stable and that the implicit truncation point defined in (4.5.14) is $r_t = 45$ for $t \in \{1, 2\}$.

For the second illustration we consider the recursive evaluation of the t-th order cumulative distribution function of a compound binomial distribution with the same parametrization as in Section 4.4.2. The probability function g is chosen to be the same as Example 8 of Panjer and Wang (1993, p. 249). It is given in Table 4.3. We set $m = 100$ and $p = 0.09$

Table 4.3: Probability function of the X_i's

x	1	2	3	4	5	6	7	8	9	10
$g(x)$	0.150	0.200	0.250	0.125	0.075	0.050	0.050	0.050	0.025	0.025

and we use the floating-point representation with $\beta = 10$ and $\rho = 10$. Thus, we have $\xi = 1000$ and $\tau = 1.17234$. Table 4.4 gives, for some orders, the values of c_t and d_t^{ρ} that

Table 4.4: Values of c_t and d_t^{ρ} for some orders

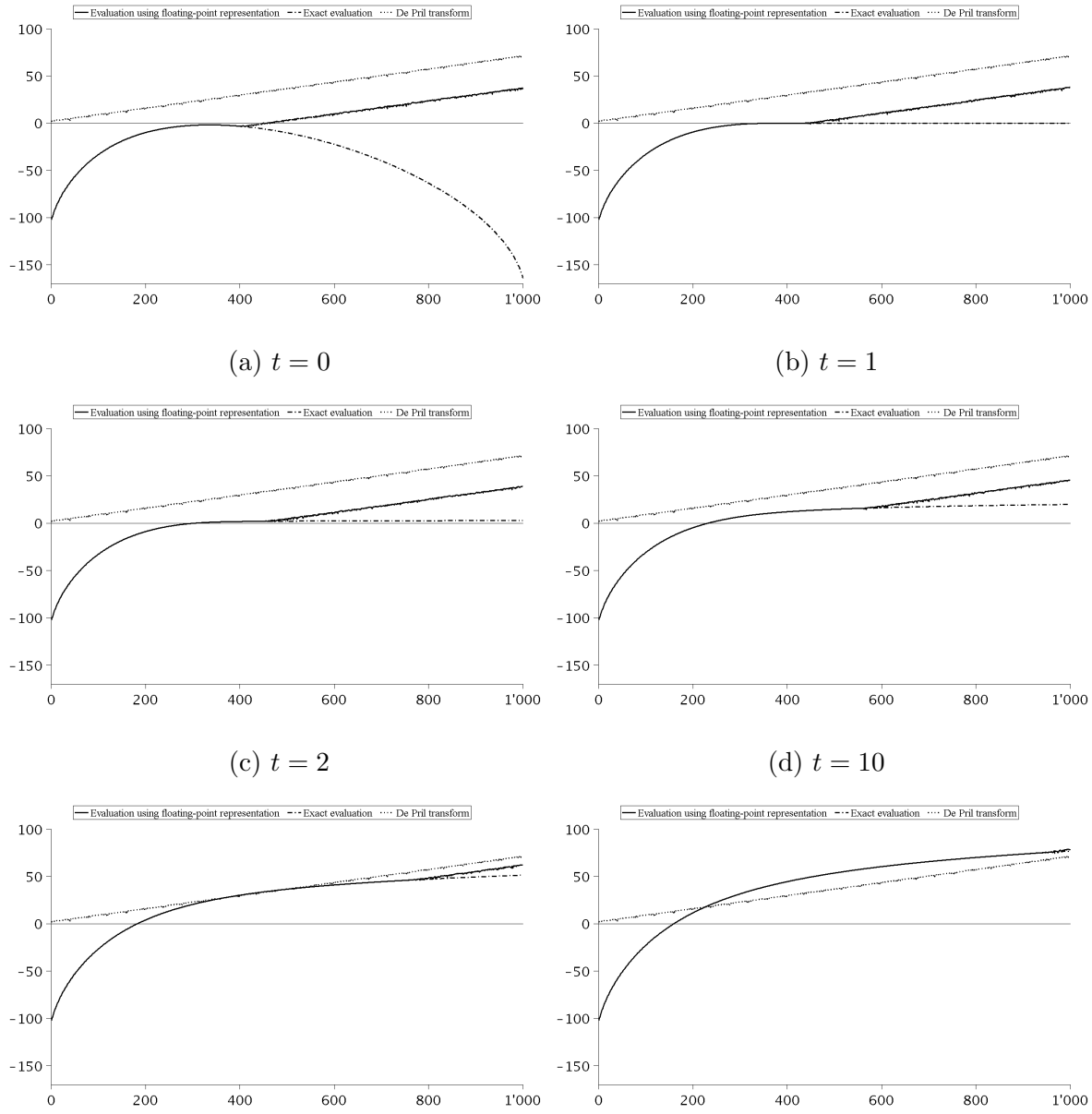
t	0	1	2	10	30	50
c_t	255	261	266	311	426	541
d_t^{ρ}	408	447	466	575	780	965

