Wójcik, Rafał; Liu, Charlie Wusuo; Guin, Jayanta

**Article**

# Direct and hierarchical models for aggregating spatially dependent catastrophe risks

Risks

**Provided in Cooperation with:**
MDPI – Multidisciplinary Digital Publishing Institute, Basel

*Suggested Citation:* Wójcik, Rafał; Liu, Charlie Wusuo; Guin, Jayanta (2019) : Direct and hierarchical models for aggregating spatially dependent catastrophe risks, Risks, ISSN 2227-9091, MDPI, Basel, Vol. 7, Iss. 2, pp. 1-22, https://doi.org/10.3390/risks7020054

This Version is available at:
https://hdl.handle.net/10419/257892

risks

MDPI

Article

# Direct and Hierarchical Models for Aggregating Spatially Dependent Catastrophe Risks

**Rafał Wójcik *, Charlie Wusuo Liu and Jayanta Guin**

AIR-Worldwide, Financial Modeling Group, 131 Dartmouth St., Boston, MA 02116, USA; wliu@air-worldwide.com (C.W.L.); jguin@air-worldwide.com (J.G.)

* Correspondence: rwojcik@air-worldwide.com; Tel.: +1-339-545-6203

Received: 31 March 2019; Accepted: 5 May 2019; Published: 8 May 2019


**Abstract:** We present several fast algorithms for computing the distribution of a sum of spatially dependent, discrete random variables to aggregate catastrophe risk. The algorithms are based on direct and hierarchical copula trees. Computing speed comes from the fact that loss aggregation at branching nodes is based on combination of fast approximation to brute-force convolution, arithmetization (regriding) and linear complexity of the method for computing the distribution of comonotonic sum of risks. We discuss the impact of tree topology on the second-order moments and tail statistics of the resulting distribution of the total risk. We test the performance of the presented models by accumulating ground-up loss for 29,000 risks affected by hurricane peril.

**Keywords:** risk aggregation; numerical convolution; copula trees; Monte-Carlo simulation; spatial correlation

## 1. Introduction

The main objective of catastrophe (CAT) modeling is to predict the likelihood, severity and socio-economic consequences of catastrophic events such as hurricanes, earthquakes, pandemias and terrorism. Insurance companies use models to prepare for the financial impact from catastrophic events. Such models offer realistic loss estimates for a wide variety of future scenarios of catastrophes. Losses computed by CAT models can either be deterministic pertaining to a specific historical event (e.g., 2011 Tohoku earthquake in Japan or 2012 Hurricane Sandy in the U.S.), or probabilistic, inferred from an ensemble of hypothetical events (Clark 2015). In the probabilistic framework, large catalogs of events are randomly simulated using Monte-Carlo (MC) methods coupled with physical/conceptual models. For example, based on historical hurricane data, distributions of parameters such as frequencies, intensities, paths etc. are estimated and then used to randomly simulate values of these parameters to obtain a footprint of potential hurricane events in the period of, e.g., 10,000 catalog years. These events are not meant to predict hurricanes from year 2020 to year 12,020, but instead, each of the 10,000 years is considered as a feasible realization of the hurricane activity in the year 2020 (Latchman 2010).

Insurance companies investigate how catalog events affect a portfolio of properties using financial risk analysis (*portfolio rollup*). This operation typically consists of first aggregating event losses per property to obtain the event total loss, and then aggregating event totals within each catalog year to obtain the aggregate annual loss. Typically, the event loss per property, the event total, and the annual loss are all characterized by finite discrete probability distributions. Finally, the mixture of annual loss distributions is used to construct the exceedance probability (EP) curve Grossi et al. (2005). The EP curve is essentially the complementary cumulative distribution function which describes the portfolio loss, and is the key for insurers to estimate the probabilities of experiencing various levels of financial impact.

The first step of risk analysis for large insurance portfolio consists of probabilistic loss aggregation over a large number of locations for each event in the stochastic catalog. In the past few decades a number of methods have been proposed to address this difficult technical issue, see Shevchenko (2010); Wang (1998) for detailed overview. These approaches can roughly be categorized into three mainstream groups: (i) parametric, see e.g., Chaubey et al. (1998); Panjer and Lutek (1983); Venter (2001), where distributions characterizing individual losses belong to some parametric family and their convolution is given by an analytic expression or parametric closed form approximation, (ii) numerical, where individual loss distributions are given in discrete, generally non-parametric form and the distribution of the total risk is obtained using a variant of numerical convolution in Evans and Leemis (2004) boosted by Fast Fourier Transform in Robertson (1992) to make computations tractable (iii) MC, where, for a number of realizations, random samples are drawn from individual loss distributions and simply added up to obtain the aggregate loss, see e.g., Arbenz et al. (2012); Côté and Genest (2015); Galsserman (2004). The approach proposed in this paper originates from the category (ii). This is because of high computing speed requirement for ground-up/gross CAT loss analysis and also due to the fact that risks at different locations and their partial aggregates are described by discrete, generally non-parametric distributions (see Section 2.4 for details). We introduce two major enhancements to loss aggregation via numerical convolution. First, the proposed algorithm operates on irregular positive supports with the size of up to 300 points and treats atoms (point probability masses at minimum and maximum loss) separately. Second, positive correlations between pairs of risks are modeled by a mixture of Split-Atom convolution in Wojcik et al. (2016) and comonotonic distribution (Dhaene et al. 2002) of the sum of risks using several risk aggregation schemes based on copula trees in Arbenz et al. (2012); Côté and Genest (2015). High computing speed of our procedure stems from the fact that, by design, we aim at reproducing only the second order moments of the aggregate risk. Numerical experiments presented in this contribution show, however, that also tail measures of risk compare favorably with the estimates obtained from large sample MC runs.

The paper is organized as follows. First, we introduce the framework for aggregation of spatially dependent risks with copula trees and discuss direct and hierarchical models given positive dependence structure. Next, we present a computationally fast way to sum the dependent risks at branching nodes of a copula tree and algorithms for determining the tree topology. Finally, we show an example of ground-up loss estimation for a historical hurricane event in the U.S. Pros and cons of the proposed aggregation models are discussed and compared to the corresponding MC approach.

## 2. Copula Trees

### *2.1. Problem Statement*

When aggregating CAT risks it is essential to account for spatial dependency between these risks relative to CAT model estimate. In general, combining dependent loss variables requires knowledge of their joint (multivariate) probability distribution. However, the available statistics describing the association between these variables are frequently limited to e.g., correlation matrix Wang (1998). To compute the aggregate loss distribution given such incomplete information, the risks are combined within copula trees (Arbenz et al. 2012; Côté and Genest 2015) where dependencies between losses are captured at each step of the aggregation using copula approach (see, e.g., Cherubini et al. 2004; Nelsen 2006). In the current study, we consider two accumulation schemes which assume non-negative correlations between pairs of risks. The question we attempt to answer is: "What is the most computationally efficient copula tree which aggregates spatially dependent risks pairwise and approximates non-parametric distribution of the sum of losses for a particular CAT event in such a way that its second order moments are reproduced?"

### 2.2. Direct Model

Following the idea in Wang (1998) and Dhaene et al. (2014) the positive dependence between risks can be represented using Fréchet copula via weighting independence and comonotonicity:

$$C_w(\mathbf{u}) = (1-w)\prod_i u_i + w\min(\mathbf{u}) \qquad 0 \le w \le 1, \forall \mathbf{u} \tag{1}$$

where $\mathbf{u} = [u_1, \ldots, u_M] \in [0,1]^M$. For portfolio of $M \geq 2$ risks $\mathbf{X} = [X_1, X_2, \ldots X_M]$, assuming each $X_k$ has finite variance, let $\mathbf{X}^\perp = [X_1^\perp, X_2^\perp, \ldots X_M^\perp]$ and $\mathbf{X}^+ = [X_1^+, X_2^+, \ldots X_M^+]$ be independent and comonotonic random vectors with the same marginals $P_{X_1}, P_{X_2}, \ldots, P_{X_M}$ as $\mathbf{X}$. By definition $P_{\mathbf{X}^\perp}(\mathbf{x}) = \prod P_{X_i}(x_i)$ and $P_{\mathbf{X}^+}(\mathbf{x}) = \min\left(P_{X_1}(x_1), P_{X_2}(x_2), \ldots, P_{X_M}(x_M)\right)$, which is equivalent to the statement that a random vector is either independent or comonotonic if and only if it has either *product* or *min* copula showed as the first and second summand in (1), respectively. For any positive dependent random vector $\mathbf{X}$ (Definition 2 and 3 in Koch and De Schepper 2011), (Collorary 2.3 in Koch and De Schepper 2006) the following bounds hold:

$$P_{\mathbf{X}^\perp}(\mathbf{x}) \le P_{\mathbf{X}}(\mathbf{x}) \le P_{\mathbf{X}^+}(\mathbf{x}) \qquad \forall \mathbf{x} \tag{2}$$

Assuming that the distribution of $\mathbf{X}$ is induced by (1) and reads:

$$P_{\mathbf{X}}(\mathbf{x}) = (1-w)P_{\mathbf{X}^\perp}(\mathbf{x}) + wP_{\mathbf{X}^+}(\mathbf{x}) \qquad 0 \le w \le 1, \forall \mathbf{x} \tag{3}$$

it follows from (Denuit et al. 2001, Theorem 3.1) and (Hürlimann 2001, Remark 2.1) that the dependent sum $S = X_1 + X_2 + \ldots + X_M$ is always bounded in convex order by the corresponding independent sum $S^\perp = X_1^\perp + X_2^\perp + \ldots + X_M^\perp$ and the comonotonic sum $S^+ = X_1^+ + X_2^+ + \ldots + X_M^+$. So,

$$S^\perp \le_{cx} S \le_{cx} S^+ \tag{4}$$

where $\le_{cx}$ denotes the convex order. By definition

$$\mathrm{E}[v(S^\perp)] \le \mathrm{E}[v(S)] \le \mathrm{E}[v(S^+)] \tag{5}$$

for all real convex functions $v$, provided the expectations exist. As a consequence $S^+$ has heavier tails than $S$ and the following variance order holds:

$$\mathrm{Var}[S^\perp] \le \mathrm{Var}[S] \le \mathrm{Var}[S^+] \tag{6}$$

In addition, $\mathrm{E}[S^\perp] = \mathrm{E}[S] = \mathrm{E}[S^+]$ since $\mathbf{X}^\perp, \mathbf{X}$ and $\mathbf{X}^+$ belong to the same Fréchet class Dhaene et al. (2014). To approximate the distribution $P_S$ of the arbitrary sum $S$ we use the weighted average

$$P_S(s) \approx (1-w)P_{S^\perp}(s) + wP_{S^+}(s) \tag{7}$$

where $P_{S^\perp}$ and $P_{S^+}$ are distributions of independent and comonotonic sums $S^\perp$ and $S^+$, respectively. Such ansatz, referred to as the *the mixture method* in Wang (1998), corresponds to the flat aggregation tree in the upper panel of Figure 1. For computational convenience, we elect to approximate the mixing coefficient $w$ as the multivariate dependence measure in Dhaene et al. (2014), so

$$w = \frac{\mathrm{Var}(S) - \mathrm{Var}(S^\perp)}{\mathrm{Var}(S^+) - \mathrm{Var}(S^\perp)} = \frac{\sum_{i\neq j}\mathrm{Cov}(X_i, X_j)}{\sum_{i\neq j}\mathrm{Cov}(X_i^+, X_j^+)} = \frac{\sum_{i\neq j}\mathrm{r}(X_i, X_j)}{\sum_{i\neq j}\mathrm{r}(X_i^+, X_j^+)} \tag{8}$$

where $r$ is the classical Pearson correlation. Since the denominator of (8) is a normalizing constant which depends only on the shape of the marginals, any general correlation matrix $\mathrm{r}(X_i, X_j) = \rho_{i,j}$ with

positive entries as, e.g., shown in Figure 2B,C can be represented by the exchangeable correlation $\rho_e$ in Figure 2A without impacting the value of $w$. That is,

$$\rho_e = \frac{\sum_{i \neq j} \rho_{i,j}}{M(M-1)} \tag{9}$$

Moreover, (8) is equivalent to the comonotonicity coefficient in Koch and De Schepper (2011) if (7) holds for all bivariate marginals:

$$P_{X_i,X_j}(x_i, x_j) = (1-w)P_{X_i^{\perp},X_j^{\perp}}(x_i, x_j) + wP_{X_i^{+},X_j^{+}}(x_i, x_j) \tag{10}$$

We observe that $w = \mathrm{r}(X_i, X_j)$ if and only if $\mathrm{r}(X_i^+, X_j^+) = 1$ which holds when all univariate marginals differ only in location and/or scale parameters (Dhaene et al. 2002).

