

Computing compound distributions faster!

P.W. den Iseger, M.A.J. Smith, R. Dekker *

Econometrics and Operations Research, Erasmus University Rotterdam, PO Box 1738, 3000 DR Rotterdam, Netherlands

Received May 1996; revised November 1996

Abstract

The use of Panjer's algorithm has meanwhile become a widespread standard technique for actuaries (Kuon et al., 1955). Panjer's recursion formula is used for the evaluation of compound distributions and can be applied to life and general insurance problems. The discrete version of Panjer's recursion formula is often applied to continuous distributions by discretizing the underlying distribution at n equidistant points, covering a large enough interval, say $[0, n\Delta]$. Panjer's recursion returns a discrete function as an approximation of the compound distribution. It is claimed that this procedure is fast, ($O(n^2)$), accurate, ($O(\Delta^2)$) and easy to understand, cf. Bühlmann (1984), Dickson (1995) and Xie (1989). In this article we propose a method based on cubic splines. The accuracy of the method is better, namely $O(\Delta^5)$. The computation time is $O(n^2)$ and hence for the same accuracy *much faster* and furthermore, the method returns a twice continuously differentiable function.

Keywords: Recursions; Compound distributions

1. Introduction

The need for a fast and accurate method to compute compound distributions is self-evident. In actuarial literature numerous papers have appeared that pay attention to this problem. Undoubtedly the most celebrated method to compute compound distributions is Panjer's discrete recursion formula. In this article we focus on continuous underlying distributions. By discretizing the continuous distribution at n equidistant points covering a large enough interval, say $[0, n\Delta]$, and using Panjer's discrete recursion formula an approximation is obtained for the compound distribution.

Since the original 1981 paper of Panjer considerable attention has been paid to Panjer's recursion and recently Dickson (1995) has published an overview on Panjer's recursion formula for evaluation of compound distributions and how it can be applied to life and general insurance problems. Two classical actuarial problems are: the calculation of the distribution of aggregate claims from a portfolio and the calculation of the probability of ultimate ruin in the classical risk model. In Dickson's article the computation speed and accuracy of Panjer's method is praised as an important factor for its success. The accuracy and computation time of the method is controlled by the discretization of the underlying distribution. More points yield a higher accuracy, $O(\Delta^2)$, and also more computation time, $O(n^2)$. The method that we present has four advantages to Panjer's discrete recursion scheme:

* Corresponding author.

- (1) In the points $i\Delta$, $i = 0 \dots n$, the accuracy is of order $O(\Delta^5)$.
- (2) The function and the first k derivatives, $k = 1, \dots, 3$, converge uniformly with an accuracy of order $O(\Delta^{4-k})$.
- (3) The computation time is of order $O(n^2)$. Hence, since the accuracy is much better than Panjer's recursion, more accurate answers can be obtained in less time.
- (4) The method returns a twice continuously differentiable function. This has as the advantage that all values in the interval $[0, n\Delta]$ are approximated. Furthermore, since also the first two derivatives are approximated, the approximation is suitable for use in optimization algorithms.

The paper is organized as follows. In Section 2 we will briefly explain the use of Panjer's discrete recursion formula to continuous underlying distributions. In Section 3 we will explain the conceptual idea of our method and in Section 4 the method will be presented in detail. In Section 5 we give some theoretical results of the method. In Section 6 we propose an improvement of the method in case the individual claim amount distribution is not analytic in zero. In Section 7 we give two numerical examples to corroborate the theoretical results.

Notation

$F(t)$	the interarrival distribution of claims
$f(t)$	the pdf of $F(t)$, i.e. $f(t) = dF(t)/dt$
$p_m(t)$	the probability of m claims in the interval $[0, t]$, i.e. $p_m(t) = F^{(m)}(t) - F^{(m+1)}(t)$, where $F^{(m)}(t)$ denotes the m -fold convolution of $F(t)$
$G(x)$	the distribution of the individual claim amount (we assume without loss of generality that $G(0) = 0$)
$g(x)$	in case the individual claim amount distribution is continuously differentiable, we denote $g(x) = dG(x)/dx$; if $G(x)$ is discrete, then $g(x)$ denotes the probability that the claim size equals x
$H_t(x)$	compound distribution of the total claim size in the interval $[0, t]$
$h_t(x)$	in case $G(x)$ is continuously differentiable we write $h_t(x) = dH_t(x)/dx$, else we denote by $h_t(x)$ the probability that the total claim size equals x
$P_\Delta(\cdot)$	the operator P_Δ projects a function onto a $(n + 3)$ dimensional space (this will be introduced specifically in the text)
n	the number of discretization points
Δ	the fineness of the discretization, i.e. $\Delta = x_n/n$, where x_n , a proposed claim amount, is the right endpoint of the approximation interval $[0, x_n]$
x_i	interpolation point, i.e. $x_i = i\Delta$