are determined by (4.5.27) and (4.5.28), respectively. We remark that both values increase

Table 4.5: Values of $\Gamma^t f(\xi)$ and $\widetilde{\Gamma^t f}(\xi)$ for some orders, approximations of $\widetilde{\Gamma^t f}(\xi)$ and relative errors at ξ

t	$\Gamma^t f(\xi)$	$\widetilde{\Gamma^t f}(\xi)$	$\frac{\lvert\varphi(\xi - c_t)\rvert\, \Gamma^t f(c_t)\, \varepsilon}{\xi}$	$\left\lvert \frac{\widetilde{\Gamma^t f}(\xi) - \Gamma^t f(\xi)}{\Gamma^t f(\xi)} \right\rvert$
0	$4.9905 \cdot 10^{-165}$	$2.0062 \cdot 10^{36}$	$1.0128 \cdot 10^{37}$	$4.0200 \cdot 10^{200}$
1	1	$1.3152 \cdot 10^{38}$	$5.8972 \cdot 10^{37}$	$1.3152 \cdot 10^{38}$
2	664.3	$1.0647 \cdot 10^{39}$	$3.1557 \cdot 10^{38}$	$1.6027 \cdot 10^{36}$
10	$7.6841 \cdot 10^{19}$	$2.0061 \cdot 10^{45}$	$1.3621 \cdot 10^{45}$	$2.6108 \cdot 10^{25}$
30	$2.3990 \cdot 10^{51}$	$1.6752 \cdot 10^{62}$	$3.8397 \cdot 10^{61}$	$6.9830 \cdot 10^{10}$
50	$7.0414 \cdot 10^{76}$	$-5.2747 \cdot 10^{78}$	$1.1894 \cdot 10^{78}$	75.909

with respect to t. Table 4.5 contains the values of $\Gamma^t f(\xi)$ and $\widetilde{\Gamma^t f}(\xi)$ with their relative error for some orders as well as the approximation of $\widetilde{\Gamma^t f}(\xi)$ that follows from (4.5.40) for each order. It shows that the relative error decreases with respect to t and that the approximation $\frac{|\varphi(\xi - c_t)|\Gamma^t f(c_t)\,\varepsilon}{\xi}$ is close to $\widetilde{\Gamma^t f}(\xi)$. Moreover, the orders defined by (4.5.36) and (4.5.37) that guarantee a stable or an accurate evaluation up to ξ are $u_c = 130$ and $u_d^\rho = 55$, respectively. Figure 4.4 shows the logarithm of the evaluation using the floating-point representation, the logarithm of the exact evaluation and the logarithm of the De Pril transform for some orders. We observe that $s\widetilde{\Gamma^t f}(s)$ grows at a rate of $\tau = 1.17234$ from the point d_t^ρ on and for each t, since $\log_{10} \widetilde{\Gamma^t f}(s)$ is parallel to $s \log_{10} \tau$ for $s > d_t^\rho$. We can also remark that d_t^ρ increases with respect to t, which means that the recursive evaluation using floating-point representation is more accurate as t increases. This statement can also be observed in Figure 4.5 where we display the relative error of $\widetilde{\Gamma^t f}$ as we go along with the recursion for the same orders as in Figure 4.4. Table 4.6 shows the values of the precisions and their upper bounds discussed in Section 4.5.3. These precisions guarantee an accurate evaluation over the entire support of S. Remark that we have $n_{ij}^\tau = m$ for the evaluation of (4.5.45). Table 4.7 contains the same precisions and their upper bounds but when we desire an accurate evaluation up to the point 600 only. Notice that the third column of Table 4.7 cannot be determined from a first evaluation since it depends on the exact value of $\Gamma^t f(600)$. We remark that the values of $\hat{\rho}_{z,t}$ obtained from (4.5.41) for $t = 30$ and $t = 50$ are smaller that ρ. This is not surprising since $d_t^\rho > 600$ for both cases. We also observe that the upper bound for the needed precision computed by (4.5.42) is smaller than $\hat{\rho}_{\xi,t}$ for each case where it can be evaluated with $z = 600$.