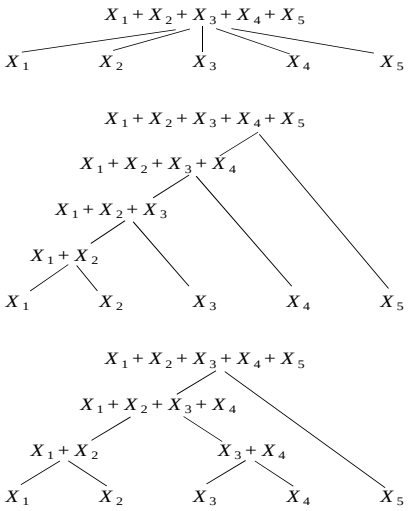



**Figure 1.** Aggregation of five risks using copula trees. Direct model (upper panel), hierarchical model with sequential topology (middle panel) and hierarchical model with closest pair topology (lower panel). The leaf nodes represent the risks whose aggregate we are interested in. The branching nodes of direct tree represent a multivariate copula model for the incoming individual risks while the branching nodes of hierarchical trees represent a bivariate copula model for the incoming pairs of individual and/or cumulative risks.

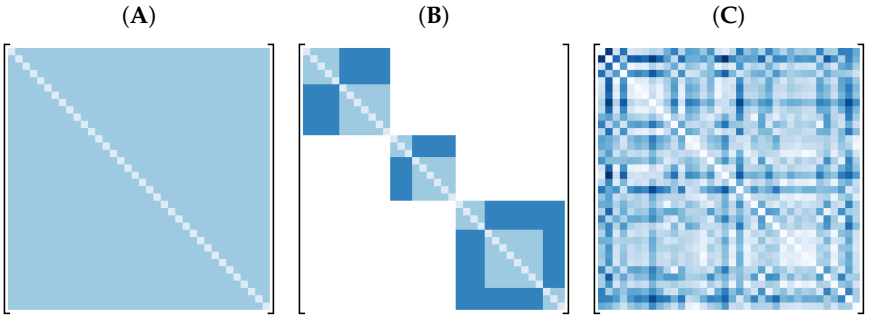



**Figure 2.** Illustration of hypothetical correlation matrices: (**A**) exchangeable, (**B**) nested block diagonal, and (**C**) unstructured correlation matrix.

### *2.3. Hierarchical Model*

If unique description of the joint distribution of individual risks is not crucial and the focus is solely on obtaining an easily interpretable model for the total risk, the individual risks can be

aggregated in a *hierarchical* way. Such process involves specification of partial dependencies between the groups of risks in different aggregation steps (Arbenz et al. 2012). For pairwise accumulation, we first select the two risks $X_i, X_j$ and construct a copula model for that pair. Then, we replace $X_i$ and $X_j$ by their sum $X_i + X_j$ and treat it as a new, combined risk. A simple example is given in the middle panel of Figure 1 depicting the sequential risk aggregation scheme in Côté and Genest (2015). With bivariate (1) inducing the convex sum approximation (7) at the branching nodes we have:

$$\begin{aligned} p_{S_1} = p_{X_1+X_2} &= (1-w_1)\cdot p_{X_1^{\perp}+X_2^{\perp}} + w_1 p_{X_1^{+}+X_2^{+}} \\ p_{S_2} &= (1-w_2)\cdot p_{S_1^{\perp}+X_3^{\perp}} + w_2 \cdot p_{S_1^{+}+X_3^{+}} \\ &\vdots \\ p_{S_4} &= (1-w_4)\cdot p_{S_3^{\perp}+X_5^{\perp}} + w_4 \cdot p_{S_3^{+}+X_5^{+}} \end{aligned} \tag{11}$$

For ease of notation, we dropped the arguments of the probability density functions (pdfs) characterizing the partial sums $S_i = X_1 + \ldots + X_{i+1}$. Observing that for the sequential tree, the partial sums $S_i^{\perp}$ and $S_i^{+}$ are symbolic abbreviations of $(X_1 + \ldots + X_{i+1})^{\perp}$ and $(X_1 + \ldots + X_{i+1})^{+}$ as opposed to $X_1^{\perp} + \ldots + X_{i+1}^{\perp}$ and $X_1^{+} + \ldots + X_{i+1}^{+}$ in (4), the partial weights $w_i$ read:

$$w_i = \frac{\mathrm{Var}[S_i] - \mathrm{Var}[S_i^{\perp}]}{\mathrm{Var}[S_i^{+}] - \mathrm{Var}[S_i^{\perp}]} \tag{12}$$

We remark, that if the order in which the aggregation is performed does not trivially follow from the tree structure, any convention can be used to make the numbering of partial sums unique, e.g., see Côté and Genest (2015). However, this has no implication on the fact that, in general, hierarchical trees do not uniquely determine the joint distribution of risk $\mathbf{X}$. To answer the research question posed in Section 2.1, this non-uniqueness is not critical. Conversely, in situations where, e.g., capital allocation is of interest (see Côté and Genest 2015), an extra conditional independence assumption in Arbenz et al. (2012) is needed. For instance, the aggregation scheme in the middle panel of Figure 1 would require:

$$\begin{aligned} &(X_1, X_2, S_1) \perp (X_3, X_4, X_5, S_2, S_3, S_4) | S_1 \\ &(X_1, X_2, X_3, S_1, S_2) \perp (X_4, X_5, S_3, S_4) | S_2 \\ &(X_1, X_2, X_3, X_4, S_1, S_2, S_3) \perp (X_5, S_4) | S_3 \end{aligned} \tag{13}$$

### *2.4. Implementation of Risk Aggregation at Branching Nodes*

A sample reordering method, inspired from Iman and Conover (1982) and assembled into the customized procedure in Arbenz et al. (2012), has recently been used to facilitate summation of risks in both the direct and hierarchical models. Despite the elegant simplicity of this approach, it comes at high computational cost for large samples. To reduce that cost and to orchestrate reproduction of the second order moments of the target sum $S$, we opt to use the following set of algorithms instead: (i) second order approximation to brute force convolution referred to as the Split-Atom convolution Wojcik et al. (2016), (ii) arithmetization (aka regriding) of loss distributions Vilar (2000), (iii) estimation of the distribution of the comonotonic sum of risks (comonotonization) and (iv) construction of the mixture distribution in (7). These algorithms are described in Sections 2.4.1–2.4.3. An individual risk $X$ is a discrete random variable expressed in terms of the damage ratio defined as loss divided by replacement value. The corresponding pdf $p_X$ is represented by zero-and-one inflated mixture:

$$p_X(x) = \begin{cases} \alpha, & \text{if} \quad x = 0 \\ (1-\alpha-\beta)\hat{p}_X(x), & \text{if} \quad 0 < x < 1 \\ \beta, & \text{if} \quad x = 1 \end{cases} \tag{14}$$

where $\alpha$ and $\beta$ are atoms at zero and one damage ratio, respectively, and $\hat{p}_X$ is the (discretized) pdf describing the main part of the mixture, see Figure 3 for an example and Section 3 for parameterization used in this study. The atoms are a common feature inferred from analysis of CAT insurance claims data. They also emerge during gross loss portfolio roll-up as a result of applying stop-loss insurance and/or re-insurance terms—deductibles and limits assembled into a variety of tiers/structures.

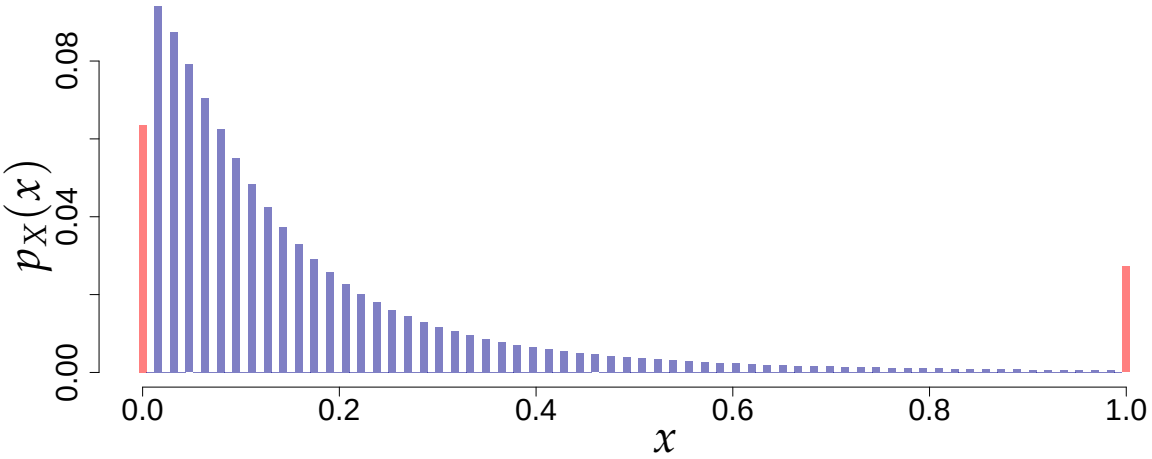



**Figure 3.** A discrete loss pdf represented as a mixture of two "spikes" (atoms) at minimum and maximum x damage ratio (red) and the main part (blue). Damage ratio is discretized on 64-point grid.

