# Monte Carlo Simulation with Mathematical Convolution of Frequency and Severity Distributions in Operational Risk Capital Model in the Basel Accords

In October 2014, the Basel Committee on Banking Supervision released a Basel Consultative Document entitled, "Operational Risk: Revisions to the Simpler Approaches," and in it describes operational risk as "the sum product of frequency and severity of risk events within a one-year time frame and defines the Operational Capital at Risk (OPCAR) as the tail-end 99.9% Value at Risk (VaR)" (Basel Committee, 2014). The Basel Consultative Document describes a Single Loss Approximation (SLA) model defined as $F_S^{-1}(p) = F_X^{-1}\left[1 - \frac{1-p}{\lambda}\right] + (\lambda - 1)E[X]$, where the inverse of the compound distribution $F_S^{-1}$ is the summation of the unexpected losses $UL = F_X^{-1}\left[1 - \frac{1-p}{\lambda}\right]$ and expected losses $EL = (\lambda - 1)E[X]$; $\lambda$ is the Poisson distribution's input parameter (average frequency per period; in this case, 12 months); and $X$ represents one of several types of continuous probability distributions representing the severity of the losses (e.g., Pareto, Log Logistic, etc.). The Document further states that this is an approximation model limited to subexponential-type distributions only and is fairly difficult to compute. The $X$ distribution's cumulative distribution function (CDF) will need to be inverted using Fourier transform methods, and the results are only approximations based on a limited set of inputs and their requisite constraints. Also, as discussed below, the SLA model proposed in the Basel Consultative Document significantly underestimates OPCAR. The OPCAR methodology estimates a bank's operational risk capital through the convolution of a single severity distribution and a single frequency distribution. Each bank's OPCAR estimate was assumed to refer to a unique operational risk category, having a specific aggregated frequency and severity of losses (Basel Committee, 2014).

The concept of significant loss events attributed to operational risk was introduced in the Basel II-IV accords by the Bank of International Settlements. Loss processes that contribute most to capital risk, the so-called high-consequence, low-frequency loss processes, with heavy-tailed loss process modeling where implications of such tail assumptions for the severity risk model is important in operational risk calculations (Peters, Targino, and Shevchenko 2013).

This paper provides a new and alternative convolution methodology to compute OPCAR that is applicable across a large variety of continuous probability distributions for risk severity and includes a comparison of their results with Monte Carlo risk simulation methods. As will be shown, both the new algorithm using numerical methods to model OPCAR and the Monte Carlo risk simulation approach tends to the same results and seeing that simulation can be readily and easily applied in the CMOL software and Risk Simulator software (source: www.realoptionsvaluation.com), we recommend using simulation methodologies for the sake of simplicity. While the Basel Committee has, throughout its Basel II-IV requirements and recommendations, sought after simplicity so as not to burden banks with added complexity, it still requires sufficient rigor and substantiated theory. Monte Carlo risk simulation methods pass the test on both fronts and are, hence, the recommended path when modeling OPCAR.

## Problem with Basel OPCAR

We submit that the SLA estimation model proposed in the Basel Consultative Document is insufficient and significantly underestimates an actual OPCAR value. A cursory examination shows

that with various $\lambda$ values, such as $\lambda = 1, \lambda = 10, \lambda = 100, \lambda = 1000$, the $UL = F_X^{-1}\left[1 - \frac{1-p}{\lambda}\right]$ will yield $\left[1 - \frac{1-p}{\lambda}\right]$ probability values ($\eta$) of 0.999, 0.9999, 0.99999, and 0.999999. $UL = F_X^{-1}[\eta]$ for any severity distribution, $X$, will only yield the severity distribution's values and not the total unexpected losses. For instance, suppose the severity distribution ($X$) of a single risk event on average ranges from \$1M (minimum) to \$2M (maximum), and, for simplicity, assume it is a Uniformly distributed severity of losses. Further, suppose that the average frequency of events is 1,000 times per year. Based on a back-of-the-envelope calculation, one could then conclude that the absolute highest operational risk capital losses will never exceed \$2B per year (this assumes the absolute worst-case scenario of \$2M loss per event multiplied by 1,000 events in that entire year). Nonetheless, using the inverse of the $X$ distribution at $\eta = 0.999999$ will yield a value close to \$2M only, and adding that to the adjusted expected value of EL (let's just assume somewhere close to \$1.5B based on the Uniform distribution) is still a far cry from the upper end of \$2B.