2. Panjer's recursion

We assume that claims arrive following an ordinary renewal process and that the individual claims are independent and identically distributed. Consider the probability (density) function of the total claim size over a time period $[0, t]$. In case $G(x)$ is a continuous distribution then, by renewal theory, we can derive the following integral equation:

$$h_t(x) = \int_0^t \int_0^x h_{t-u}(x-y)g(y)f(u) dy du. \quad (2.1)$$

In case $G(x)$ is a discrete distribution over $x_i = i\Delta$, we have the equivalent discrete version

$$h_t(i\Delta) = \int_0^t \sum_{j=1}^{i-1} h_{t-u}((i-j)\Delta)g(j\Delta)f(u) du. \quad (2.2)$$

In the article, Panjer (1981), Panjer has shown that in case

$$p_m(t) = \left(a(t) + \frac{b(t)}{m} \right) p_{m-1}(t) \tag{2.3}$$

for some $a(t)$ and $b(t)$, we can get rid of the first integral in Eqs. (??) and (??), yielding the following result:

$$h_t(x) = p_1(t)g(x) + \int_0^x (a(t) + b(t)y/x)g(y)h_t(x - y) dy \tag{2.4}$$

or

$$h_t(i \Delta) = \sum_{j=1}^i (a(t) + b(t)j/i)g(j \Delta)h_t((i - j)\Delta). \tag{2.5}$$

In Sundt and Jewell (1981) it has been shown that Eq. (??) holds only in the following cases: $p_m(t)$ is a Poisson ($p_m(t) = e^{-\lambda t}(\lambda t)^m/m! \rightarrow a(t) = 0, b(t) = \lambda t$), binomial ($p_m(t) = k(t)!/(m!(k(t) - m)!)p(t)^m(1 - p(t))^{k(t)-m} \rightarrow a(t) = -p(t)(1 - p(t)), b(t) = (k(t) + 1)a(t)$), negative binomial ($p_m(t) = (k(t) + m - 1)!/(m!(k(t) - 1)!)p(t)^m(1 - p(t))^{k(t)} \rightarrow a(t) = p(t), b(t) = (k(t) - 1)p(t)$) or a geometric probability ($p_m(t) = p(t)^m(1 - p(t)) \rightarrow a(t) = p(t), b(t) = 0$).

The discrete version (??) is usually referred to as Panjer’s recursion formula and provides us with a fast scheme to compute the probabilities $h_t(x_i), i = 1, \dots, n$. We will pay attention to the continuous version of Panjer’s recursion. A popular method to solve the continuous version of Panjer’s recursion (??) is by discretizing the function $G(x)$ (or $g(x)$) such that the discrete version can be used again. We will shortly describe two elaborations of this idea.

In numerical analysis the simplest way to approximate an integral numerically is by writing the integrand as a piecewise linear function. By the idea of the usual midpoint method in numerical analysis, cf. Xie (1989), we obtain the general approximation

$$\int_0^{x_n} \phi(x) d\psi(x) \approx \sum_{i=1}^n \phi((i - 0.5)x_n/n)(\psi(ix_n/n) - \psi((i - 1)x_n/n)) \tag{2.6}$$

for any given function $\phi(\cdot)$ and $\psi(\cdot)$. If we apply this idea to Eq. (??), after first performing partial integration, we can derive the following approximation:

$$h_t(x_n) \approx \frac{p_1(t)g(x_n) - 0.5\Delta(a(t) + b(t))g(x_n)h_t(0) + \sum_{i=1}^n \Delta(a(t) + b(t)i/n)g(x_i)h(x_n - x_i)}{1 - 0.5\Delta a(t)g(0)}. \tag{2.7}$$

This corresponds slightly modified to the discrete version of Panjer’s recursion.

Another method, proposed by Panjer himself and Lutek (1983), is to discretize the cumulative distribution $G(x)$ instead of $g(x)$. Basically we get $\tilde{G}(x) = G(x_i)$ for all x in a predefined environment of x_i . The x_i and the environments can also be chosen e.g. such that the expectation, variance and some cumulative probabilities are matched or the $\tilde{G}(x)$ is stochastically greater or smaller than $G(x)$. Suppose we set $\tilde{g}(x_i) = G(x_i) - G(x_{i-1})$ (i.e. stochastically smaller) we get

$$h_t(x_i) \approx \sum_{j=1}^i (a(t) + b(t)x_j/x_i)\tilde{g}(x_j)h_t(x_i - x_j). \tag{2.8}$$

Notice that this procedure may require numerical integration in case $G(x)$ is not analytically known, e.g. when $G(x)$ follows a Gamma distribution with a noninteger shape parameter. It can be shown that the approximation $h_t(x_i)$

has an error of order $O(\Delta^2)$, cf. Burden and Faires (1989). The main advantage of both methods is that they are easy to understand and easily implemented. In Section 3 we will describe a more sophisticated approach to solve the continuous version of Panjer’s recursion, which is much more accurate and returns a C^2 function instead of a discrete function.