#### 2.4.1. Split-Atom Convolution

The pdf $p_{S^\perp}$ of a sum of two independent discrete random variables $X$ and $Y$ with pdfs $p_X$ and $p_Y$ respectively, can be computed as:

$$p_{S^\perp}(s) = p_X \oplus p_Y = \sum_x p_X(x) p_Y(s-x) \tag{15}$$

The classical way of implementing (15) for random variables defined on irregular support is brute force (BF) convolution in Evans and Leemis (2004). Let $N_x$ and $N_y$ be the number of points discretizing $p_X$ and $p_Y$ support, respectively. The complexity of BF convolution is $O\big(N_x N_y log(N_x N_y)\big)$ because computing all possible products of probabilities and all possible sums of losses is $O(N_x N_y)$, and redundancy removal is $O(N_x N_y log(N_x N_y))$, see Evans and Leemis (2004) for details. Such high computational demand makes the BF algorithm impractical for convolving large number of CAT loss distributions. Our solution in Wojcik et al. (2016), referred to as the Split-Atom convolution, reduces the computational cost to $O(N'_x N'_y)$ where $N'_x < N_x$ and/or $N'_y < N_y$. The idea is to separate two atoms ("spikes") at min/max losses and arithmetize (regrid) the main parts of loss distributions. The regriding is the key to accelerating computations. The Split-Atom approach preserves min/max losses from different loss perspective (e.g., insurer, re-insurer, insured, FAC underwriter etc.) and enhances the accuracy of $p_S$ reproduction in situations where substantial probability mass is concentrated in the atoms. This is relevant for ground-up and, even more so, for gross loss estimation when stop-loss insurance is applied. Various ways of splitting the atoms and compacting main parts of loss distributions can be considered depending on computing speed and memory requirements and organization/arrangement of the original and convolution grids. An example is given in Algorithm 1. The original grids are non-uniform: the main parts of $p_X$ and $p_Y$ are defined on regular grids with spans $h_x$ and $h_y$, but the spacing between atoms and the main parts is arbitrary. Convolution grid is designed to have the same irregular spacing for preserving min/max losses. The speedup comes from execution of step 28 in Algorithm 1 using Algorithm 2. Depending on the application, other variants of the Split-Atom convolution are conceivable. For example, the 9-products approach in Algorithm 1 not only reproduces min/max losses but also probabilities describing these losses, or the

4-products approach which only splits the right atom to gain extra computing speed. See Appendix A, Algorithm A1.

**Algorithm 1:** Split-Atom Convolution: 9-products

**Input** : Two discrete pdfs $p_X$ and $p_Y$ with supports:
$x = \{x_1, x_2, x_2 + h_x, x_2 + 2h_x, \ldots, x_2 + (N_x - 3)h_x, x_{N_x}\}$;
$y = \{y_1, y_2, y_2 + h_y, y_2 + 2h_y, \ldots, y_2 + (N_y - 3)h_y, y_{N_y}\}$;
and probabilities: $p_X(x) = \sum_{i=1}^{N_x} \delta(x - x_i)p_X(x_i)$; $p_Y(y) = \sum_{i=1}^{N_y} \delta(y - y_i)p_Y(y_i)$
$N_s$ - maximum number of points for discretizing convolution grid

$x^{(1)} = \{x_1\}$; $p_X^{(1)}(x) = \delta(x - x^{(1)})p_X(x_1)$// split the left atom of $p_X$
$x^{(2)} = \{x_{N_x}\}$; $p_X^{(2)}(x) = \delta(x - x_i^{(2)}) \cdot p_X^{(2)}(x_i^{(2)})$// split the right atom of $p_X$
$x^{(3)} = \{x_2, x_2 + h_x, x_2 + 2h_x, \ldots, x_2 + (N_x - 3)h_x\}$; $p_X^{(3)}(x) = \sum_{i=1}^{N_x - 2} \delta(x - x_i^{(3)})p_X^{(3)}(x_i^{(3)})$
$y^{(1)} = \{y_1\}$; $p_Y^{(1)}(y) = \delta(y - y^{(1)})p_Y(y_1)$// split the left atom of $p_Y$
$y^{(2)} = \{y_{N_y}\}$; $p_Y^{(2)}(y) = \delta(y - y_i^{(2)}) \cdot p_Y^{(2)}(y_i^{(2)})$// split the right atom of $p_Y$
$y^{(3)} = \{y_2, y_2 + h_y, x_2 + 2h_y, \ldots, y_2 + (N_y - 3)h_y\}$; $p_Y^{(3)}(y) = \sum_{i=1}^{N_y - 2} \delta(y - y_i^{(3)})p_Y^{(3)}(y_i^{(3)})$
$h_{s^\star} = \frac{x_{N_X} + y_{N_Y} - (x_1 + y_1)}{N_{s^\star} - 1}$// corresponding step size of the main part of convolution grid $s$
$h_s = \max(h_x, h_y, h_{s^\star})$// final step size of the main part of convolution grid $s$
$N_s = \lfloor \frac{x_{N_X} + y_{N_Y} - (x_1 + y_1)}{h_s} \rfloor$ // corresponding number of points
// set irregular convolution grid:
**if** $h_s \geq (x_2 + y_2 - x_1 - y_1)$ **then**
    $s = \{x_1 + y_1, x_2 + y_2, x_2 + y_2 + h_s, \ldots, x_2 + y_2 + (N_s - 3)h_s, x_{N_x} + y_{N_y}\}$
**else**
    $s = \{x_1 + y_1, x_2 + y_2 - h_s, x_2 + y_2, \ldots, x_2 + y_2 + (N_s - 4)h_s, x_{N_x} + y_{N_y}\}$
**end if**
$x^{(3)'} = \{x_2, x_2 + h_s, x_2 + 2h_s, \ldots, x_2 + (N_x - 3)h_s\}$// discretize $x^{(3)}$ with $h_s$
$p_X^{(3)'}(x) = \sum_{i=1}^{N'_x} \delta(x - x_i^{(3)'})p_X^{(3)'}(x_i^{(3)'})$// regrid $p_X^{(3)}$
$y^{(3)'} = \{y_2, y_2 + h_s, y_2 + 2h_s, \ldots, y_2 + (N_y - 3)h_s\}$// discretize $y^{(3)}$ with $h_s$
$p_Y^{(3)'}(y) = \sum_{i=1}^{N'_y} \delta(y - y_i^{(3)'})p_Y^{(3)'}(y_i^{(3)'})$// regrid $p_Y^{(3)}$
$\mathcal{B}^{(1)} = p_X^{(1)} \oplus p_Y^{(1)}$// Brute Force convolution
$\mathcal{B}^{(2)} = p_X^{(2)} \oplus p_Y^{(1)}$// --''--
$\mathcal{B}^{(3)} = p_X^{(3)'} \oplus p_Y^{(1)}$// --''--
$\mathcal{B}^{(4)} = p_X^{(1)} \oplus p_Y^{(2)}$// --''--
$\mathcal{B}^{(5)} = p_X^{(2)} \oplus p_Y^{(2)}$// --''--
$\mathcal{B}^{(6)} = p_X^{(3)'} \oplus p_Y^{(2)}$// --''--
$\mathcal{B}^{(7)} = p_X^{(1)} \oplus p_Y^{(3)'}$// --''--
$\mathcal{B}^{(8)} = p_X^{(2)} \oplus p_Y^{(3)'}$// --''--
$\mathcal{B}^{(9)} = p_X^{(3)'} \oplus p_X^{(3)'}$// --''--
Regrid $\mathcal{B}^{(1-9)}$ onto convolution grid $s$

**Output** : Discrete probability density function $p_{S^\perp}$ of the independent sum $S^\perp = X + Y$ with the support $s = \{s_1, s_2, s_2 + h_s, s_2 + 2h_s, \ldots, s_2 + (N_s - 3)h_s, s_{N_s}\}$ and the associated probabilities $p_{S^\perp}(s) = \sum_{k=1}^{N_s} \delta(s - s_k)p_{S^\perp}(s_k)$, where $s_{N_s} - [s_2 + (N_s - 3)h_s] \leq h_s$, $s_2 - s_1 \leq h_s$, $h_s \geq \max(h_1, h_2)$.

**Algorithm 2:** Brute force convolution for supports with the same span

**Input** : Two discrete probability density functions $p_X$ and $p_Y$, where the supports of $X$ and $Y$ are defined using the same span $h$ as: $x = \{x_1, x_1 + h, \ldots, x_1 + (N_x - 1)h\}$, $y = \{y_1, y_1 + h, \ldots, y_1 + (N_y - 1)h\}$ and the associated probabilities as $p_X(x) = \sum_{i=1}^{N_x} \delta(x - x_i) p_X(x_i)$, $p_Y(y) = \sum_{j=1}^{N_y} \delta(y - y_i) p_Y(y_j)$

1: $s = \{x_1 + y_1, x_1 + y_1 + h, x_1 + y_1 + 2h, \ldots, x_1 + y_1 + (N_x + N_y - 2)h\}$ `// compute convolution support`

2: $p_{S^\perp}(s) = \mathbf{0}$ `// initialize`

3: $p_{S^\perp}(s_{i+j-1}) \leftarrow p_{S^\perp}(s_{i+j-1}) + p_X(x_i) p_Y(y_j) \quad 1 \le i \le N_x, 1 \le j \le N_y$ `// compute probabilities`

**Output** : Discrete probability density function $p_{S^\perp}$ of the independent sum $S^\perp = X + Y$ with the support $s = \{s_1, s_2, \ldots, s_{N_x+N_y-1}\}$ and the corresponding probabilities $p_{S^\perp}(s) = \sum_{k=1}^{N_x+N_y-1} \delta(s_k) p_{S^\perp}(s_k)$.

Further acceleration of Algorithm 1 can be achieved using Fast Fourier Transform (FFT) and applying convolution theorem, (Wang 1998, Section 3.3.1) to convolve the main parts of $p_X$ and $p_Y$. Therefore, step 28 in Algorithm 1, is replaced with the following product:

$$p_X^{(3)'} \oplus p_Y^{(3)'} \overset{FFT}{\rightarrow} P_{\Omega_X}^{(3)'} P_{\Omega_Y}^{(3)'} \overset{IFFT}{\rightarrow} p_{X+Y}^{(3,3)'} \rightarrow \mathcal{B}^{(9)} \tag{16}$$

where $P_{\Omega_1}^{(3)'}$ and $P_{\Omega_2}^{(3)'}$ represent FFTs of the probabilities $p_X^{(3)'}$ and $p_Y^{(3)'}$. IFFT in (16) stands for the Inverse Fast Fourier Transform. Implementation of (16) requires the supports $x^{(3)'}$ and $y^{(3)'}$ have the same span $h_s$, which is guaranteed by steps 16 and 18 in Algorithm 1, and also have the same range containing power-of-2 numbers of points, which can be guaranteed by extending one or both supports and padding zero probabilities.

#### 2.4.2. Regriding

Split-Atom convolution requires that two pdfs being convolved have the main part of their supports discretized using the common span (grid step size) $h_s$, see step 8 in Algorithm 1. Prior to convolution, either one or both of these pdfs may need arithmetization hereafter referred to as the regriding. In general, this operation takes the discrete pdf $p_X$ defined on fine scale support with the span $h$ and determines the new arithmetic pdf $p_{X'}$ on coarse scale support with the span $h' > h$ with the property $\sum_x x^m p_X(x) = \sum_{x'} {x'}^m p_{X'}(x')$ of equating $m$ moments with $p_X$ (see, e.g., Gerber 1982; Vilar 2000; Walhin and Paris 1998).