(a) $t = 0$ (b) $t = 1$

(c) $t = 2$ (d) $t = 10$

(e) $t = 30$ (f) $t = 50$

Figure 4.4: $\log_{10} \widetilde{\Gamma^t f}(s)$, $\log_{10} \Gamma^t f(s)$ and $\log_{10} |\varphi|$

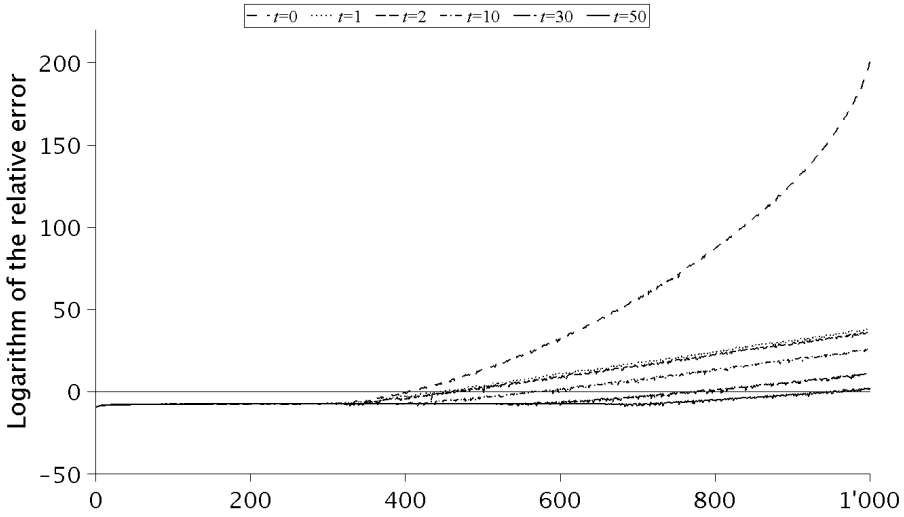


Figure 4.5: Logarithm of the relative error occurring from an evaluation using the floating-point representation for some orders

Table 4.6: Values of $\hat{\rho}_{\xi,t}$ and their upper bounds that guarantee an accurate evaluation up to ξ

t	$\hat{\rho}_{\xi,t}$	From (4.5.41)	From (4.5.42)	From (4.5.43)	From (4.5.45)
0	210.60	211.31	-	-	-
1	48.12	47.77	50.96	68.59	68.75
2	46.20	45.68	50.52	68.59	68.75
10	35.42	35.25	47.50	68.59	68.75
30	20.84	20.20	39.50	68.59	68.75
50	11.88	11.23	31.48	68.59	68.75

Table 4.7: Approximations for $\hat{\rho}_{z,t}$ and their upper bounds that guarantee an accurate evaluation for the first 600 points

t	$\hat{\rho}_{z,t}$	From (4.5.42)	From (4.5.43)	From (4.5.45)
0	42.02	-	-	-
1	20.33	23.51	40.97	41.35
2	18.73	23.18	40.97	41.35
10	11.35	20.13	40.97	41.35
30	3.21	12.15	40.97	41.35
50	0.37	4.15	40.97	41.35

4.7 Conclusion

We have seen that the convergence of divergence rate of the De Pril transform has a large effect on the stability against round-off errors of the recursive evaluation of t-th order cumulative distribution functions. In particular, the evaluation of such a function with $t \geq 1$ is strongly stable if the De Pril transform converges to zero. We also obtain that the only way to get an accurate evaluation of a given order is to increase the precision of the floating-point representation. An upper bound for the precision that is necessary in order to guarantee an accurate evaluation can be determined from the De Pril transform without doing a first evaluation with a small precision.