Figure 1 shows a more detailed calculation that proves the Basel Consultative Document's SLA approximation method significantly understates the true distributional operational Value at Risk amount. In the figure, we test four examples of a Poisson–Weibull convolution. The Poisson distribution with Lambda risk event frequency $\lambda = 10, \lambda = 25, \lambda = 50$, and $\lambda = 100$ are tested, together with a Weibull risk severity distribution: $\alpha = 1.5$ and $\beta = 2.5$. These values are shown as highlighted cells in the figure. Using the Basel OPCAR model, we compute the UL and EL. In the UL computation, we use $UL = F_X^{-1}\left[1 - \frac{1-p}{\lambda}\right] = F_X^{-1}[\eta]$. The column labeled PROB is $\eta$. The ICDF $X$ column denotes the $UL = F_X^{-1}[\eta]$. By applying the inverse of the Weibull CDF on the probability, we obtain the $UL$ values. Next, the $EL$ calculations are simply $EL = (\lambda - 1)E[X]$ with $E[X]$ being the expected value of the Weibull distribution $X$, where $E[X] = \beta\Gamma\left[1 + \frac{1}{\alpha}\right]$. The OPCAR is simply $UL + EL$. The four OPCAR results obtained are 31.30, 65.87, 122.82, and 236.18.

We then tested the results using Monte Carlo risk simulation using the Risk Simulator software (source: www.realoptionsvaluation.com) by setting four Poisson distributions with their respective $\lambda$ values and a single Weibull distribution with $\alpha = 1.5$ and $\beta = 2.5$. Then, the Weibull distribution is multiplied by each of the Poisson distributions to obtain the four Total Loss Distributions. The simulation was run for 100,000 trials and the results are shown in Figure 1 as forecast charts at the bottom. The Left Tail $\leq$ 99.9% quantile values were obtained and can be seen in the charts (116.38, 258.00, 476.31, and 935.25). These are significantly higher than the four OPCAR results.

Next, we ran a third approach using the newly revised convolution algorithm we propose in this article. The convolution model shows the same values as the Monte Carlo risk simulation results: 116.38, 258.00, 476.31, and 935.25 when rounded to two decimals. The inverse of the convolution function computes the corresponding CDF percentiles, and they are all 99.9% (rounded to one decimal; see the Convolution and Percentile columns in Figure 1). Using the same inverse of the convolution function and applied to the Basel Consultative Document's SLA model results, we found that the four SLA results were at the following OPCAR percentiles: 75.75%, 66.94%, 62.78%, and 60.38%, again significantly different than the requisite 99.9% Value at Risk level for operational risk capital required by the Basel Committee.

Therefore, due to this significant understatement of operational capital at risk, the remainder of this article focuses on explaining the theoretical details of the newly revised convolution model we developed that provides exact OPCAR results under certain conditions. We then compare the results using Monte Carlo risk simulation methods using Risk Simulator software as well as the Credit, Market, Operational, and Liquidity (CMOL) Risk software (source: www.realoptionsvaluation.com).

Finally, the caveats and limitations of this new approach as well as conclusions and recommendations are presented.
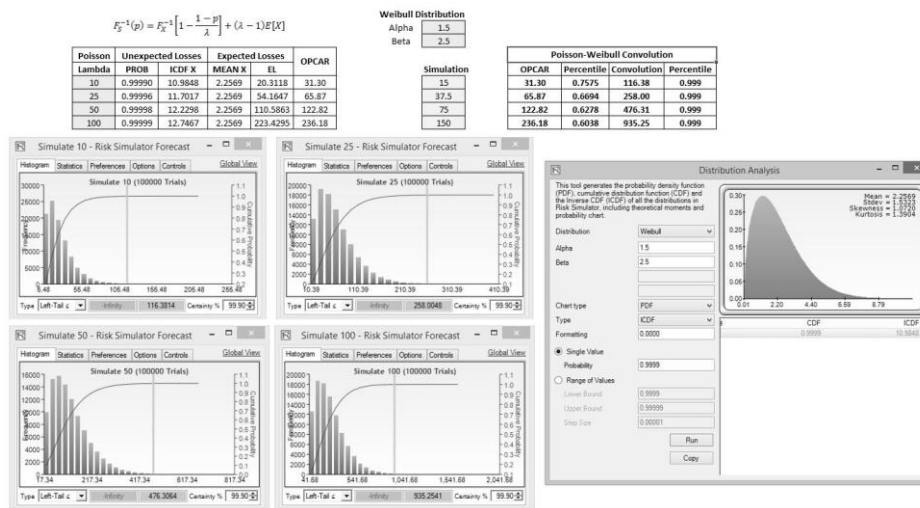