For $m = 1$, *linear regriding* or mass dispersal in Gerber (1982) redistributes a probability mass on the original grid to two neighboring points on the new grid such that only $\mathrm{E}[X]$ is preserved. This is achieved by solving a $2 \times 2$ local linear system, see Algorithm 3, Step 6 and animation therein. For $m = 2$, the standard approach is to apply the method of *local moment matching* (LMM) which reproduces local second order moments of $p_X$ in predefined intervals on the new grid, assuring preservation of both $\mathrm{E}[X]$ and $\mathrm{Var}[X]$ (Gerber 1982; Vilar 2000; Walhin and Paris 1998). Despite technical simplicity of LMM, there are two caveats: "wiggly" behavior and/or negative probabilities in $p_{X'}$ (see, e.g., Table 4 in Walhin and Paris 1998). The first one is due to the fact that LMM performs local moment matching in fixed intervals $(x_k, x_k + 2h'], (x_k + 2h', x_k + 4h'], (x_k + 4h', x_k + 6h'] \ldots$, etc. but ignores matching the moments in $(x_k + h', x_k + 3h'], (x_k + 3h', x_k + 5h'], (x_k + 5h', x_k + 7h'] \ldots$, etc. A simple improvement is proposed in Appendix A, Algorithm A2. The second caveat is that solving the $3 \times 3$ local linear system for matching the local moments (Equation (16) in Gerber 1982) guarantees a negative in the three dispersed probability masses. The negative mass can only be balanced out by a positive one dispersed from solving the next local linear system if the positive mass is greater than the negative one. Upon completion of the LMM algorithm, negative probabilities could still exist at arbitrary locations on the new grid. One way to handle this issue is to retreat to linear regriding as

in Panjer and Lutek (1983). Another way is to use linear goal programming in Vilar (2000), i.e., for every span $h'$ determine the arithmetic pdf which conserves the first moment $\mathrm{E}[X]$ and has the nearest second moment to the target $\mathrm{Var}[X]$. Therefore, the negative mass is mitigated by sacrificing the quality of reproducing $\mathrm{Var}[X]$. Here, we propose an alternative two-stage strategy listed in Algorithms 4 and 5 hereafter referred to as the *4-point regriding*.

**Algorithm 3:** Linear regriding

**Input :** Discrete pdf $p_X$ with fine scale support $x = \{x_1, x_1 + h, \ldots, x_1 + (N-1)h\}$ and associated probabilities $p_X(x) = \sum_{i=1}^{N} \delta(x - x_1 - (i-1)h) \cdot p_X(x_1 + (i-1)h)$; the support of coarse scale discrete pdf $p_{X'}$: $x' = \{x'_1, x'_1 + h', \ldots, x'_1 + (N'-1)h'\}$.
**Requirement:** $x'_1 \le x_1$, $x'_{N'} \ge x_N$, $h < h'$.

1: For convenience, let $\{x_1, x_2, \ldots, x_N\} \leftarrow x$ and $\{x'_1, x'_2, \ldots, x'_N\} \leftarrow x'$
2: Initialize $p_{X'}(x') \leftarrow \mathbf{0}$
3: **if** $x'_1 = x_1$ **then**
4: $\quad p_{X'}(x'_1) \leftarrow p_{X'}(x'_1) + p_X(x_1)$
5: **end if**
6: **for** $i < N'$, $j \le N$, $x'_i < x_j \le x'_{i+1}$ **do**
7:
$$\begin{bmatrix} p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \end{bmatrix} \leftarrow \begin{bmatrix} p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \end{bmatrix} + \begin{bmatrix} 1 & 1 \\ x'_i & x'_{i+1} \end{bmatrix}^{-1} \begin{bmatrix} p_X(x_j) \\ x_j p_X(x_j) \end{bmatrix}$$
8: **end for**

**Output:** Discrete pdf $p_{X'}(x')$

**Example 1**. Linear regriding of fine scale pdf (blue bars) onto new grid (orange circles). The resulting coarse scale pdf is represented as green bars.

In Stage I, a probability mass is dispersed to the two neighboring points and the two end points (first and last point on the new grid) such that the mean and the second moment are preserved. The local linear system for matching moments (Algorithm 4, Step 15) guarantees (i) positive masses added to the neighboring points and negative masses added to the end points, (ii) probabilities added to the neighboring points are inversely proportional to distances between these neighboring points and the projecting point. Property (ii) mimics linear regriding and completes a $4 \times 4$ invertible local linear system along with the other three constraints for matching moments. Negative probabilities resulted from solving the system are typically small in absolute value because the end points are usually far from the projecting point. In practice, Stage I rarely leaves negative probabilities at one or both endpoints of the new grid on completion of the algorithm.

If this is not the case, in Stage II, the negative probability at the first (last) point is dispersed to the second (second to last), the third (third to last), and the last (first) point, respectively, subject to second order moment matching, see Steps 6 and 16 in Algorithm 5. The local linear system in step 7 (or 17) guarantees positive mass dispersed to the third (third to last) point and negative masses to the first (last) point and to the second (second to last) point respectively, see animation in Algorithm 5. Probability at the first (last) point is then set to zero and the second (second to last) point becomes the first (last) point on the new grid, reducing min/max range of $p_{X'}$ support. The algorithm alternates between the first and the last points until both hold nonnegative probabilities, or, until no more points are available for balancing out the negative probabilities (see the animation in Example 3, Algorithm 5). The latter indicates 4-point regriding failure and invokes linear regriding.

**Algorithm 4:** 4-point regridding, Stage I

**Input :** Discrete pdf $p_X$ with fine scale support $x = \{x_1, x_1 + h, \ldots, x_1 + (N-1)h\}$ and associated probabilities $p_X(x) = \sum_{i=1}^{N} \delta(x - x_1 - (i-1)h) \cdot p_X(x_i + (i-1)h)$; the support of coarse scale probability mass function $p_{X'}$: $x' = \{x'_1, x'_1 + h', \ldots, x'_1 + (N'-1)h'\}$.
**Requirement:** $x'_1 \leq x_1$, $x'_{N'} \geq x_N$, $h < h'$.

For convenience, let $\{x_1, x_2, \ldots, x_N\} \leftarrow x$ and $\{x'_1, x'_2, \ldots, x'_{N'}\} \leftarrow x'$.
**if** $\min(N, N') < 5$ **then**
  Go to Algorithm 3.
**end if**
$p_{X'}(x') \leftarrow \mathbf{0}$; $i \leftarrow 1$; $j \leftarrow 1$; $i^* \leftarrow N'$
**if** $x'_1 = x_1$ **then**
  $p_{X'}(x'_1) \leftarrow p_X(x_1)$; $i \leftarrow i + 1$
  **if** $x_2 > x'_2$ **then**
    $j \leftarrow j + 1$
  **end if**
**end if**
**if** $x'_{N'} = x_N$ **then**
  $p_{X'}(x'_{N'}) \leftarrow p_X(x_N)$; $i^* \leftarrow i^* - 1$
**end if**
**for** $i < i^*$, $j \leq N$, $x'_i < x_j \leq x'_{i+1}$ **do**

$$\begin{bmatrix} p_{X'}(x'_1) \\ p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \\ p_{X'}(x'_{N'}) \end{bmatrix} \leftarrow \begin{bmatrix} p_{X'}(x'_1) \\ p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \\ p_{X'}(x'_{N'}) \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 & 1 \\ x'_1 & x'_i & x'_{i+1} & x'_{N'} \\ x'^2_1 & x'^2_i & x'^2_{i+1} & x'^2_{N'} \\ 0 & x_j - x'_i & x_j - x'_{i+1} & 0 \end{bmatrix}^{-1} \begin{bmatrix} p_X(x_j) \\ x_j p_X(x_j) \\ x_j^2 p_X(x_j) \\ 0 \end{bmatrix}$$

**end for**
**if** $\min\left(p_{X'}(x'_1),\ p_{X'}(x'_{N'})\right) < 0$ **then**
  Go to Algorithm 5.
**end if**

**Output:** Discrete pdf $p_{X'}(x')$.

**Example 2**. 4-point regriding of fine scale pdf (blue bars) onto new grid (orange circles). The resulting coarse scale pdf is represented as green bars.

**Algorithm 5:** 4-point regridding, Stage II

**Input :** Discrete pdf $p_{X'}(x')$ from Algorithm 4; $\min\left(p_{X'}(x'_1),\ p_{X'}(x'_{N'})\right) < 0$.

$u \leftarrow 1$; $t \leftarrow N'$; $B \leftarrow$FALSE; $\triangleleft$ $B =$TRUE implies negative probabilities reconciliation fails.
**loop**
  **if** $t - u < 3$ **then**
    $B \leftarrow$TRUE; **break**
  **end if**
  **if** $p_{X'}(x'_u) < 0$ **then**

$$\begin{bmatrix} p_{X'}(x'_{u+1}) \\ p_{X'}(x'_{u+2}) \\ p_{X'}(x'_t) \end{bmatrix} \leftarrow \begin{bmatrix} p_{X'}(x'_{u+1}) \\ p_{X'}(x'_{u+2}) \\ p_{X'}(x'_t) \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ x'_{u+1} & x'_{u+2} & x'_t \\ x'^2_{u+1} & x'^2_{u+2} & x'^2_t \end{bmatrix}^{-1} \begin{bmatrix} p_{X'}(x'_u) \\ x_h p_{X'}(x'_u) \\ x_h^2 p_{X'}(x'_u) \end{bmatrix}$$

    $p_{X'}(x'_u) \leftarrow 0$; $u \leftarrow u + 1$
  **end if**
  **if** $\min\left(p_{X'}(x'_u),\ p_{X'}(x'_t)\right) \geq 0$ **then**
    **break**
  **end if**
  **if** $t - u < 3$ **then**
    $B \leftarrow$TRUE; **break**
  **end if**
  **if** $p_{X'}(x'_t) < 0$ **then**

**Example 3**. Mitigation of negative probabilities (red bars) in 4-point regriding.

$$\begin{bmatrix} p_{X'}(x'_{t-1}) \\ p_{X'}(x'_{t-2}) \\ p_{X'}(x'_u) \end{bmatrix} \leftarrow \begin{bmatrix} p_{X'}(x'_{t-1}) \\ p_{X'}(x'_{t-2}) \\ p_{X'}(x'_u) \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ x'_{t-1} & x'_{t-2} & x'_u \\ x'^2_{t-1} & x'^2_{t-2} & x'^2_u \end{bmatrix}^{-1} \begin{bmatrix} p_{X'}(x'_t) \\ x_t p_{X'}(x'_t) \\ x_t^2 p_{X'}(x'_t) \end{bmatrix}$$

    $p_{X'}(x'_t) \leftarrow 0$; $t \leftarrow t - 1$
  **end if**
  **if** $\min\left(p_{X'}(x'_u),\ p_{X'}(x'_t)\right) \geq 0$ **then**
    **break**
  **end if**
**end loop**
**if** $B =$TRUE **then**
  Go to Algorithm 3.
**end if**

**Output:** Discrete pdf $p_{X'}(x')$.

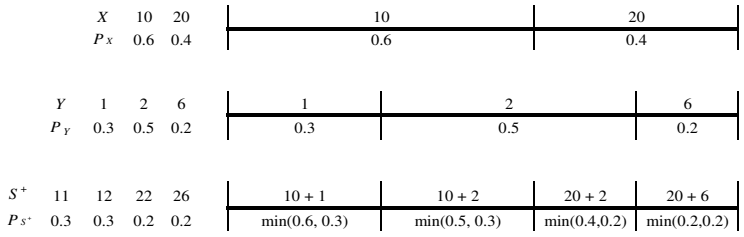
### 2.4.3. Comonotonization and Mixture Approximation

The basic construction idea for $p_S^+$ is listed in Algorithm 6 and illustrated by Example 3. The method proceeds recursively, defining the next element of the new distribution $p_S^+$ based on the first elements of $p_X$ and $p_Y$ then modifying $p_X$ and $p_Y$, respectively. It requires $O(N_x + N_y)$ operations.