Bibliography

Adelson, R. M. (1966). Compound Poisson distributions. *Operations Research Quarterly 17*, 73–75.

Ambagaspitiya, R. S. (1995). A family of discrete distributions. *Insurance: Mathematics and Economics 16*, 107 – 127.

Antzoulakos, D. L. and S. Chadjiconstantinidis (2004). On mixed and compound mixed Poisson distributions. *Scandinavian Actuarial Journal*, 161–188.

Beall, G. and R. R. Rescia (1953). A generalization of Neyman's contagious distributions. *Biometrics 9*, 354–386.

Bühlmann, H. (1984). Numerical evaluation of the compound Poisson distribution: Recursion or fast Fourier transform. *Scandinavian Actuarial Journal*, 116–126.

Carver, H. C. (1919). On the graduation of frequency distributions. *Proceedings of the Casualty Actuarial and Statistical Society 6*, 52–72.

Cash, J. R. (1980). *Stable Recursions: with Applications to the Numerical Solution of Stiff Systems*. London: Academic Press.

Cauchy, A. L. (1829). *Exercices de mathématiques*. Quatrième année. Paris.

Chadjiconstantinidis, S. and G. Pitselis (2009). Further improved recursions for a class of compound Poisson distributions. *Insurance: Mathematics and Economics 44*, 278 – 286.

Chan, B. (1982a). Recursive formulas for aggregate claims. *Scandinavian Actuarial Journal*, 38–40.

Chan, B. (1982b). Recursive formulas for discrete distributions. *Insurance: Mathematics and Economics 1*, 241–243.

De Pril, N. (1985). Recursions for convolutions of arithmetic distributions. *ASTIN Bulletin 15*, 135–139.

De Pril, N. (1986a). Improved recursions for some compound distributions. *Insurance: Mathematics and Economics 5*, 129–132.

De Pril, N. (1986b). On the exact computation of the aggregate claims distribution in the individual life model. *ASTIN Bulletin 16*, 109–112.

De Pril, N. (1988). Improved approximations for the aggregate claims distribution of a life insurance portfolio. *Scandinavian Actuarial Journal*, 61–68.

De Pril, N. (1989). The aggregate claims distribution in the individual model with arbitrary positive claims. *ASTIN Bulletin 19*, 9–24.

De Pril, N. and J. Dhaene (1992). Error bounds for compound Poisson approximations of the individual risk model. *ASTIN Bulletin 22*, 135–148.

Dhaene, J. and N. De Pril (1994). On a class of approximative computation methods in the individual risk model. *Insurance: Mathematics and Economics 14*, 181–196.

Dhaene, J., C. Ribas, and R. Vernic (2006). Recursions for the individual risk model. *Acta Mathematicae Applicatae Sinica, English Series 22*, 543–564.

Dhaene, J. and B. Sundt (1997). On error bounds for approximations to aggregate claims distributions. *ASTIN Bulletin 27*, 243–262.

Dhaene, J. and B. Sundt (1998). On approximating distributions by approximating their De Pril transforms. *Scandinavian Actuarial Journal*, 1–23.

Dhaene, J. and M. Vandebroek (1995). Recursions for the individual model. *Insurance: Mathematics and Economics 16*, 31–38.

Dhaene, J., G. E. Willmot, and B. Sundt (1999). Recursions for distribution functions and stop-loss transforms. *Scandinavian Actuarial Journal*, 52–65.

Eisele, K.-T. (2006). Recursions for compound phase distributions. *Insurance: Mathematics and Economics 38*, 149–156.

