



## Another approach to the evaluation of a certain multivariate compound distribution

Elena-Gratiela ROBE-VOINEA and Raluca VERNIC

### Abstract

In this work, we consider the multivariate aggregate model introduced in [11], model that takes into account the case when different types of claims affect in the same time an insurance portfolio under some specific assumptions related to the number of claims. For the probability function of the corresponding multivariate compound distribution, [11] obtained an exact recursive formula proved using the properties of the probability generating function. In this paper, we present a new shorter proof of the same formula that we also extend to a new form. Moreover, we present an alternative approximate method to evaluate the compound distribution based on the Fourier transform, and we compare both methods on a numerical example.

### 1 Introduction

To be able to evaluate premiums and ruin probabilities, actuaries study aggregate claims distributions corresponding to insurance portfolios. These distributions result from individual or collective models. In this paper, we consider a collective model in a multivariate setting. This model has been introduced in [10] to study aggregate claims in the case when different types of claims simultaneously affect an insurance portfolio (e.g., floods, storms or earthquakes),

---

Key Words: insurance model, multivariate aggregate claims, multinomial distribution, recursions, Fast Fourier Transform.

2010 Mathematics Subject Classification: 62P05, 91B30, 91B70.

Received: August, 2015

Revised: November, 2015

Accepted: November, 2015

and has the form:

$$(S_1, \dots, S_m) = \left( \sum_{l=0}^{N_1} U_{1l} + \sum_{k=0}^{N_0} L_{1k}, \dots, \sum_{l=0}^{N_m} U_{ml} + \sum_{k=0}^{N_0} L_{mk} \right), \quad (1)$$

where  $m \geq 2$  is the number of different types of claims affecting the portfolio,  $S_k$  denotes the aggregate claims of type  $k$ ,  $N_k$  the number of claims of only type  $k$ , and  $N_0$  the number of common claims. As mentioned in [10], model (1) extends to the multivariate case the one from [5]. The resulting multivariate distribution is called *compound distribution*, being built from the distributions of the claims and of their numbers. We also denote  $\mathbf{S} = (S_1, \dots, S_m)$  and  $\mathbf{N} = (N_0, \dots, N_m)$ .

Considering that all the involved random variables (r.v.s) are of discrete type, the usual assumptions related to model (1) are :

- Each set of claim sizes  $(U_{jl})_{l \geq 1}$  are non-negative, independent and identically distributed (i.i.d.) r.v.s as a generic r.v. denoted  $U_j, 1 \leq j \leq m$ , independent of the claim numbers and of the other claim sizes, including  $(L_{1k}, \dots, L_{mk})$ .
- The random vectors  $(L_{1k}, \dots, L_{mk})_{k \geq 1}$  are also non-negative i.i.d. as the generic random vector  $\mathbf{L} = (L_1, \dots, L_m)$ , and independent of the claim numbers. The components of  $\mathbf{L}$ , however, are usually dependent.
- By convention,  $U_{j0} = L_{j0} = 0, \forall j = \overline{1, m}$ .

An exact recursion for the evaluation of the distribution of this model under the assumption that  $\mathbf{N}$  follows a multivariate Poisson distribution is presented in [10]. In [11], two different assumptions on  $\mathbf{N}$  are considered that we shall also take into account in the following:

**A1** The first one is related to the total number of claims  $N = N_0 + N_1 + \dots + N_m$ , whose probability function (p.f.) is assumed to satisfy the Panjer-type recursion

$$\Pr(N = n) = \left( a + \frac{b}{n} \right) \Pr(N = n - 1), \forall n \geq 1,$$

for some constants  $a, b \in \mathbb{R}$  (for details on Panjer's class, see [8] or [13]);