**Algorithm 6:** Distribution of the comonotonic sum

**Input :** Two discrete probability density functions $p_X$ and $p_Y$, with irregular positive supports $x = \{x_1, x_2, \ldots, x_{N_x-1}, x_{N_x}\}$; $y = \{y_1, y_2, \ldots, y_{N_y-1}, y_{N_y}\}$ and associated probabilities $p_X(x) = \sum_{i=1}^{N_x} \delta(x - x_i) p_X(x_i)$; $p_Y(y) = \sum_{i=1}^{N_y} \delta(y - y_i) p_Y(y_i)$

$i_x \leftarrow 1, i_y \leftarrow 1, i_s \leftarrow 1$
**loop**
  $s^+_{i_s} \leftarrow x_{i_x} + y_{i_y}$
  $p \leftarrow p_X(x_{i_x}) - p_Y(x_{i_y})$
  **if** $p > 0$ **then**
    $p_{S^+}(s^+_{i_s}) \leftarrow p_Y(y_{i_y})$
    $p_X(x_{i_x}) \leftarrow p$
    $i_y \leftarrow i_y + 1$
    **if** $i_y > N_y$ **then**
      **break**
    **end if**
  **else**
    $p_{S^+}(s^+_{i_s}) \leftarrow p_X(x_{i_x})$
    $p_X(x_{i_x}) \leftarrow -p$
    $i_x \leftarrow i_x + 1$
    **if** $i_x > N_x$ **then**
      **break**
    **end if**
  **end if**
  $i_s \leftarrow i_s + 1$
**end loop**




**Example 3**: To visualize the algorithm on the right, consider two pdfs $p_X$ and $p_Y$ listed in the left panel of the above cartoon. These pdfs are represented as subsegments of the 0-1 probability line. Quantiles of the same order are added and recursively registered as points on the $S^+$ support. At the start of the algorithm, we (i) subtract the smaller of $\{p_X(x_1), p_Y(y_1)\}$ from the larger, (ii) assign the smaller to $p_{S^+}(s^+_1)$, (iii) replace the larger with the residual, and (iv) zero the smaller. Next, we move on to the first *nonzero* probabilities in $p_X$ and $p_Y$, and repeat the above procedure for computing $p_{S^+}(s^+_2)$. The algorithm ends once the probabilities in both $p_X$ and $p_Y$ are exhausted.

**Output:** Discrete probability density function $p_{S^+}$ of the comonotonic sum $S^+ = X + Y$ satisfying $P^{-1}_{S^+}(u) = P^{-1}_X(u) + P^{-1}_Y(u)$, where $u \sim U(0,1)$ and $P^{-1}_X$, $P^{-1}_Y$ are the inverse cumulative distribution functions of $X$ and $Y$ respectively. The support of $p_{S^+}$ is defined as $s^+ = \{s^+_1, s^+_2, \ldots^+, s^+_{N_s-1}, s^+_{N_s}\}$ and associated probabilities as $p_{S^+}(s^+) = \sum_{i=1}^{N_s} \delta(s^+ - s^+_i) p_{S^+}(s^+_i)$ where $N_s \leq N_x + N_y$.

Once $p_{S^\perp}$, $p_{S^+}$ and $w$ are known, the mixture distribution $p_S$ is composed using (11) . In general, supports of $p_{S^\perp}$ and $p_{S^+}$ cover the same range but are defined on different individual grids, so arithmetizing on common grid is needed. If 4-point regriding is used, the target $\mathrm{E}[S]$ is preserved exactly and the target $\mathrm{Var}[S]$ is preserved with small error due to occasional retreats to linear regriding.

### *2.5. Order of Convolutions and Tree Topology*

Following Wojcik et al. (2016), to further minimize this error, convolutions should be ordered to assure that two distributions undergoing convolution have supports covering approximately same domains ($x_{N_x} - x_1 \approx y_{N_y} - y_1$) with the same span ($h_x \approx h_y$) in Algorithm 1. Since convolution is computationally more expensive than comonotonization, we assume that the order of convolutions governs the order of comonotonizations. Therefore, topology of a particular aggregation tree should be determined by the order of convolutions. For example, *ascending order* arranges risks at leaf nodes from smallest to largest maxima prior to aggregation, and then accumulates the risks using closest pair strategy depicted in the bottom panel of Figure 1. More sophisticated risk sorting strategy originates from the balanced multi–way number partitioning and is referred to as the balanced largest–first differencing method in Zhang et al. (2011). Here, one seeks to split a collection of numbers (risk maxima) into subsets with (roughly) the same cardinality and subset sum. In general, sorting based convolution orders are useful where no specific way to group risks at leaf nodes of aggregation tree is of importance for total risk analysis. When the goal is to assess the impact of spatial dependencies among elements within a CAT risk portfolio on aggregate loss, geographical grouping of properties (locations) affected by a particular CAT event is crucial (Einarsson et al. 2016). To account for such

patterns we propose to use the *recursive nearest neighbor order* (RNN) of convolutions where depth-first search in Cormen et al. (2009) is performed to define an aggregation tree, see example in Figure 4. In contrast with sorting based strategies, the RNN keeps the original (predefined) ordering of risks at the leaf nodes of the aggregation tree intact.

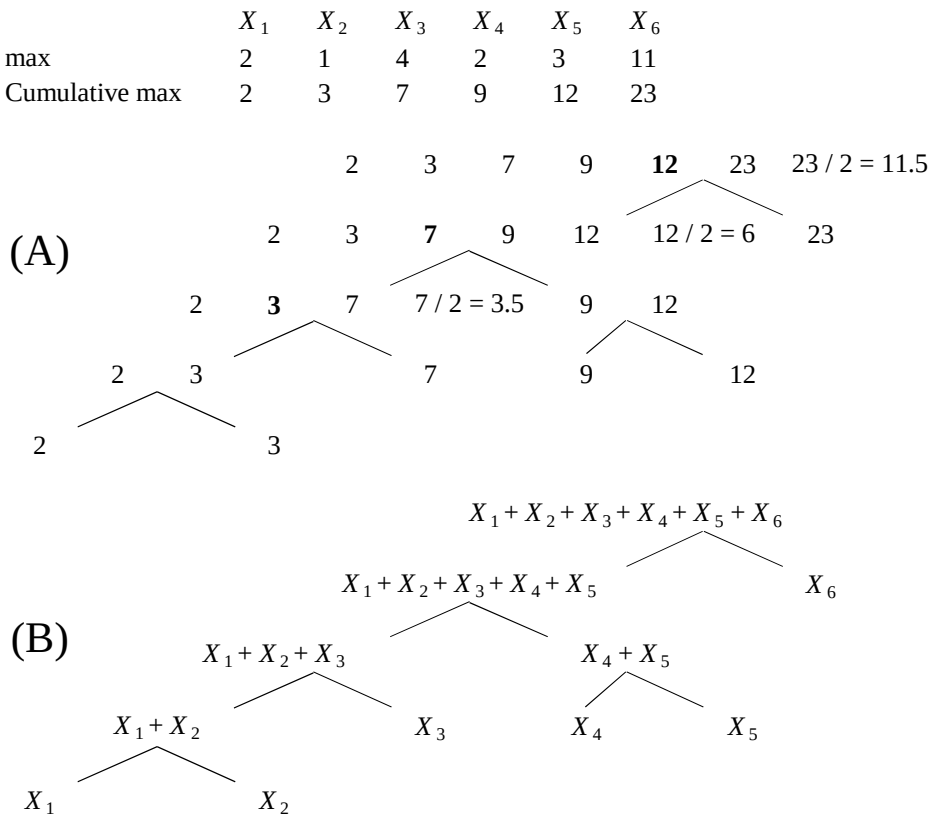



**Figure 4.** An example of RNN approach for determining topology of hierarchical risk aggregation tree for six risks with zero minima. The maxima and cumulative maxima characterizing losses for the six risks are presented in the upper panel. (**A**) The algorithm takes the largest cumulative max and halves it to obtain the number $c$. Then, it binary searches for the number closest to $c$ except for the last element in the sequence. This number (showed in bold) becomes the cumulative maximum of the new subsequence. The search is repeated until the subsequence consists of two elements. (**B**) The resulting hierarchical aggregation tree.

It must be mentioned that order of convolutions is implicitly affecting the topology of the direct aggregation model in (7). This is shown in Figure 5 where the convex sum approximation at the root node is composed of aggregates constructed using, e.g., the RNN approach in Figure 4.

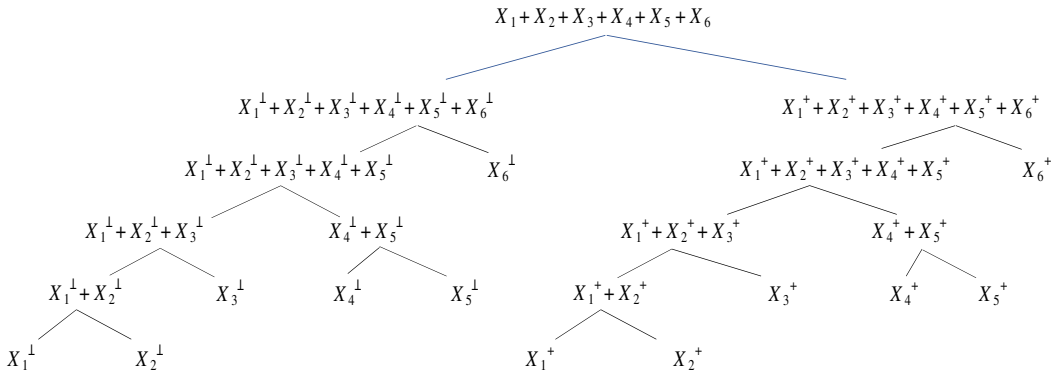



**Figure 5.** An example of recursive nearest neighbor (RNN) approach for determining topology of direct risk aggregation tree for six risks shown in Figure 4. Note that the order of comonotonic aggregation follows the order of independent aggregation.

## 3. Results

As an example of loss analysis, we estimated ground-up loss for a major hurricane event in the US affecting 29,139 locations from portfolio of an undisclosed insurance company. Loss $X_k$ at $k$th location is described by inflated transformed beta pdf, i.e., the main part in (14) is parameterized using transformed beta in Venter (2013). We assume that the mean of each pdf is a function of CAT model prediction and that covariance between risks is a function of model error, so

$$\begin{aligned} X_k &\sim InTrBeta(\boldsymbol{\Theta}_k) \\ \mathrm{E}(X_k) &= g(I_k) \\ \mathrm{Cov}(X_i, X_j) &= \Sigma_{i,j} = \mathrm{E}[(X_i - g(I_i))(X_j - g(I_j))] \end{aligned} \tag{17}$$

where $\boldsymbol{\Theta}_k$ is the parameter vector, $I_k$ stands for hurricane peril intensity end $g$ is the damage function which transforms intensity into damage ratio. The intensity is expressed as wind speed and computed using the U.S. hurricane model in AIR-Worldwide (2015). Loss distributions are discretized on 64-point grid. Spatial dependency is described by $\boldsymbol{\Sigma}$ with nested block diagonal structure in Figure 2B. This structure is a consequence of using exchangeable correlation in spatial bins at two scales: 1 km and 20 km. The values assigned to off-diagonal elements of $\boldsymbol{\Sigma}$ are 0.07 if two locations belong to the same 1 km bin and 0.02 if two location belong to the same 20 km bin but different 1 km bins, respectively. Correlation between any two locations in different 20 km bins is set to zero. Details of our estimation methodology are given in Einarsson et al. (2016). Portfolio rollup was performed via direct, hierarchical sequential and hierarchical RNN models using Split-Atom convolution with linear and 4-point regriding. Note that for the hierarchical sequential model, the order of convolutions simply corresponds to the order of locations in insurance portfolio. The maximum number of points discretizing the aggregate distribution grid was set to 256. Additionally, to keep the discretization dense at the bulk of the aggregate distribution, we investigated the effect of right tail truncation at losses with probabilities $\leq 10^{-10}$ .