3. Projection method

The idea of the projection method is to project the integral equation Eq. (2.4), i.e. the functions y , $g(y)$ and $h(y)$, onto a certain space (e.g. the space of all step functions) and to solve the integral equation within this space. As a matter of fact, Panjer’s discrete recursion scheme can be seen as a particular case of the projection method.

In Panjer’s discrete recursion scheme as implemented by (??) the functions y , $g(y)$, $h(y)$ in (??) are approximated by square functions. Let us explain; let $\phi_i(x)$ be the function that equals 1 for $x_i \leq x < x_{i+1}$ and zero otherwise. Further let us assume that $x_i = i\Delta$. If we write $g(y) \approx \sum_{i=0}^n \gamma_i \phi_i(y)$ with $\gamma_i = (G(x_i) - G(x_{i-1}))/\Delta$, the function y in the integral of Eq. (??) by $y \approx \sum_{i=0}^n i\Delta \phi_i(y)$ and $h_t(x - y) \approx \sum_{i=0}^n \eta_i \phi_i(x - y)$, where the η_i are unknown, then after substitution into Panjer’s continuous recursion (??) we get

$$\sum_{i=0}^n \eta_i \phi_i(x) = (a(t) + b(t))p_0(t) \sum_{i=0}^n \gamma_i \phi_i(x) + \int_0^x \left(a(t) + b(t) \sum_{i=0}^n i\Delta \phi_i(y)/x \right) \sum_{i=0}^n \gamma_i \phi_i(y) \sum_{i=0}^n \eta_i \phi_i(x - y) dy.$$

Since the functions $\phi_i(\cdot)$ are known we can perform the integration analytically. Noticing that $\gamma_0 = 0$, putting $p_0(t) = \eta_0\Delta$ and setting x equal to x_{n+1}^- , the left-hand limit to x_{n+1} , yields

$$\eta_n = \sum_{i=1}^n (a(t) + b(t)i/n)\gamma_i \Delta \eta_{n-i}$$

in which we recognize again Panjer’s discrete recursion.

Clearly, square functions are not very suitable for obtaining good approximations. Using the above methodology an improvement of Panjer’s recursion becomes self-evident.

Instead of using square functions we suggest cubic splines, cf. Burden and Faires (1989). A cubic spline, $S(x)$, is defined by

$$S(x) = \begin{cases} 0, & x \leq -2, \\ \frac{1}{6}\{(2-x)^3 - 4(1-x)^3 - 6x^3 + 4(1+x)^3\}, & -2 < x \leq -1, \\ \frac{1}{6}\{(2-x)^3 - 4(1-x)^3 - 6x^3\}, & -1 < x \leq 0, \\ \frac{1}{6}\{(2-x)^3 - 4(1-x)^3\}, & 0 < x \leq 1, \\ \frac{1}{6}(2-x)^3, & 1 < x \leq 2, \\ 0, & 2 < x. \end{cases}$$

In Fig. 1 $S(x)$ is depicted.

The functions $\phi_i(x)$ are now given by $\phi_i(x) = S(x - x_i/\Delta)$. If we substitute $h(x) = \sum_{i=-1}^{n+1} \eta_i \phi_i(x)$ (a spline approximation is defined over $i = -1, \dots, n + 1$, cf. Burden and Faires (1989)), with η_i unknown, into Eq. (??)

Consider again Eq. (??) with $h_t(x - y)$ in the integral replaced by the approximation $h_t(x) \approx \sum_{i=-1}^{n+1} \eta_i \phi_i(x)$:

$$h_t(x) = p_1(t)g(x) + \int_0^x (a(t) + b(t)y/x) g(y) \sum_{j=-1}^{n+1} \eta_j \phi_j(x - y) dy.$$

Let us introduce the function

$$J(x) = \int_0^x (a(t) + b(t)y/x) g(y) \sum_{j=-1}^{n+1} \eta_j \phi_j(x - y) dy.$$