**A2** The second one concerns the conditional distribution of  $\mathbf{N}$  given  $N = n$ , which is assumed to be multinomial  $Mnom(n; p_1, \dots, p_m)$  with the

parameters  $n \in \mathbb{N}$  and  $p_1, \dots, p_m \in (0, 1)$  such that  $p_0 := 1 - \sum_{i=1}^m p_i \in (0, 1)$ . We recall that (see, e.g., [6]), with  $n = \sum_{i=0}^m n_i$ ,

$$\Pr(N_0 = n_0, N_1 = n_1, \dots, N_m = n_m | N = n) = \frac{n!}{\prod_{i=0}^m n_i!} \prod_{i=0}^m p_i^{n_i}.$$

Both recursions obtained by [10] and [11] extend to the multivariate setting the ones presented by [5] in the bivariate case  $m = 2$ . In [11], the proof of the recursion for the corresponding multivariate compound distribution is based on properties of the probability generating function (p.g.f.); in next section, we shall give a new shorter proof to this recursive formula and simultaneously obtain a new such recursion. Moreover, in Section 3 we present an approximate alternative method to evaluate the compound distribution based on the Fast Fourier Transform (FFT), which has the advantage of being less time consuming than the recursive method especially when one needs to evaluate the distribution's tail. We compare the two methods on a numerical example and conclude.

To simplify the writing, we introduce more notation: we denote by  $f_{\mathbf{S}}$  the p.f. of  $\mathbf{S}$ , by  $f_j$  the p.f. of  $U_j, 1 \leq j \leq m$ , and by  $f_{\mathbf{L}}$  the p.f. of  $\mathbf{L}$ . The p.g.f. of a r.v. is denoted by  $g$  indexed with that r.v., and similarly, its characteristic function is denoted by  $\varphi$  indexed with its name. Also,  $\mathbf{n} = (n_0, \dots, n_m), \mathbf{t} = (t_1, \dots, t_m), \mathbf{0} = (0, \dots, 0), \mathbf{x} = (x_1, \dots, x_m)$ , similarly  $\mathbf{y}, \mathbf{z}$  and  $\mathbf{1}$ , while the difference  $\mathbf{x} - \mathbf{y}$  is componentwise.

## 2 The recursion - a shorter proof

From [11], we have that for the general model (1) under the assumptions (A1-A2), the p.g.f. of  $\mathbf{S}$  is given by

$$g_{\mathbf{S}}(\mathbf{t}) = g_N \left( \sum_{j=1}^m p_j g_{U_j}(t_j) + p_0 g_{\mathbf{L}}(\mathbf{t}) \right). \quad (2)$$

Using the properties of the p.g.f., the following recursive formula (3) has been proved in [11]. We shall now present a shorter proof based on similar recursions already proved in [12]. In the same time, we shall also obtain the new alternative recursive formula (4).

**Proposition 2.1.** *Under the assumptions (A1-A2) of model (1), the following*

starting value and recursive formulas hold:

$$f_{\mathbf{S}}(\mathbf{0}) = g_{\mathbf{S}}(\mathbf{0}) = g_N \left( \sum_{j=1}^m p_j f_j(0) + p_0 f_{\mathbf{L}}(\mathbf{0}) \right);$$

$$f_{\mathbf{S}}(\mathbf{x}) = K \left\{ a \sum_{\substack{j=1 \\ j \neq k}}^m p_j \sum_{y_j=1}^{x_j} f_j(y_j) f_{\mathbf{S}}(x_1, \dots, x_{j-1}, x_j - y_j, x_{j+1}, \dots, x_m) \right.$$

$$+ p_k \sum_{y_k=1}^{x_k} \left( a + b \frac{y_k}{x_k} \right) f_k(y_k) f_{\mathbf{S}}(x_1, \dots, x_{k-1}, x_k - y_k, x_{k+1}, \dots, x_m)$$

$$\left. + p_0 \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_k}{x_k} \right) f_{\mathbf{L}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) \right\}, \quad (3)$$