The runs above were compared with Monte Carlo (MC) simulations. We applied Latin Hypercube Sampling in McKay et al. (1979) with sample reordering in Arbenz et al. (2012) to impose correlations between samples. We generated 30 realizations with 1 MM samples each. Second order statistics and Tail Value at Risk (or p%-TVaR, which measures the expectation of all the losses that are greater than the loss at a specified p% probability, see, e.g., Artzner et al. 1999; Latchman 2010) for these runs are presented in Table 1. For the direct model, linear regriding inflates the variance of the total risk. This behavior is alleviated by tail truncation and/or 4-point regriding coupled with RNN order of convolutions. Variance of the total risk obtained from the sequential model with linear regriding and no truncation is higher than the corresponding variance of the RNN model. This is due to increasing

size of the partial aggregate pdf support at upper branching nodes of the sequential tree as compared to that of the individual risks being cointegrated to these nodes. Again, tail truncation and 4-point regriding tackles this effect. For the RNN model, the reproduction of the tail statistics is poor if linear regriding is applied with or without tail truncation. These errors are reduced by 4-point method in Algorithms 4 and 5. Figure 6 shows comparisons between the aggregate distributions for direct and hierarchical models. Models based on linear regriding (the upper row) lead to coarse discretization of the final aggregation grid, with obvious shape mismatch between the sequential model and MC run. This is due to the lack of mechanism for keeping the dense discretization of the bulk of the total loss distribution and, for sequential model, due to increasing differences in max losses between partial aggregates and risks being cointegrated at branching nodes. Such mechanism is included in 4-point regriding (lower row in Figure 6). Interestingly, the grid of the total loss for direct model is still coarse.

**Table 1.** Mean ($\mu$), standard deviation ($\sigma$) and tail Value-at-Risk ($\text{TVaR}_{\%}$) at levels 1%, 5% and 10% of the total hurricane risk for 29,139 locations for (A) direct, (B) hierarchical sequential and (C) hierarchical RNN aggregation models compared to the average values from 30 realizations of MC runs. The losses are in [$MM]. Numbers in brackets represent percentage errors relative to MC simulations.

| | | MC | Linear Regriding No Truncation | Linear Regriding Tail Truncation | 4-Point Regridding |
|---|---|---|---|---|---|
| (A) | $\mu$ | 44.3 | 44.3 (0.00%) | 44.3 (0.00%) | 44.3 (0.00%) |
| | $\sigma$ | 7.1 | 10.9 (53.02%) | 7.1 (0.00%) | 7.1 (0.00%) |
| | $\text{TVaR}_{1\%}$ | 51.4 | 2550.9 (4864.2%) | 2550.8 (4864.0%) | 2550.8 (4863.9%) |
| | $\text{TVaR}_{5\%}$ | 48.5 | 502.1 (934.7%) | 501.4 (932.8%) | 501.9 (933.8%) |
| | $\text{TVaR}_{10\%}$ | 47.6 | 252.0 (429.7%) | 250.7 (427.1%) | 251.0 (427.6%) |
| (B) | $\mu$ | 44.3 | 44.3 (0.00%) | 44.3 (0.00%) | 44.3 (0.00%) |
| | $\sigma$ | 7.1 | 7.6 (6.11%) | 7.1 (0.00%) | 7.1 (0.00%) |
| | $\text{TVaR}_{1\%}$ | 93.7 | 81.2 (−13.3%) | 84.7 (−9.6%) | 88.0 (−6.1%) |
| | $\text{TVaR}_{5\%}$ | 66.3 | 62.5 (−5.8%) | 68.5 (3.4%) | 66.8 (0.8%) |
| | $\text{TVaR}_{10\%}$ | 58.6 | 67.3 (15.0%) | 60.5 (3.3%) | 59.8 (2.1%) |
| (C) | $\mu$ | 44.3 | 44.3 (0.00%) | 44.3 (0.00%) | 44.3 (0.00%) |
| | $\sigma$ | 7.1 | 7.2 (1.55%) | 7.1 (0.00%) | 7.1 (0.00%) |
| | $\text{TVaR}_{1\%}$ | 75.9 | 151.0 (98.9%) | 95.0 (25.2%) | 76.0 (0.1%) |
| | $\text{TVaR}_{5\%}$ | 62.9 | 184.4 (193.4%) | 98.0 (55.8%) | 63.5 (0.9%) |
| | $\text{TVaR}_{10\%}$ | 58.4 | 95.8 (63.9%) | 87.0 (48.9%) | 59.1 (1.1%) |

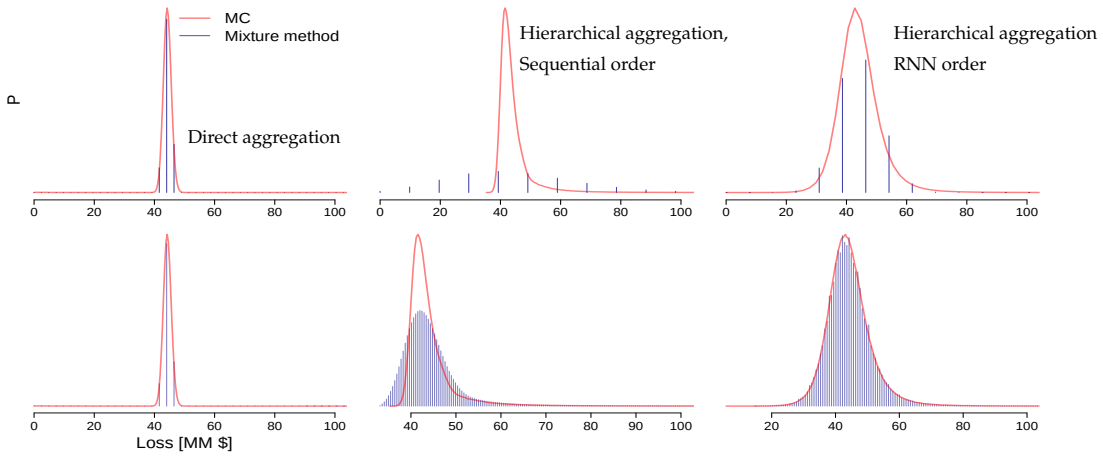



**Figure 6.** MC (red line) and convolution/comonotoinization based (blue bars) distributions of the total risk for 29,139 locations affected by hurricane peril using different aggregation models with linear regriding (upper row) and 4-point regriding (lower row). No tail truncation was applied. For consistency, the losses are plotted in [0; $100 MM] interval.

Here, the support span of $p_{S^+}$ is much larger than that of $p_{S^\perp}$. Placing combination of both sums on the same grid causes the discretization of the comonotonic sum to dominate. Conversely, the bulk is well resolved by hierarchical models, however, only the RNN model reproduces the shape of the total loss distribution obtained from MC run.

Processing times of the investigated models are listed in Table 2. Clearly, the proposed models outperform MC simulations. This is because (i) sampling from skewed loss distributions requires large sample size to resolve higher order moments of the total loss distribution while the implementation of the mixture method presented in the paper operates on 256-point grid and guarantees reproduction of the second order moments only, (ii) floating point addition (MC) and multiplication (the mixture method) have the same performance on (most) modern processors, (iii) large samples cause out-of-cache IO during reshuffling and sorting, i.e., the complexity $\mathrm{O}(N \log N)$ for fast sort does not apply.

Further, we investigated reproduction of the second order moments, 1,5,10%-TVaR and run times as a function of maximum grid size permitted during aggregation. These results are displayed in Figure 7. Oscillation of error curves (Figure 7A–E) is caused by variability in the total risk support size attributed to the maximum permitted grid size and to the risk aggregation scheme used. As expected, estimates of the mean ($\mu$) and the standard deviation ($\sigma$) perfectly reproduce their target values regardless of the support size. For small support sizes, the RNN model approximates TVaR better than sequential model (see Figure 7A–C). The pronounced underestimation of 1%-TVaR in Figure 7C is because 4-point regriding in Algorithm 5 eliminates negative probabilities at the expense of truncating min/max range of convolution and/or comonotonization support. The remedy is to increase the maximum support size. The sequential order has roughly linear growth of grid size because at the $l$th hierarchy of the sequential tree (see Figure 1), the left child node is always the aggregation of the first $l-1$ distributions, which guarantees the $l$th hierarchy has a grid size (before possible truncation) greater than that of the left child. The RNN order has nonlinear growth of grid size because for the $l$th aggregation, the two child nodes could be the aggregations of any number less than $l$ of distributions, see Figure 4. In turn, grid sizes in the sequential run are larger on average than those in the RNN run. The latter yields higher speed shown in Figure 7F.

**Table 2.** Processing times [s] for different risk aggregation models. MC run is a single realization with 1 MM samples. The mixture method implementation for hierarchical trees was optimized for performance with nested block diagonal correlation structure in Figure 2B. Intel i7-4770 CPU @ 3.40 GHz architecture with 16 GB RAM was used.

| Aggregation Model | MC | Linear Regridding No Truncation | Linear Regriding Tail Truncation | 4-Point Regriding |
|---|---|---|---|---|
| Direct | 1539 | 0.25 | 0.26 | 0.33 |
| Hierarchical, sequential | 12,769 | 0.34 | 0.35 | 0.44 |
| Hierarchical, RNN | 10,512 | 0.40 | 0.41 | 0.52 |

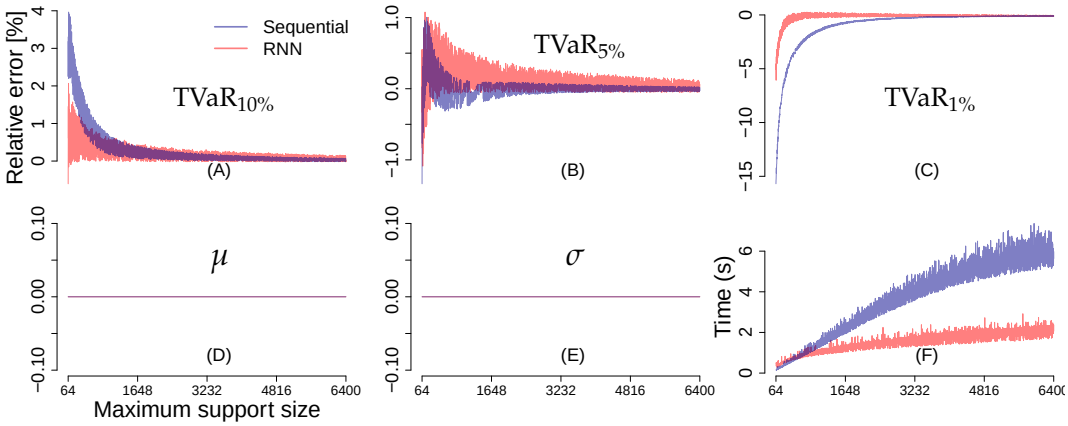



**Figure 7.** (**A**–**E**) Percentage errors in statistics of the total risk relative to the corresponding values obtained from MC simulations. Risk aggregation was performed for 29,139 locations affected by hurricane peril using sequential (blue) and RNN (red) models with 4-point regriding and maximum support size varying from 64 to 6400; (**F**) shows the average time cost of five runs for each maximum support size.