Since the operator P_Δ is linear, it follows that

$$P_\Delta(h_t) = P_\Delta(p_1(t)g) + P_\Delta(J).$$

By definition this equals

$$Q\eta = P_\Delta(p_1(t)g) + K\eta \tag{4.2}$$

with K an $(n + 3) \times (n + 3)$ matrix, where the elements K_{ij} are given by

$$K_{ij} = \begin{cases} \Delta \lim_{x \downarrow 0} \frac{d}{dx} \int_0^x (a(t) + b(t)y/x) g(y) \phi_j(x - y) dy, & i = -1, \\ \int_0^{i\Delta} (a(t) + b(t)y/(i\Delta)) g(y) \phi_j(i\Delta - y) dy, & i = 0, \dots, n, \\ \Delta \lim_{x \uparrow n\Delta} \frac{d}{dx} \int_0^x (a(t) + b(t)y/x) g(y) \phi_j(x - y) dy, & i = n + 1. \end{cases} \tag{4.3}$$

The first row of the matrix is easily simplified:

$$K_{-1j} = \Delta \lim_{x \downarrow 0} \frac{d}{dx} \int_0^x (a(t) + b(t)y/x) g(y) \phi_j(x - y) dy = \Delta(a(t) + b(t))g(0)\phi_j(0),$$

which equals zero for $j > 1$.

Consider the following integrals:

$$\mathcal{L}a(l, k) = \int_{l+k}^{l+k+1} \Delta g(\Delta t) S(l - t) dt \quad \text{and} \quad \mathcal{L}b(l, k) = \int_{l+k}^{l+k+1} \Delta t g(\Delta t) S(l - t) dt,$$

where l ranges from -1 to $n + 1$ and k from -2 to 1 . The evaluation of these integrals can be done by e.g. using a Gaussian quadrature formula. In case $a(t) = 0$ (Poisson arrivals) the integrals $\mathcal{L}a(l, k)$ need not to be computed. The middle rows, $i = 0, \dots, n$, give

$$K_{ij} = \sum_{k=\max(0, i-j-2)}^{\min(i-1, i-j+1)} a(t) \mathcal{L}a((i - j), k) + \frac{b(t)}{i} \mathcal{L}b((i - j), k) \quad \text{for } j \leq i + 1$$

For $j > i + 1$ the elements K_{ij} are zero. The derivative of a spline is a piecewise second degree polynomial and hence the derivative can be written as a piecewise convex combination of splines again. Therefore the last row can be rewritten as

$$\begin{aligned}
 K_{n+1,j} &= \Delta \lim_{x \uparrow n\Delta} \frac{d}{dx} \int_0^x (a(t) + b(t)y/x)g(y)\phi_j(x-y) dy \\
 &= \Delta(a(t) + b(t))g(n\Delta)\phi_j(0) + \Delta \int_0^{n\Delta} -b(t)y/(n\Delta)^2 g(y)\phi_j(n\Delta - y) dy \\
 &\quad + \Delta \int_0^{n\Delta} (a(t) + b(t)y/(n\Delta))g(y)\phi'_j(n\Delta - y) dy \\
 &= \Delta(a(t) + b(t))g(n\Delta)S(-j) - \sum_{k=\max(0,n-j-2)}^{\min(n-1,n-j+1)} \frac{b(t)}{n^2} \mathcal{L}b(n-j,k) \\
 &\quad + \sum_{k=\max(0,n-j-2)}^{\min(n-1,n-j+1)} \sum_{i=-2}^1 \alpha_{i,n-j-k-1} (a(t)\mathcal{L}a(i+1+k,k) + \frac{b(t)}{n} \mathcal{L}b(i+1+k,k)),
 \end{aligned}$$

where the α_{ij} , with i and j ranging from -2 up to 1 , are given by

$$\{\alpha_{ij} | i = -2, \dots, 1(\text{rows}), j = -2, \dots, 1(\text{columns})\} = \begin{bmatrix} 11/6 & 1/3 & -1/6 & 1/3 \\ -3 & 1/2 & 1 & -3/2 \\ 3/2 & -1 & -1/2 & 3 \\ -1/3 & 1/6 & -1/3 & -11/6 \end{bmatrix}.$$

The system that we have to solve is given by $(Q - K)\eta = P_\Delta(p_1(t)g)$. The matrix $(Q - K)$ has a special structure. It is an upper band-width matrix with band-width 1, i.e. $(Q - K)_{i,j} = 0$ for $j > i + 1$, except for the element $(Q - K)_{-1,1}$. This makes a factorization of the matrix $(Q - K)$ in a lower and an upper triangular matrix very easy. Moreover it can be proven that such a decomposition always exists and is numerically stable, see Smith (1997). For completeness we will give the factorization known as Doolittle's method, cf. Burden and Faires (1989), adapted to the matrix $(Q - K)$. We write $(Q - K) = L \cdot U$.