for  $x_k \geq 1, x_j \geq 0, \forall j \neq k$ ; and

$$f_{\mathbf{S}}(\mathbf{x}) = K \left\{ \sum_{j=1}^m p_j \sum_{y_j=1}^{x_j} \left( a + b \frac{y_j}{x_+} \right) f_j(y_j) \right.$$

$$\times f_{\mathbf{S}}(x_1, \dots, x_{j-1}, x_j - y_j, x_{j+1}, \dots, x_m)$$

$$\left. + p_0 \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_+}{x_+} \right) f_{\mathbf{L}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) \right\}, \quad \mathbf{x} > \mathbf{0}, \quad (4)$$

where  $K = \left[ 1 - a \left( \sum_{j=1}^m p_j f_j(0) + p_0 f_{\mathbf{L}}(\mathbf{0}) \right) \right]^{-1}$  and  $x_+ = \sum_{i=1}^m x_i$ .

*Proof.* Under the assumptions (A1-A2), model (1) can be represented as

$$(S_1, \dots, S_m) = \sum_{k=0}^N (C_{1k}, \dots, C_{mk}),$$

where, as before,  $N = N_0 + N_1 + \dots + N_m, C_{j0} = 0, \forall j = \overline{1, m}$ , while the random vectors  $(C_{1k}, \dots, C_{mk})_{k \geq 1}$  are i.i.d. as the generic  $\mathbf{C}$  having the p.f.

$$f_{\mathbf{C}}(\mathbf{y}) = \sum_{j=1}^m I(y_i = 0, \forall i = \overline{1, m}, i \neq j) p_j f_j(y_j) + p_0 f_{\mathbf{L}}(\mathbf{y}). \quad (5)$$

Here  $I$  denotes the indicator function defined by  $I(A) = 1$  if  $A$  is true, and 0 otherwise. From [12] (see also [13], formulas (15.4) and (15.5), respectively), it holds that

$$f_{\mathbf{S}}(\mathbf{x}) = \frac{1}{1 - af_{\mathbf{C}}(\mathbf{0})} \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_k}{x_k} \right) f_{\mathbf{C}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}), \quad x_k \geq 1, \quad (6)$$

$$f_{\mathbf{S}}(\mathbf{x}) = \frac{1}{1 - af_{\mathbf{C}}(\mathbf{0})} \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_+}{x_+} \right) f_{\mathbf{C}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}), \quad \mathbf{x} > \mathbf{0}. \quad (7)$$

By inserting (5) into (6) we obtain for  $x_k \geq 1$ ,

$$\begin{aligned} f_{\mathbf{S}}(\mathbf{x}) &= \frac{1}{1 - a \left( \sum_{j=1}^m p_j f_j(0) + p_0 f_{\mathbf{L}}(\mathbf{0}) \right)} \\ &\times \left\{ \sum_{j=1}^m p_j \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_k}{x_k} \right) I(y_i = 0, \forall i = \overline{1, m}, i \neq j) f_j(y_j) \right. \\ &\quad \left. \times f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) + p_0 \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_k}{x_k} \right) f_{\mathbf{L}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) \right\}. \end{aligned}$$

In the first sum, we separately consider the cases  $j = k$  and  $j \neq k$ .

- When  $j = k$ , combining  $I(y_i = 0, \forall i = \overline{1, m}, i \neq k)$  with  $\mathbf{y} > \mathbf{0}$  yields  $y_k \geq 1$ , hence the middle term of (3);

- When  $j \neq k$ , from  $I(y_i = 0, \forall i = \overline{1, m}, i \neq j)$  we clearly have  $y_k = 0$ , therefore

$$\sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_k}{x_k} \right) I(y_i = 0, \forall i = \overline{1, m}, i \neq j) f_j(y_j) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y})$$

becomes

$$\sum_{y_j=1}^{x_j} a f_j(y_j) f_{\mathbf{S}}(x_1, \dots, x_{j-1}, x_j - y_j, x_{j+1}, \dots, x_m),$$

from where formula (3) is immediate.