Oscillation of the computing time curves is mainly due to the cache mechanism (Hennessy and Patterson 2007, Chapter 5.5) in modern computer design. Varying support sizes leads to different *cache miss* frequencies during the entire run, and a cache miss is resolved by spending time reading from upper memory hierarchies. In consequence, a slightly larger support size could yield higher speed if it is better adapted to the cache hierarchies of the architecture where the program runs. A good example can be found at https://stackoverflow.com/questions/11413855/why-is-transposing-a-matrix-of-512x512-much-slower-than-transposing-a-matrix-of.

## 4. Conclusions

In this paper we presented several computationally fast models for CAT risk aggregation under spatial dependency. These models are variants of copula trees. Operations at branching nodes were facilitated by the Split-Atom convolution and estimation of distribution of the comonotonic sum. Using an example of ground-up loss estimation for 29,000 locations affected by a major hurricane event in the U.S., we have shown that the mixture method coupled with 4-point regriding and RNN order of convolutions, preserves target mean and variance of the total risk. Values of tail statistics compare favorably with those from, computationally slower, large sample MC runs.

## 5. Future Research Directions

In future work we aim to use the methods presented here for gross loss accumulation when complex financial terms act as non-linear transformations on partial loss aggregates in hierarchical trees. In such case the target second order statistics of the total risk are unknown and can only be inferred from MC runs. Since (sometimes nested) application of deductibles and limits makes the emergence of probability atoms even more pronounced than in the case of ground-up loss, fast gross loss aggregation strategies could potentially benefit from the Split-Atom approach. Another challenge is to develop a capital back allocation methodology which is in concert with our second order moment preservation requirement. This is crucial, for example, when risks in a geographical domain are part of one sub-limit by building and contents combined, and second sub-limit for business interruption; at the same time, some risks are not a part of any sub-limit.

Investigating how our methods can be extended to include negative correlations between pairs of risks is another exciting topic for future study. Some thoughts on this research direction are given in Appendix B.

**Author Contributions:** Conceptualization, R.W. and J.G.; methodology, R.W. and C.W.L.; software, C.W.L.; validation, R.W., C.W.L. and J.G.; formal analysis, R.W. and C.W.L.; investigation, R.W., C.W.L. and J.G.; resources, J.G.; data curation, C.W.L.; writing—original draft preparation, R.W. and C.W.L.; writing—review and editing, R.W. and C.W.L.; visualization, C.W.L.; supervision, R.W. and J.G.; project administration, R.W. and J.G.; funding acquisition, J.G.

**Funding:** This research received no external funding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

**Algorithm A1:** Split-Atom Convolution: 4-products

**Input :** Two discrete pdfs $p_X$ and $p_Y$ with supports:
$x = \{x_1, x_1 + h_x, x_1 + 2h_x, \ldots, x_1 + (N_x - 2)h_x, x_{N_x}\}$;
$y = \{y_1, y_1 + h_y, y_1 + 2h_y, \ldots, y_1 + (N_y - 2)h_y, y_{N_y}\}$;
and probabilities: $p_X(x) = \sum_{i=1}^{N_x} \delta(x - x_i) p_X(x_i)$; $p_Y(y) = \sum_{i=1}^{N_y} \delta(y - y_i) p_Y(y_i)$;
$N_s$ - maximum number of points for discretizing convolution grid

1: $x^{(1)} = \{x_1, x_1 + h_x, x_1 + 2h_x, \ldots, x_1 + (N_x - 2)h_x\}$; $p_X^{(1)}(x) = \sum_{i=1}^{N_x - 2} \delta(x - x_i^{(1)}) p_X^{(1)}(x_i^{(1)})$
2: $x^{(2)} = \{x_{N_x}\}$; $p_X^{(2)}(x) = \delta(x - x_i^{(2)}) \cdot p_X^{(2)}(x_i^{(2)})$ `// split the right atom of` $p_X$
3: $y^{(1)} = \{y_1, y_1 + h_y, y_1 + 2h_y, \ldots, y_1 + (N_y - 2)h_y\}$; $p_Y^{(1)}(y) = \sum_{i=1}^{N_y - 2} \delta(y - y_i^{(1)}) p_Y^{(1)}(y_i^{(1)})$
4: $y^{(2)} = \{y_{N_y}\}$; $p_Y^{(2)}(y) = \delta(y - y_i^{(2)}) \cdot p_Y^{(2)}(y_i^{(2)})$ `// split the right atom of` $p_Y$
5: $h_{s^\star} = \frac{x_{N_x} + y_{N_y} - (x_1 + y_1)}{N_{s^\star} - 1}$ `// corresponding step size of the main part of convolution grid` $s$
6: $h_s = \max(h_x, h_y, h_{s^\star})$ `// final step size of the main part of convolution grid` $s$
7: $N_s = \lceil \frac{x_{N_x} + y_{N_y} - x_1 - y_1}{h_s} \rceil + 1$ `// corresponding number of points`
8: `// set irregular convolution grid:`
9: $s = \{x_1 + y_1, x_1 + y_1 + h_s, \ldots, x_1 + y_1 + (N_s - 2)h_s, x_{N_x} + y_{N_y}\}$
10: $N_{x^{(1)\prime}} = \lceil \frac{x_1 + (N_x - 2)h_x - x_1}{h_s} \rceil + 1$ `// size of the regridded main part`
11: $x^{(1)\prime} = \{x_1, x_1 + h_s, x_1 + 2h_s, \ldots, x_1 + (N_{x^{(1)\prime}} - 1)h_s\}$ `// discretize` $x^{(1)}$ `with` $h_s$
12: $p_X^{(1)\prime}(x) = \sum_{i=1}^{N_x'} \delta(x - x_i^{(1)\prime}) p_X^{(1)\prime}(x_i^{(1)\prime})$ `// regrid` $p_X^{(1)}$
13: $N_{y^{(1)\prime}} = \lceil \frac{y_1 + (N_y - 2)h_y - y_1}{h_s} \rceil + 1$
14: $y^{(1)\prime} = \{y_1, y_1 + h_s, y_1 + 2h_s, \ldots, y_1 + (N_{y^{(1)\prime}} - 1)h_s\}$
15: $p_Y^{(1)\prime}(y) = \sum_{i=1}^{N_y'} \delta(y - y_i^{(1)\prime}) p_Y^{(1)\prime}(y_i^{(1)\prime})$
16: $\mathcal{B}^{(1)} = p_X^{(1)\prime} \oplus p_Y^{(1)\prime}$ `// Brute Force convolution`
17: $\mathcal{B}^{(2)} = p_X^{(2)} \oplus p_Y^{(1)\prime}$ `// --''--`
18: $\mathcal{B}^{(3)} = p_X^{(1)\prime} \oplus p_Y^{(2)}$ `// --''--`
19: $\mathcal{B}^{(4)} = p_X^{(2)} \oplus p_Y^{(2)}$ `// --''--`
20: Regrid $\mathcal{B}^{(1-4)}$ onto convolution grid $s$

**Output :** Probability mass function $p_S$ of the independent sum $S = X + Y$ with the support
$s = \{s_1, s_1 + h_s, s_1 + 2h_s, \ldots, s_1 + (N_s - 2)h_s, s_{N_s}\}$ and the associated probabilities
$p_S(s) = \sum_{k=1}^{N_s} \delta(s - s_k) p_S(s_k)$, where $s_{N_s} - [s_1 + (N_s - 2)h_s] \leq h_s$.

**Algorithm A2:** Modified local moment matching

**Input :** Discrete pdf $p_X$ with fine scale support $x = \{x_1, x_1 + h, \ldots, x_1 + (N-1)h\}$ and associated probabilities $p_X(x) = \sum_{i=1}^{N} \delta(x - x_1 - (i-1)h) \cdot p_X(x_1 + (i-1)h)$; the support of coarse scale probability mass function $p_{X'}$: $x' = \{x'_1, x'_1 + h', \ldots, x'_1 + (N'-1)h'\}$.
**Requirement:** $x'_1 \leq x_1$, $x'_{N'} \geq x_N$, $h < h'$.

1: For convenience, let $\{x_1, x_2, \ldots, x_N\} \leftarrow x$ and $\{x'_1, x'_2, \ldots, x'_{N'}\} \leftarrow x'$
2: Initialize $p_{X'}(x') \leftarrow \mathbf{0}$
3: **if** $x'_1 = x_1$ **then**
4: $\quad p_{X'}(x'_1) \leftarrow p_{X'}(x'_1) + p_X(x_1)$
5: **end if**
6: **for** $i < N'$, $j \leq N$, $x'_i < x_j \leq x'_{i+2}$, $x_j - x'_i \leq x'_{i+3} - x_j$ **do**
7:

$$\begin{bmatrix} p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \\ p_{X'}(x'_{i+2}) \end{bmatrix} \leftarrow \begin{bmatrix} p_{X'}(x'_i) \\ p_{X'}(x'_{i+1}) \\ p_{X'}(x'_{i+2}) \end{bmatrix} + \begin{bmatrix} 1 & 1 & 1 \\ x'_i & x'_{i+1} & x'_{i+2} \\ x'^2_i & x'^2_{i+1} & x'^2_{i+2} \end{bmatrix}^{-1} \begin{bmatrix} p_X(x_j) \\ x_j p_X(x_j) \\ x_j^2 p_X(x_j^2) \end{bmatrix}$$

8: **end for**

**Output:** Pdf $p_{X'}(x')$

## Appendix B

The mixture method implemented in the paper accounts only for positive correlations between pairs of risks. There is no reason, however, to believe that negative correlations do not exist. A situation when the CAT model errors are increasing at one location in space and they are decreasing at an adjacent location, or vice versa, induces negative correlation entries in the covariance frame $\mathrm{r}(X_i, X_j) = \rho_{i,j}$ where $\rho_{i,j} \in [-1; 1], i, j = 1, \ldots, M$. Representing that type of dependence in the direct aggregation model would require extension of (3) which includes $M$-dimensional Fréchet–Hoeffding lower bound $W(\mathbf{u}) = max(u_1 + u_2 + \ldots + u_M - (M-1), 0)$. While $W$ is a countermonotonic copula for $M = 2$, it is not a copula for $M \geq 3$, see Lee and Ahn (2014). Furthermore, minimum copula for $M \geq 3$ does not exist. Lee and Ahn (2014); Lee et al. (2017) give a new equivalent definition for multidimensional countermonotonicity referred to as the $d$-countermonotonicity or $d$-CM for short. They have shown that d-CM copulas constitute a minimal class of copulas and that, e.g., minimal Archimedian copula in McNeil and Nešlehová (2009), sometimes used as an approximation to Fréchet–Hoeffding lower bound, is $d$-CM. Let $S^- = X_1^- + X_2^- + \ldots, +X_M^-$ be the sum of losses when the individual risks are $d$-CM s. We aim at extending the representation of the arbitrary sum $S$ in (7) to:

$$S = w_1 S^- + w_2 S^{\perp} + w_3 S^+ \tag{A1}$$

where $w_1, w_2$ and $w_3$ are non-negative mixing weights such that $w_1 + w_2 + w_3 = 1$. By analogy with (8), we require that the weights in (A1) should be chosen such that the target moments E[S] and Var$[S]$ are preserved. Observing that the left side of:

$$\mathrm{Var}[S] = w_1 \mathrm{Var}[S^-] + w_2 \mathrm{Var}[S^{\perp}] + (1 - w_1 - w_2)\mathrm{Var}[S^+] \tag{A2}$$

can further be decomposed into two overlapping parts:

$$\begin{aligned} \mathrm{Var}[S_{w^-}] &= w_1 \mathrm{Var}[S^-] + w_2 \mathrm{Var}[S^{\perp}] \\ \mathrm{Var}[S_{w^+}] &= w_2 \mathrm{Var}[S^{\perp}] + (1 - w_1 - w_2)\mathrm{Var}[S^+] \end{aligned} \tag{A3}$$

where:

$$\begin{aligned}\mathrm{Var}[S_{w^-}] &= \sum_{i'} \sigma_{i'}^2 + \rho_{i'j'} \sum_{i' \neq j'} \sigma_{i'} \sigma_{j'} \\ \mathrm{Var}[S_{w^+}] &= \sum_{i''} \sigma_{i''}^2 + \rho_{i''j''} \sum_{i'' \neq j''} \sigma_{i''} \sigma_{j''}\end{aligned} \tag{A4}$$