Doolittle's factorization (adapted).

```

Set  $L_{-1,-1} = 1$  and  $U_{-1,-1} = (Q - K)_{-1,-1}$ 
Set  $U_{-1,j} = (Q - K)_{-1,j}$  for  $j = 0 \dots 1$ 
Set  $L_{j,-1} = (Q - K)_{j,-1}/U_{-1,-1}$  for  $j = 0 \dots (n + 1)$ 
For  $i = 0 \dots (n + 1)$  do
  Set  $L_{i,i} = 1$  and  $U_{i,i} = (Q - K)_{i,i} - L_{i,i-1}U_{i-1,i}$ .
  Except in case  $i = 1$ , then set  $U_{1,1} = (Q - K)_{1,1} - L_{1,0}U_{0,1} - L_{1,-1}U_{-1,1}$ 
  For  $j = (i + 1) \dots (n + 1)$  do
    In case  $i = 0$  then  $U_{0,1} = (Q - K)_{0,1} - L_{0,-1}U_{-1,1}$ 
    else  $U_{i,i+1} = (Q - K)_{i,i+1}$ 
    In case  $i = 1$  then  $L_{j,1} = ((Q - K)_{j,1} - L_{j,0}U_{0,1} - L_{j,-1}U_{-1,1})/U_{1,1}$ 
    else  $L_{j,i} = ((Q - K)_{j,i} - L_{j,i-1}U_{i-1,i})/U_{i,i}$ 
  end
end

```

end

The vector η follows now by solving the triangular systems

$$L\alpha = P_{\Delta}(p_1(t)g) \quad \text{and} \quad U\eta = \alpha$$

subsequently for α and η .

Let us for convenience resume the method shortly in pseudo-code.

The projection method based on cubic splines to solve the equation

$$h_t(x) = p_1(t)g(x) + \int_0^x (a(t) + b(t)y/x)g(y)h_t(x-y) dy$$

Input: $g(\cdot)$, t , $a(t)$, $b(t)$, $p_0(t)$, the number of interpolation points n and the approximation interval $[0, x_n]$.

Set $\Delta = x_n/n$ and $p_1(t) = (a(t) + b(t))p_0(t)$.

Step 1. Compute the $(n+3)$ vector $P_{\Delta}(p_1(t)g)$.

The index runs from -1 to $n+1$. See Eq. (??).

Step 2. Compute the $(n+3) \times (n+3)$ matrix K . See Eq. (??) and further.

Construct the matrix Q according to Eq. (??).

Step 3. Compute the upper and lower triangular matrices L and U such that $L \cdot U = Q - K$ by means of Doolittle's factorization.

Step 4. Solve subsequently the systems $L\alpha = P_{\Delta}(p_1(t)g)$ and $U\eta = \alpha$ for α and η .

Output: The vector η .

The approximation is now given by $h_t(x) \approx \sum_{i=-1}^{n+1} \eta_i \phi_i(x)$, with $\phi_i(x) = S(x/\Delta - i)$ and $x \in [0, x_n]$.

5. A theoretical result

In this section we consider the performance of the projection method. In the following theorem we state the accuracy of the approximation. We will denote the approximation of $h_t(x)$ by $\tilde{h}_t(x)$.

Theorem 1. *Suppose the probability density function $g(x)$ of the claim size is four times differentiable on the intervals $[k\Delta, (k+1)\Delta]$, $k = 0, \dots, (n-1)$, $g \in \mathcal{C}^2[0, n\Delta]$ and $|g^{(4)}(x)| < L$ for $x \in [0, n\Delta]$ then for the approximation of $h_t(x)$ by means of the method described in Section 4 it holds that*

$$|h_t^{(k)}(x) - \tilde{h}_t^{(k)}(x)| < C_k \Delta^{4-k} \quad x \in [0, n\Delta] \quad \text{and} \quad k = 0, 1, 2, 3$$

and in the interpolation points we have

$$|h_t(i\Delta) - \tilde{h}_t(i\Delta)| < D\Delta^5, \quad i = 0, \dots, n$$

Here C_k and D are constants independent of the number of interpolation points n . Moreover we have $\tilde{h}_t(x) \in \mathcal{C}^2[0, n\Delta]$.

Proof. A proof of these results is given in Smith (1997). \square

6. Normalization

With Panjer’s recursion, $h_t(i\Delta)$ is just a multiple of $h_t(0)$. This means that an error in $h_t(0)$ will immediately cause an error in $h_t(i\Delta)$. This also holds for the projection method. If we make an error in the computation of K_{0j} , $j = -1, \dots, 1$, then this causes our approximation to be a multiple off from the real function. This can lead to serious problems in case the function $g(x)$ is not analytic in zero, e.g. the lognormal density is not analytic in nil and the Weibull density is not even defined in zero in case its shape parameter is smaller than 1. In these cases the method still converges with an accuracy of $O(\Delta^5)$, but with a large Δ the method does not seem to perform as well as it does with a large Δ for functions that are neat in zero.