Similarly, by inserting (5) into (7) we obtain formula (4) for  $\mathbf{x} > \mathbf{0}$  as follows

$$\begin{aligned} f_{\mathbf{S}}(\mathbf{x}) &= K \left\{ \sum_{j=1}^m p_j \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_+}{x_+} \right) I(y_i = 0, \forall i = \overline{1, m}, i \neq j) f_j(y_j) \right. \\ &\quad \left. \times f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) + p_0 \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_+}{x_+} \right) f_{\mathbf{L}}(\mathbf{y}) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) \right\}, \end{aligned}$$

and

$$\begin{aligned} & \sum_{\mathbf{0} < \mathbf{y} \leq \mathbf{x}} \left( a + b \frac{y_+}{x_+} \right) I(y_i = 0, \forall i = \overline{1, m}, i \neq j) f_j(y_j) f_{\mathbf{S}}(\mathbf{x} - \mathbf{y}) \\ &= \sum_{y_j=1}^{x_j} \left( a + b \frac{y_j}{x_+} \right) f_j(y_j) f_{\mathbf{S}}(x_1, \dots, x_{j-1}, x_j - y_j, x_{j+1}, \dots, x_m), \end{aligned}$$

which completes the proof.  $\square$

### 3 Alternative method - Fast Fourier Transform

#### 3.1 The FFT algorithm

Apart the exact methods (like the recursive one presented above), some approximate techniques have also been proposed for evaluating aggregate claims distributions, with the purpose to simplify calculations and reduce the computing time (for details on these methods see, e.g., [7] or [13]). The Fast Fourier Transform is such a technique that strongly reduces the computing time, especially when one needs to evaluate the tail of the distribution. Moreover, this technique can be applied to models for which there is no recursion available. This is why the FFT received special attention in the actuarial literature, see, e.g., [2], [3], [5], [7], or [9]. In the following, we shall present the FFT algorithm corresponding to our model and compare it with the recursive method.

A FFT is an algorithm that computes the discrete Fourier transform and its inverse extremely fast. We recall that given an  $m$ -variate function  $f(\mathbf{x})$  defined on the integer values  $x_j = 0, 1, \dots, r_j - 1, 1 \leq j \leq m$ , its discrete Fourier transform  $\tilde{f}$  can be defined by (definition used in Matlab, in which we implemented the algorithm)

$$\tilde{f}(\mathbf{c}) = \sum_{x_1=0}^{r_1-1} \dots \sum_{x_m=0}^{r_m-1} f(\mathbf{x}) \exp \left\{ -2\pi i \sum_{j=1}^m \frac{x_j c_j}{r_j} \right\}, \quad c_j = 0, \dots, r_j - 1, \quad 1 \leq j \leq m,$$

having the inverse mapping

$$f(\mathbf{x}) = \frac{1}{\prod_{j=1}^m r_j} \sum_{c_1=0}^{r_1-1} \dots \sum_{c_m=0}^{r_m-1} \tilde{f}(\mathbf{c}) \exp \left\{ 2\pi i \sum_{j=1}^m \frac{x_j c_j}{r_j} \right\}, \quad x_j = 0, \dots, r_j - 1, \quad 1 \leq j \leq m.$$

For more details on Fourier transforms and their applications (major application in signal and image processing) see, e.g., [1].