Eisele, K.-T. (2008). Recursions for multivariate compound phase variables. *Insurance: Mathematics and Economics 42*, 65–72.

Embrechts, P. and M. Frei (2009). Panjer recursion versus FFT for compound distributions. *Mathematical Methods of Operations Research 69*, 497–508.

Euler, L. (1748). *Introductio in analysin infinitorum*. Lausanne: Marc-Michel Bousquet et Cie.

Euler, L. (1751). Observationes analyticae variae de combinationibus. *Commentarii academiae scientiarum Petropolitanae 13*, 64–93.

Euler, L. (1755). *Institutiones calculi differentialis cum eius usu in analysi finitorum ac doctrina serierum*. St. Petersburg: Academiae scientiarum Petropolitanae.

Gathy, M. and C. Lefèvre (2010). On the Lagrangian Katz family of distributions as a claim frequency model. *Insurance: Mathematics and Economics 47*, 76 – 83.

Gautschi, W. (1967). Computational aspects of three-term recurrence relations. *SIAM Review 9*, 24–82.

Gerber, H. U. (1979). *An Introduction to Mathematical Risk Theory*. Philadelphia: S.S. Heubner Foundation Monograph Series 8.

Gerber, H. U. (1984). Error bounds for the compound Poisson approximation. *Insurance: Mathematics and Economics 3*, 191–194.

Gerhold, S., U. Schmock, and R. Warnung (2010). A generalization of Panjer's recursion and numerically stable risk aggregation. *Finance and Stochastics 14*, 81–128.

Granlund, T. (2007, August). *GNU MP: The GNU Multiple Precision Arithmetic Library* (4.2.2 ed.).

Grübel, R. and R. Hermesmeier (1999). Computation of compound distributions I: Aliasing errors and exponential tilting. *ASTIN Bulletin 29*, 197–214.

Hess, K., A. Liewald, and K. Schmidt (2002). An extension of Panjer's recursion. *ASTIN Bulletin 32*, 283–297.

Hesselager, O. (1994). A recursive procedure for calculation of some compound distributions. *ASTIN Bulletin 24*, 19–32.

Hesselager, O. (1996). Recursions for certain bivariate counting distributions and their compound distributions. *ASTIN Bulletin 26*, 35–52.

Hipp, C. (1985). Approximation of aggregate claims distributions by compound Poisson distributions. *Insurance: Mathematics and Economics 4*, 227–232.

Hipp, C. (1986). Improved approximations for the aggregate claims distribution in the individual model. *ASTIN Bulletin 16*, 89–100.

Hipp, C. (2006). Speedy convolution algorithms and Panjer recursions for phase-type distributions. *Insurance: Mathematics and Economics 38*, 176–188.

Johnson, N. L., A. W. Kemp, and S. Kotz (2005). *Univariate Discrete Distributions* (Third ed.). Wiley Series in Probability and Statistics. Hoboken, NJ: John Wiley & Sons.

Kaas, R., M. Goovaerts, J. Dhaene, and M. Denuit (2008). *Modern Actuarial Risk Theory – Using R*. Berlin Heidelberg: Springer-Verlag.

Katz, L. (1945). *Characteristics of frequency functions defined by first-order difference equations*. Dissertation, University of Michigan, Ann Arbor, MI.

Katz, L. (1965). Unified treatment of a broad class of discrete probability distributions. In G. P. Patil (Ed.), *Classical and Contagious Discrete Distributions*, Oxford, pp. 64–93. Pergamon Press.

Khatri, C. G. and I. R. Patel (1961). Three classes of univariate discrete distributions. *Biometrics 17*, 567–575.

Knuth, D. (1992). Convolution polynomials. *The Mathematica Journal 2*, 67–78.

Kornya, P. S. (1983). Distribution of the aggregate claims in the individual risk theory model. *Transactions of Society of Actuaries 35*, 823–836.

Neyman, J. (1939). On a new class of "contagious" distributions, applicable in entomology and bacteriology. *The Annals of Mathematical Statistics 10*, 35–57.

Oliver, J. (1967). Relative error propagation in the recursive solution of linear recurrence relations. *Numerische Mathematik 9*, 323–340.