We propose to solve this problem by normalization. In case we know $h_t(\hat{x})$ in some \hat{x} exactly then we can correct the error made over $[0, \Delta]$ by multiplying our approximation such that in \hat{x} the approximation is correct. We propose the following scheme, which increases the computation time of the procedure to $(n^2 \log(n))$, but which accuracy increase easily compensates for the extra effort. We assume that the number of points, n , is a power of 2.

Input: The number of points $n = 2^k$ and the approximation interval $0, \dots, x_n$.

Output: The approximation $\tilde{h}_t(x)$, for $0 \leq x \leq x_n$.

Step 0. Set $i = 1$ and $x_n^i = x_n$.

Step 1. Use the projection method with n points to approximate $h_t(x)$ on the interval $0, \dots, x_n^i$. Call the approximation $h_t^i(x)$.

Step 2. Set $i = i + 1$ and $x_n^i = \frac{1}{2}x_n^{i-1}$. In case $x_n^i \geq \Delta$ (or $i \leq k$) then return to Step 1.

Step 3. For i from k to 1 set $h_t^{i-1}(x) := h_t^i(x) \times h_t^i(x_n^i) / h_t^{i-1}(x_n^i)$.

Step 4. Set $\tilde{h}_t(x) = h_t^1(x)$.

7. Numerical examples

In this section we will illustrate the performance of the projection method by two examples. We are interested in the distribution of aggregate claims over one year. In the first example the individual claim amount distribution is given by an exponential distribution with mean 1. The number of claims per year is Poisson distributed with mean 10. The approximation interval of the aggregate claim size is $0, \dots, 45$. Since the individual claim amount distribution is exponential we can give an explicit expression for the distribution of the aggregate claim size. For our set of parameters we have

$$h_1(x) = 10 e^{-10-x} \sum_{k=0}^{\infty} \frac{(10x)^k}{k!(k+1)!} \tag{7.1}$$

In Tables 1 and 2 the results are presented of the Projection method and Panjer’s recursion. The function $\tilde{h}_1(x)$ is evaluated in 15 points between 0 and 45 and of this evaluation the digits are printed up to the first wrong one. Each number in the top row denotes the number of interpolation points (n) in which the interval $[0, 45]$ has been divided, series denotes the analytic solution as given by (??). The discretization procedure that we have used for Panjer’s recursion is somewhat different from what is mentioned in Section 3. We have taken the following:

$$g(0)\Delta = G(\frac{1}{2}\Delta),$$

$$g(i\Delta)\Delta = G((i + \frac{1}{2})\Delta) - G((i - \frac{1}{2})\Delta) \quad \text{for } i = 1, \dots, n.$$

Table 1
Projection method and analytical solution

Claim size	64	128	256	512	Series
2.8125	0.02451420	0.024514280	0.024514285	0.02451428622	0.024514286253
5.6250	0.07163555	0.0716352534	0.071635233	0.0716352325	0.071635232428
8.4375	0.092855	0.09285613	0.092856144	0.09285614531	0.092856145376
11.2500	0.0773004	0.07730076	0.077300783	0.0773007846	0.077300784654
14.0625	0.0481190	0.04811914	0.048119156	0.04811915730	0.048119157333
16.8750	0.02427761	0.024277626	0.024277626	0.024277627020	0.024277627023
19.6875	0.0104251	0.010425169	0.0104251687	0.010425168677	0.010425168676
22.5000	0.003935502	0.0039355013	0.003935501269	0.003935501265	0.003935501265
25.3125	0.001336083	0.001336086	0.00133608701	0.001336087029	0.001336087030
28.125	0.000414806	0.0004148090	0.00041480920	0.000414809215	0.000414809216
30.9375	0.000119277	0.0001192795	0.00011927963	0.000119279641	0.000119279641
33.7500	0.000032084	0.00003208554	0.00003208558	0.000032085590	0.000032085590
36.5625	0.0000081380	0.00000813832	0.000008138343	0.000008138344	0.000008138344
39.3750	0.0000019590	0.00000195911	0.00000195911	0.000001959120	0.000001959120
42.1875	0.00000044	0.000000450002	0.000000450004	0.000000450004	0.000000450004