To apply the FFT method, the values  $r_j$  must be powers of two for all  $j$ . For our model (1), we shall use the following algorithm proposed in [9] based on the FFT and its inverse (IFFT), algorithm that generalizes the one considered by [5] in the bivariate case. The algorithm needs the characteristic function  $\varphi_{\mathbf{S}}$  of  $\mathbf{S}$ , which can be express in terms of the p.g.f. of  $\mathbf{N}$  and of the characteristic functions of  $U_i$  and of  $\mathbf{L}$  as (see [9])

$$\begin{aligned} \varphi_{\mathbf{S}}(\mathbf{t}) &= g_{\mathbf{N}}(\varphi_{\mathbf{L}}(\mathbf{t}), \varphi_{U_1}(t_1), \dots, \varphi_{U_m}(t_m)) \\ &= g_{\mathbf{N}}\left(\sum_{j=1}^m p_j \varphi_{U_j}(t_j) + p_0 \varphi_{\mathbf{L}}(\mathbf{t})\right). \end{aligned} \tag{8}$$

For the last equality, we applied the formula of  $g_{\mathbf{N}}$  obtained in [11] for model (1) under the assumptions (A1-A2).

**FFT Algorithm**

*Step 1.* Set the truncation points for the r.v.s claim sizes  $U_j$  at  $r_j, 1 \leq j \leq m$ , and for  $\mathbf{L}$  at  $(r_1, \dots, r_m)$ . The truncated claim size distributions result as  $\mathbf{f}_j = \{f_j(0), f_j(1), \dots, f_j(r_j - 1)\}$  for  $U_j, 1 \leq j \leq m$ , and  $\mathbf{f}_0 = [f_{\mathbf{L}}(j_1, \dots, j_m)]_{j_1, \dots, j_m}$  for  $\mathbf{L}$ , where  $0 \leq j_l \leq r_l - 1, 1 \leq l \leq m$ . If necessary, the resulting vectors  $\mathbf{f}_j$  or the table  $\mathbf{f}_0$  can be padded with zeros to force the  $r_j$ s to be powers of two.

*Step 2.* Apply the one-dimensional FFT to  $\mathbf{f}_j$  yielding the vector  $\tilde{\mathbf{f}}_j, 1 \leq j \leq m$ ; then apply the multidimensional FFT to  $\mathbf{f}_0$ , yielding the multidimensional table  $\tilde{\mathbf{f}}_0$ .

*Step 3.* Use formula (8) to obtain the discrete characteristic function  $\tilde{\varphi}_{\mathbf{S}}(\mathbf{j}) = g_{\mathbf{N}}(\tilde{\mathbf{f}}_0(\mathbf{j}), \tilde{\mathbf{f}}_1(j_1), \dots, \tilde{\mathbf{f}}_m(j_m)), 0 \leq j_l \leq r_l - 1, 1 \leq l \leq m$ .

*Step 4.* Apply the multidimensional IFFT to  $\tilde{\varphi}_{\mathbf{S}}$  to obtain the p.f. of  $\mathbf{S}$ .

**Remark 3.1.** *The usual way to find the optimal  $r_j$ s consists in gradually increasing them (e.g., 32, 64, 128, 256 etc.) until the differences between the solutions obtained for the current values of the  $r_j$ s and the previous ones are no more significant.*

**Remark 3.2.** *When the claim sizes distributions are heavy tailed it is recommended to use the so-called “exponential tilting” method, which consists in applying an exponential change of measure to the claim sizes distributions that forces their tails to decrease at an exponential rate. This method reduces the “aliasing error” generated by the truncation of the claim sizes distributions and by the “wrap around” effect caused by the discrete Fourier transform (this effect means placing below the truncation point the compound mass which lies beyond this point). For more details on this method and its significance see, e.g., [4], while for its application to model (1) see [9].*