Olver, F. W. J. (1964). Error analysis of Miller's recurrence algorithm. *Mathematics of Computation 18*, 65–74.

Panjer, H. H. (1980). The aggregate claims distribution and stop-loss reinsurance. *Transactions of the Society of Actuaries 32*, 523–545.

Panjer, H. H. (1981). Recursive evaluation of a family of compound distributions. *ASTIN Bulletin 12*, 22–26.

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

Panjer, H. H. and S. Wang (1995). Computational aspects of Sundt's generalized class. *ASTIN Bulletin 25*, 5–17.

Panjer, H. H. and G. E. Willmot (1982). Recursions for compound distributions. *ASTIN Bulletin 13*, 1–11.

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

Plackett, R. L. (1969). Stochastic models of capital investment. *Journal of the Royal Statistical Society. Series B (Methodological) 11*, 1–28.

Schröter, K. J. (1990). Recursions for compound distributions. *Scandinavian Actuarial Journal*, 161–175.

Seal, H. L. (1971). Numerical calculation of the Bohman-Escher family convolution-mixed negative binomial distribution functions. *Mitteilungen der Vereinigung Schweizerischer Versicherungsmathematiker 71*, 71–94.

Shiu, E. S. W. (1983). Discussion on Kornya's paper: Distribution of aggregate claims in the individual risk theory model. *Transactions of Society of Actuaries 35*, 847–850.

Stroh, E. (1978). Actuarial note: The distribution functions of collective risk theory as linear compounds. *Actuarial Research Clearing House*.

Ströter, B. (1985). The numerical evaluation of the aggregate claim density function via integral equations. *Blatter der Deutschen Gesellschaft fur Versicherungs-mathematik 17*, 1–14.

Sundt, B. (1992). On some extensions of Panjer's class of counting distributions. *ASTIN Bulletin 22*, 61–80.

Sundt, B. (1995). On some properties of De Pril transforms of counting distributions. *ASTIN Bulletin 25*, 19–31.

Sundt, B. (1998). A generalization of the De Pril transform. *Scandinavian Actuarial Journal*, 41–48.

Sundt, B. (1999a). On multivariate Panjer recursions. *ASTIN Bulletin 29*, 29–45.

Sundt, B. (1999b). Recursions for convolutions of discrete uniform distributions revisited. *Insurance: Mathematics and Economics 24*, 15–21.

Sundt, B. (2000). On multivariate Vernic recursions. *ASTIN Bulletin 30*, 111–122.

Sundt, B. (2002). Recursive evaluation of aggregate claims distributions. *Insurance: Mathematics and Economics 30*, 297–322.

Sundt, B., J. Dhaene, and N. De Pril (1998). Some results on moments and cumulants. *Scandinavian Actuarial Journal*, 24–40.

Sundt, B. and O. Ekuma (1999). The De Pril transform of a compound R_k distribution. *Bulletin of the Swiss Association of Actuaries*, 179–189.

Sundt, B. and W. S. Jewell (1981). Further results on recursive evaluation of compound distributions. *ASTIN Bulletin 12*, 27–39.

Sundt, B. and R. Vernic (2009). *Recursions for Convolutions and Compound Distributions with Insurance Applications*. EAA Lecture Notes. Berlin Heidelberg: Springer-Verlag.

Tilley, J. A. (1980). Discussion on the aggregate claims distribution and stop-loss reinsurance. *Transactions of the Society of Actuaries 32*, 538–544.

Vernic, R. (1999). Recursive evaluation of some bivariate compound distributions. *ASTIN Bulletin 29*, 315–325.

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

White, R. P. and T. N. E. Greville (1959). On computing the probability that exactly k out of n independent events will occur. *Transactions of Society of Actuaries 11*, 88–99.

Williams, R. E. (1980). Computing the probability density function for aggregate claims. *Proceedings of the Canadian Institute of Actuaries 11*, 39–47.

Zahar, R. V. M. (1977). A mathematical analysis of Miller's algorithm. *Numerische Mathematik 27*, 427–447.