The negative and positive correlations $\rho_{i'j'} \in [-1;0)$ and $\rho_{i''j''} \in (0;1]$ are defined for some indices $i', j' = 1, \ldots, N'$ and $i'', j'' = 1, \ldots, N''$. Expressing $\mathrm{Var}[S^-]$ using (A2) and substituting into (A3) yields:

$$\begin{aligned}\mathrm{Var}[S_{w^-}] &= \mathrm{Var}[S] - (1 - w_1 - w_2)\mathrm{Var}[S^+] \\ \mathrm{Var}[S_{w^+}] &= w_2 \mathrm{Var}[S^\perp] + (1 - w_1 - w_2)\mathrm{Var}[S^+] \\ \mathrm{Var}[S^-] &= \frac{\mathrm{Var}[S] - w_2 \mathrm{Var}[S^\perp] - (1 - w_1 - w_2)\mathrm{Var}[S^+]}{w_1}\end{aligned} \tag{A5}$$

Solving (A5) for the weights leads to:

$$\begin{aligned}w_1 &= \frac{\mathrm{Var}[S_{w^-}](\mathrm{Var}[S^\perp] - \mathrm{Var}[S^+]) + \mathrm{Var}[S](\mathrm{Var}[S^+] - \mathrm{Var}[S^\perp]) + \mathrm{Var}[S^+](\mathrm{Var}[S^\perp] - \mathrm{Var}[S_{w^+}])}{\mathrm{Var}[S^+]\mathrm{Var}[S^\perp]} \\ w_2 &= \frac{\mathrm{Var}[S_{w^-}] + \mathrm{Var}[S_{w^+}] - \mathrm{Var}[S]}{\mathrm{Var}[S^\perp]}\end{aligned} \tag{A6}$$

The results above indicate that for the direct model the natural and open research question is to construct a $d$-CM distribution with the variance $\mathrm{Var}[S^-]$ in computationally efficient manner. This issue can easily be addressed for hierarchical models where risk accumulation is performed pairwise. In this case, whenever negative correlations occur between pairs of risks and/or their partial sums, we simply replace the bivariate version of (1) with:

$$C_w(u_1, u_2) = (1 - w) \prod_i u_i + w \max(u_1 + u_2 - 1, 0) \tag{A7}$$

at the branching nodes. To compute $w$ we use (8) and replace $\sum_{i \neq j} \mathrm{r}(X_i^+, X_j^+)$ with $\sum_{i \neq j} \mathrm{r}(X_i^-, X_j^-)$ in the denominator. Comonotonization in Algorithm 6 with reversed order of losses and their probabilities is used to get the countermonotonic distribution of $S^-$. The latter requires an extra regriding and sorting operation so the increase in algorithmic complexity as compared to the mixture method with the comonotonic dependency is $O(N + N \log(N))$.

## References

AIR-Worldwide. 2015. AIR Hurricane Model for the United States. Available online: https://www.air-worldwide.com/publications/brochures/documents/air-hurricane-model-for-the-united-states-brochure (accessed on 5 May 2019).

Arbenz, Philipp, Christoph Hummel, and Georg Mainik. 2012. Copula based hierarchical risk aggregation through sample reordering. *Insurance: Mathematics and Economics* 51: 122–33. [CrossRef]

Artzner, Philippe, Freddy Delbaen, Jean-Marc Eber, and David Heath. 1999. Coherent measures of risk. *Mathematical Finance* 9: 203–28. [CrossRef]

Chaubey, Yogendra P., Jose Garrido, and Sonia Trudeau. 1998. On the computation of aggregate claims distributions: Some new approximations. *Insurance: Mathematics and Economics* 23: 215–30. [CrossRef]

Cherubini, Umberto, Elisa Luciano, and Walter Vecchiato. 2004. *Copula Methods in Finance*. Wiley Finance Series. Chichester: John Wiley and Sons Ltd.

Clark, K. 2015. *Catastrophe Risk*. International Actuarial Association/Association Actuarielle Internationale. Available online: http://www.actuaries.org/LIBRARY/Papers/RiskBookChapters/IAA_Risk_Book_Iceberg_Cover_and_ToC_5May2016.pdf (accessed on 5 May 2019).

Cormen, Thomas H., Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. 2009. *Introduction to Algorithms*, 3rd ed. Cambridge: MIT Press.

Côté, Marie-Pier, and Christian Genest. 2015. A copula-based risk aggregation model. *The Canadian Journal of Statistics* 43: 60–81. [CrossRef]

Denuit, Michel, Jan Dhaene, and Carmen Ribas. 2001. Does positive dependence between individual risks increase stop-loss premiums? *Insurance: Mathematics and Economics* 28: 305–8. [CrossRef]

Dhaene, Jan, Michel Denuit, Marc J. Goovaerts, Rob Kaas, and David Vyncke. 2002. The concept of comonotonicity in actuarial science and finance: Theory. *Insurance: Mathematics and Economics* 31: 3–33. [CrossRef]

Dhaene, Jan, Daniël Linders, Wim Schoutens, and David Vyncke. 2014. A multivariate dependence measure for aggregating risks. *Journal of Computational and Applied Mathematics* 263: 78–87. [CrossRef]

Einarsson, Baldvin, Rafał Wójcik, and Jayanta Guin. 2016. Using intraclass correlation coefficients to quantify spatial variability of catastrophe model errors. Paper Presented at the 22nd International Conference on Computational Statistics (COMPSTAT 2016), Oviedo, Spain, August 23–26. Available online: http://www.compstat2016.org/docs/COMPSTAT2016_proceedings.pdf (accessed on 5 May 2019).

Evans, Diane L., and Lawrence M. Leemis. 2004. Algorithms for computing the distributions of sums of discrete random variables. *Mathematical and Computer Modelling* 40: 1429–52. [CrossRef]

Galsserman, Paul. 2004. *Monte Carlo Methods in Financial Engineering*. New York: Springer.

Gerber, Hans U. 1982. On the numerical evaluation of the distribution of aggregate claims and its stop-loss premiums. *Insurance: Mathematics and Economics* 1: 13–18. [CrossRef]

Grossi, Patricia, Howard Kunreuther, and Don Windeler. 2005. An introduction to catastrophe models and insurance. In *Catastrophe Modeling: A New Approach to Managing Risk*. Edited by Patricia Grossi and Howard Kunreuther. Huebner International Series on Risk. Insurance and Economic Security. Boston: Springer Science+Business Media.

Hennessy, John L., and David A. Patterson. 2007. *Computer Architecture: A Quantitative Approach*, 4th ed. San Francisco: Morgan Kaufmann.

Hürlimann, Werner. 2001. Analytical evaluation of economic risk capital for portfolios of gamma risks. *ASTIN Bulletin* 31: 107–22. [CrossRef]

Iman, Ronald L., and William-Jay Conover. 1982. A distribution-free approach to inducing rank order correlation among input variables. *Communications in Statistics-Simulation and Computation* 11: 311–34. [CrossRef]

Koch, Inge, and Ann De Schepper. 2006. The Comonotonicity Coefficient: A New Measure of Positive Dependence in a Multivariate Setting. Available online: https://ideas.repec.org/p/ant/wpaper/2006030.html (accessed on 5 May 2019).

Koch, Inge, and Ann De Schepper. 2011. Measuring comonotonicity in m-dimensional vectors. *ASTIN Bulletin* 41: 191–213.

Latchman, Shane. 2010. Quantifying the Risk of Natural Catastrophes. Available online: http://understandinguncertainty.org/node/622 (accessed on 5 May 2019).

Lee, Woojoo, and Jae Youn Ahn. 2014. On the multidimensional extension of countermonotonicity and its applications. *Insurance: Mathematics and Economics* 56: 68–79. [CrossRef]

Lee, Woojoo, Ka Chun Cheung, and Jae Youn Ahn. 2017. Multivariate countermonotonicity and the minimal copulas. *Journal of Computational and Applied Mathematics* 317: 589–602. [CrossRef]

McKay, Michael D., Richard J. Beckman, and William J. Conover. 1979. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics* 21: 239–45.

McNeil, Alexander J., and Johanna Nešlehová. 2009. Multivariate Archimedean copulas, d-monotone functions and l1–norm symmetric distributions. *The Annals of Statistics* 37: 3059–97. [CrossRef]

Nelsen, Roger B. 2006. *An Introduction to Copulas*. Springer Series in Statistics. New York: Springer.

Panjer, Harry H., and B. W. Lutek. 1983. Practical aspects of stop-loss calculations. *Insurance: Mathematics and Economics* 2: 159–77. [CrossRef]

Robertson, John. 1992. The computation of aggregate loss distributions. *Proceedings of the Casualty Actuarial Society* 79: 57–133.

Shevchenko, Pavel V. 2010. Calculation of aggregate loss distributions. *The Journal of Operational Risk* 5: 3–40. [CrossRef]

Venter, Gary G. 2001. D.e. Papush and g.s. Patrik and f. Podgaits. *CAS Forum*. Available online: http://www.casact.org/pubs/forum/01wforum/01wf175.pdf (accessed on 5 May 2019).

Venter, Gary G. 2013. Effects of Parameters of Transformed Beta Distributions. *CAS Forum*. Available online: https://www.casact.org/pubs/forum/03wforum/03wf629c.pdf (accessed on 5 May 2019).

Vilar, José L. 2000. Arithmetization of distributions and linear goal programming. *Insurance: Mathematics and Economics* 27: 113–22. [CrossRef]

Walhin, J. F., and J. Paris. 1998. On the use of equispaced discrete distributions. *ASTIN Bulletin* 28: 241–55. [CrossRef]

Wang, Shaun. 1998. Aggregation of correlated risk portfolios: Models and algorithms. *Proceedings of the Casualty Actuarial Society* 85: 848–939.

Wojcik, Rafał, Charlie Wusuo Liu, and Jayanta Guin. 2016. Split-Atom convolution for probabilistic aggregation of catastrophe losses. In *Proceedings of 51st Actuarial Research Conference (ARCH 2017.1)*. SOA Education and Research Section. Available online: https://www.soa.org/research/arch/2017/arch-2017-iss1-guin-liu-wojcik.pdf (accessed on 5 May 2019).

Zhang, Jilian, Kyriakos Mouratidis, and HweeHwa Pang. 2011. Heuristic algorithms for balanced multi-way number partitioning. Paper Presented at the International Joint Conference on Artificial Intelligence (IJCAI), Barcelona, Spain, July 16–22. Research Collection School Of Information Systems. Menlo Park: AAAI Press, pp. 693–98.