Table 2
Panjer's recursion

64	128	256	512	1024	2048	4096	8192
0.01	0.020	0.022	0.023	0.023	0.0241	0.0243	0.0244
0.06	0.06	0.06	0.070	0.0710	0.0713	0.0714	0.0715
0.091	0.0923	0.0927	0.09281	0.09284	0.092851	0.092854	0.092855
0.08	0.079	0.078	0.078	0.0777	0.0775	0.0774	0.07735
0.05	0.05	0.049	0.0489	0.0485	0.0483	0.0482	0.04817
0.027	0.026	0.025	0.0248	0.0245	0.0244	0.0243	0.0243
0.012	0.011	0.011	0.0107	0.0105	0.01049	0.01046	0.01044
0.004	0.004	0.00418	0.0040	0.00399	0.00396	0.00395	0.00394
0.0016	0.0015	0.0014	0.00138	0.00135	0.00134	0.00134	0.001339
0.0005	0.00047	0.00044	0.00043	0.00042	0.000418	0.000416	0.000415
0.00015	0.00013	0.00012	0.00012	0.00012	0.00012	0.0001198	0.0001195
0.00004	0.000037	0.000034	0.000033	0.0000327	0.0000324	0.0000322	0.0000321
0.00001	0.000009	0.0000088	0.0000085	0.0000083	0.0000082	0.00000818	0.00000816
0.000002	0.000002	0.000002	0.000002	0.000002	0.00000198	0.00000197	0.00000196
0.0000006	0.0000005	0.00000049	0.00000047	0.00000046	0.000000455	0.000000452	0.000000451

This way the discretization of $G(x) \approx \sum_{i=0}^{\lfloor x/\Delta \rfloor} g(i\Delta)\Delta$ is not stochastically smaller or greater as in Section 3, but circles around $G(x)$ crossing it from below at each discretization point. This yields a better performance of Panjer's recursion. Specifically we have set $h_t(0) = 1$ and computed $h_t(x_k) = (10/k) \sum_{i=1}^k ig(x_i)\Delta h_t(x_k - x_i)$, $k = 1, \dots, n$. Subsequently we have scaled $h_t(x)$, i.e., $h_t(x) := \exp(-10(1 - g(0)\Delta))h_t(x)/\Delta$.

In the second example the individual claim amount distribution is a lognormal distribution with mean 1 and variance 3. The rate of the Poisson process equals 20 and the interval of interest is $0, \dots, 60$. This example is taken from Dickson (1995). In this case an explicit formula is not available. For comparison we have taken the approximation of the projection method with 2048 points to be exact. This is of course not necessarily true, but we can assume that the digits printed this way for the projection method with 1024 points are correct up to the last one. Since the lognormal density is not analytic in nil we have used the normalization procedure. The results are presented in Tables 3 and 4.

From the tables it is clear that the projection method needs far less points to attain a high order of accuracy. Moreover using more points really makes a difference while with Panjer's recursion the convergence is very slow. Besides the better accuracy it should be stressed that the projection method has also as an advantage that it returns

Table 3
Projection method's result

Claim size	64	128	256	512	1024	2048
3.75	0.0011808	0.00118055	0.001180584	0.0011805859	0.00118058587	0.0011805858549
7.50	0.01485	0.0148416	0.014841580	0.14841584	0.0148415838	0.014841583599
11.25	0.04006	0.0400348	0.040034303	0.040034308	0.0400343074	0.040034307176
15.00	0.0545	0.054471	0.05447058	0.054470591	0.0544705904	0.054470590789
18.75	0.0513	0.051288	0.051287987	0.05128799	0.0512879897	0.051287989817
22.50	0.03899	0.038968	0.038967447	0.0389674489	0.03896744883	0.038967448822
26.25	0.02605	0.026041	0.026040783	0.02604078424	0.026040784234	0.026040784233
30.00	0.01615	0.0161388	0.0161385457	0.01613854582	0.016138545838	0.016138545836
33.75	0.009608	0.0096014	0.00960122117	0.00960122118	0.0096012211956	0.0096012211957
37.50	0.005617	0.0056131	0.00561301094	0.00561301094	0.005613010955	0.0056130109563
41.25	0.003279	0.00327709	0.0032770262	0.003277026301	0.003277026306	0.0032770263072
45.00	0.001933	0.001932	0.00193198566	0.001931985675	0.0019319856784	0.0019319856786
48.75	0.001159	0.0011588	0.00115878453	0.001158784543	0.0011587845443	0.0011587845444
52.50	0.0007109	0.00071043	0.00071041781	0.0007104178264	0.00071041782660	0.00071041782669
56.25	0.0004466	0.00044634	0.0044633886	0.0004463388753	0.00044633887522	0.00044633887521