**3.2 A numerical illustration**

In the following, we consider the numerical example presented in [11] to illustrate the application of recursion (3). For the same data, we shall also apply the FFT algorithm and compare the results from both methods. Hence, taking  $m = 3$ , we assume that the distribution of the total number of claims is Poisson, i.e.,  $N \sim Po(\lambda), \lambda > 0$ , for which  $a = 0, b = \lambda$  (see, e.g., [13]), and the p.g.f.  $g_N(t) = e^{\lambda(t-1)}$ ; numerically, we took  $\lambda = 5$ . The claim sizes distributions are the following:

$$f_1 \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0.3 & 0.2 & 0.3 & 0.2 \end{pmatrix}, f_2 \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0.4 & 0.1 & 0.3 & 0.2 \end{pmatrix},$$

$$f_3 \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0.2 & 0.3 & 0.4 & 0.1 \end{pmatrix};$$

$$(f_{\mathbf{L}}(0, i, j))_{i,j=0,1} = \begin{bmatrix} 0.15 & 0.1 \\ 0.05 & 0.2 \end{bmatrix}, (f_{\mathbf{L}}(1, i, j))_{i,j=0,1} = \begin{bmatrix} 0.2 & 0.12 \\ 0.1 & 0.08 \end{bmatrix},$$

while the multinomial parameters values are  $p_0 = 0.25, p_1 = 0.25, p_2 = 0.3, p_3 = 0.2$ . When applying the recursive formulas, the p.f. values  $f_{\mathbf{S}}$  must be computed in a certain order, see [11].

We also applied the FFT algorithm presented above (implemented in Matlab), without using the exponential tilting since our claim sizes distributions are not heavy-tailed. We varied the  $r_j$ s starting with  $r_j = 8, j = \overline{1, 3}$  till  $r_j = 64, j = \overline{1, 3}$ , this last value yielding about the same p.f. as the recursive formula; this can be seen from Table 1, where we displayed the largest absolute error between the exact values (recursive formulas) and the FFT ones evaluated till  $f_{\mathbf{S}}(\mathbf{r} - \mathbf{1}), \mathbf{r} = (r_1, r_2, r_3)$ , i.e.,

$$Err(\mathbf{r}) := \max_{\mathbf{0} \leq \mathbf{x} \leq \mathbf{r} - \mathbf{1}} |f_{\mathbf{S}}(\mathbf{x}) - f_{\mathbf{S}}^{FFT}(\mathbf{x})|.$$

In the same table, we also compared the computing time requested for both techniques used to evaluate the corresponding p.f. till  $\mathbf{r} - \mathbf{1}$ . We note that the more values of  $f_{\mathbf{S}}$  we compute, the better performs the FFT algorithm. As expected, the speed improvement produced by the FFT is obvious when evaluating tail values of  $f_{\mathbf{S}}$ .

**Table 1.** Accuracy and speed (computing time in seconds) of FFT when varying the  $r_j$ s

	$r_j = 8$	$r_j = 16$	$r_j = 32$	$r_j = 64$
Exact	0.0443	0.4974	19.6658	1394.7
FFT	0.0863	0.0953	0.19099	0.8947
$Err(\mathbf{r})$	$0.18 \times 10^{-2}$	$3.04 \times 10^{-6}$	$3.34 \times 10^{-13}$	$5.2 \times 10^{-18}$

In Table 2, we also present some values of  $f_{\mathbf{S}}$ ; the differences between the values obtained by both techniques being so small, we display only the exact values. We can conclude that for these data, it is good enough to take  $r_j = 32, j = \overline{1, 3}$ .

**Table 2.** Some values of  $f_{\mathbf{S}}$

$\mathbf{x}$	(3, 3, 3)	(7, 7, 7)	(10, 10, 10)	(15, 15, 15)	(20, 20, 20)
$f_{\mathbf{S}}(\mathbf{x})$	$3.59 \times 10^{-3}$	$1.47 \times 10^{-5}$	$4.56 \times 10^{-8}$	$3.72 \times 10^{-13}$	$4.62 \times 10^{-19}$

#### 4 Some conclusions and future work

To conclude, two types of techniques can be used to evaluate multivariate compound distributions: exact techniques (like the recursive method presented in Section 2) and approximate ones (like the FFT described in Section 3; we also mention here the simulation method). The choice of the technique should be based first on the existing formulas (i.e., if there is a recursion available, or if it is easy to find one) and also on the purpose of the study. More precisely, from the example above, we conclude that if we need accurate values from the compound distribution's tail, then the FFT method is recommended for its high speed; however, one should be careful with the choice of the optimal values of the truncation points  $r_j$ 's. Another advantage of the FFT is that it is already implemented in Matlab, even for higher dimensions. On the other hand, if one needs very exact values of  $f_{\mathbf{S}}(\mathbf{x})$ , but only for small  $\mathbf{x}$ 's, the recursive method works very good, given that we have recursive formulas for the corresponding multivariate compound distribution.