Table 4
Panjer's results

64	128	256	512	1024	2048	4096	8192	16384	32768
0.0019	0.0010	0.00097	0.0010	0.00110	0.00114	0.00116	0.00117	0.00117	0.00117
0.017	0.013	0.013	0.0141	0.0144	0.0146	0.0147	0.0147	0.01481	0.01483
0.041	0.038	0.038	0.039	0.039	0.039	0.039	0.039	0.04001	0.04002
0.053	0.053	0.0541	0.0542	0.0543	0.05443	0.05445	0.05446	0.05446	0.05446
0.050	0.0516	0.0517	0.0515	0.0514	0.0513	0.0513	0.0513	0.05129	0.05129
0.037	0.039	0.039	0.039	0.039	0.039	0.039	0.03899	0.03898	0.03897
0.025	0.0267	0.0266	0.0264	0.0262	0.0261	0.02608	0.02606	0.02605	0.026046
0.015	0.0166	0.0165	0.0163	0.0162	0.0162	0.01617	0.01615	0.01614	0.01614
0.0093	0.0099	0.0098	0.0097	0.00968	0.00964	0.00962	0.00961	0.009606	0.009603
0.0054	0.0058	0.0057	0.0057	0.00566	0.00563	0.00562	0.005618	0.005615	0.005614
0.00320	0.0033	0.0033	0.0033	0.0033	0.00329	0.00328	0.00328	0.003278	0.0032778
0.0018	0.00199	0.00198	0.00196	0.00194	0.001939	0.001935	0.001933	0.001932	0.001932
0.00113	0.00119	0.00119	0.00117	0.00116	0.00116	0.00116	0.001159	0.001159	0.001159
0.00069	0.00073	0.00072	0.00072	0.000715	0.000713	0.000711	0.000711	0.0007107	0.0007105
0.00043	0.00045	0.00045	0.00045	0.000449	0.000447	0.000447	0.0004467	0.0004465	0.0004464

a C^2 function, so that all values over the interval of interest are approximated and differentiation or integration can be done analytically.

We have implemented the projection method in Borland Pascal ©7.0 (Borland International Inc., Scotts Valley, USA, 1992) and Panjer's recursion in GAUSS™, (Aptech Systems Inc., Maple Valley, USA, 1994), on a 486 33 Mhz machine. Borland Pascal © has as an advantage that for-loops are performed much faster than in GAUSS™. GAUSS™ has as an advantage that data-structures are allowed to be quite large. Since with the Projection method we do not expect to need a large number of points we have implemented it in Borland Pascal ©. Therefore, comparison of cpu-time needed for both methods is not fair. To give an idea, the cpu time of the projection method was approximately given by $2.5n^2 \cdot 10^{-5}$ ($2.5n^2 \cdot 10^{-5} \log n$ with normalization) s and Panjer's recursion by $2n^2 \cdot 10^{-5}$ s. To attain an accuracy of 4 digits we should need more than 8192 points with Panjer's recursion in the first example. 8192 points took already over 21 min. With the Projection method 64 points were already satisfactory taking 0.3 s. For example 2 the same holds, only here the Projection method took 1.2 s because of the normalization.

8. Conclusions

In this article we have presented a powerful tool for the computation of compound distributions in case the individual claim amount density is twice continuously differentiable. Compared with Panjer's recursion the method is much faster to yield the same accuracy. Evenmore the method returns a twice continuously differentiable function which can be evaluated in any point and allows for analytical differentiation and integration.

Acknowledgements

The authors would like to thank the anonymous referee for detailed comments.

References

- Borland Pascal ©7.0 (1992). Borland International Inc., Scotts Valley, USA.
- Bühlmann, H. (1984). Numerical evaluation of the compound Poisson distribution: recursion or Fast Fourier Transform? *Scandinavian Actuarial Journal*, 116–126.
- Burden, R.L. and J.D. Faires (1989). *Numerical Analysis*. PWS-KENT Publishing Company, Boston.
- Dickson, D.C.M. (1995). A review of Panjer's recursion formula and its applications, *British Actuarial Journal* 1, 107–124.
- GAUSSTM (1994). Aptech Systems Inc., Maple Valley, USA.
- Kuon, S., M. Radtke and A. Reich (1993). An appropriate way to switch from the individual risk model to the collective one, *ASTIN Bulletin* 23, 23–54.
- Lutek, B.W. and H.H. Panjer (1983). Practical aspects of stop-loss calculations. *Insurance: Mathematics and Economics* 2, 159–177.
- Panjer, H.H. (1981). Recursive evaluation of a family of compound distributions. *ASTIN Bulletin* 12, 22–26.
- Smith, M.A.J. (1997). On the availability of failure prone systems, Ph.D. thesis, Econometric Institute, Erasmus University Rotterdam.
- Sundt, B. and W.S. Jewell (1981). Further results on recursive evaluation of compound distributions. *Astin Bulletin* 12, 27–39.
- Xie, M. (1989). On the solution of renewal type integral equations. *Communications in Statistics B* 18, 281–293.