Therefore, it is important to continue to look for recursions for other types of multivariate compound models. Apart this, another direction for future work that we would like to explore is the simulation technique; it would be interesting to compare its results with the ones from the already studied methods.

**Acknowledgment.** The authors wish to thank the referee for the helpful suggestions, especially on Monte-Carlo simulation.

#### References

- [1] W. L. Briggs, *The DFT: An Owners' Manual for the Discrete Fourier Transform*, Siam, (1995).
- [2] H. Bühlmann, *Numerical evaluation of the compound Poisson distribution: recursion or fast Fourier transform?* *Scand. Actuar. J.*, **1984**(2) (1984), 116-126.

- [3] P. Embrechts, R. Grübel and S. M. Pitts, *Some applications of the fast Fourier transform algorithm in insurance mathematics. This paper is dedicated to Professor WS Jewell on the occasion of his 60th birthday*, Stat. Neerl., **47(1)** (1993), 59-75.
- [4] R. Grubel and R. Hermesmeier, *Computation of compound distributions I: aliasing errors and exponential tilting*, ASTIN Bull., **29(2)** (1999), 197-214.
- [5] T. Jin and J. Ren, *Recursions and fast Fourier transforms for a new bivariate aggregate claims model*, Scand. Actuar. J., **2014(8)** (2014), 729-752.
- [6] N.L. Johnson, S. Kotz and N. Balakrishnan, *Discrete Multivariate Distributions*, Wiley, New York, 1997.
- [7] S.A. Klugman, H.H. Panjer and G.E. Willmot, *Loss Models: From Data to Decisions, 2nd edn.*, Wiley-Interscience, Hoboken, 2004.
- [8] H. H. Panjer, *Recursive evaluation of a family of compound distributions*, ASTIN Bull., **12(1)** (1981), 22-26.
- [9] E.G. Robe-Voinea and R. Vernic, *Fast Fourier Transform for Multivariate aggregate claims*, Comput. Appl. Math., online first, DOI: 10.1007/s40314-016-0336-6 (2016).
- [10] E.G. Robe-Voinea and R. Vernic, *On a multivariate aggregate claims model with multivariate Poisson counting distribution*, to appear in Proceedings of the Romanian Academy - Series A (2015).
- [11] E.G. Robe-Voinea and R. Vernic, *On the recursive evaluation of a certain multivariate compound distribution*, Acta Math. Appl. Sin-E, **32(4)** (2016), 913920.
- [12] B. Sundt, *On multivariate Panjer recursions*, ASTIN Bull., **29(1)** (1999), 29-45.
- [13] B. Sundt and R. Vernic, *Recursions for convolutions and compound distributions with insurance applications*, EAA Lectures Notes, Springer-Verlag Berlin, 2009.

Elena-Gratiela ROBE-VOINEA,  
Faculty of Mathematics and Computer Science,  
University of Bucharest,  
14 Academiei St., 010014, Bucharest, Romania  
E-mail: elena\_robe@yahoo.com

Raluca VERNIC,  
Faculty of Mathematics and Computer Science,  
"Ovidius" University of Constanta,  
124 Mamaia Blvd., 900527 Constanta, Romania  
and Institute for Mathematical Statistics and Applied Mathematics,  
Calea 13 Septembrie 13, 050711 Bucharest, Romania  
Email: rvernic@univ-ovidius.ro