**Figure 1: Comparing Basel OPCAR, Monte Carlo Risk Simulation, and the Convolution Algorithm**

## Literature Review and Theoretical Constructs

Convolution is one of the primary concepts of linear system theory. It answers the problem of finding the system zero-state response due to any input—the most important problem for linear systems (Gajic, 2003). In developing continuous-time convolution, the procedure is much the same as in discrete-time events although in the continuous-time case the signal is represented first as a linear combination of narrow rectangles (basically a staircase approximation to the time function). As the width of these rectangles becomes infinitesimally small, they behave like impulses. The superposition of these rectangles to form the original time function in its limiting form becomes an integral, and the representation of the output of a linear, time-invariant system as a linear combination of delayed impulse responses also becomes an integral. The resulting integral is referred to as the convolution integral and is similar in its properties to the convolution sum for discrete-time signals and systems (Oppenheim and Willsky, 1996). An associated approach, the convolutional neural network is a feed-forward neural network that is generally used to analyze visual images by processing data with grid-like topology. In convolution operation, the arrays are multiplied element-wise, and the product is summed to create a new array. The first three elements in the matrix are multiplied by the elements of matrix b. The product is summed to get the result. The next three elements from the matrix are multiplied by the elements in matrix b, and the product is summed up. This process continues until the convolution operation is complete (SimpliLearn, 2022).

Numerical simulation can sometimes be time-consuming when simulating the evolution of distributions. Convolution is a more efficient way to analyze the evolution of the entire population of data distributions and their VaR. Convolution using modified scaled input distributions can match

the results of numerical simulation perfectly for independent data (Siu, 2022). In fact, modified scale distributions can be simplified using average order approximations of the distributions' functions (Ramaré, 2022) as will be discussed later.

In addition, the Taylor expansion series can be used to model the convolution series and deduce explicit formulas for the relevant coefficients within the n-fold general sequential fractional derivatives (Luchko, 2022). Also, the convolution of distributions can be achieved using fractional differentiation and integration (Georgiev, 2021). A combination of these convolution methods can also be applied to neural networks, creating a convolution neural network (CNN) model suitable for applications in artificial intelligence such as face recognition as well as natural language processing application (Kaur and Agrawal,2022). CNN can also be applied to simulate complex weather patterns when there are multiple probabilistic weather effects interacting in a certain location (Sayeed et al, 2021).

Let $X$, $Y$, and $Z$ be real-valued random variables whereby $X$ and $Y$ are independently distributed with no correlations. Further, we define $F_X$, $F_Y$, and $F_Z$ as their corresponding CDFs, and $f_x$, $f_y$, $f_z$ are their corresponding PDFs. Next, we assume that $X$ is a random variable denoting the Frequency of a certain type of operational risk occurring and is further assumed to have a discrete Poisson distribution. $Y$ is a random variable denoting the Severity of the risk (e.g., monetary value or some other economic value) and can be distributed from among a group of continuous distributions (e.g., Fréchet, Gamma, Log Logistic, Lognormal, Pareto, Weibull, etc.). Therefore, $Frequency \times Severity$ equals the $Total\ Risk\ Losses$, which we define as $Z$, where $Z = X \times Y$.

Then the Total Loss formula, which is also sometimes known as the Single Loss Approximation (SLA) model, yields:

$$F_Z(t) = P(Z < t) = \sum_k P(XY < t \mid X = k) \times P(X = k)$$

$$F_Z(t) = P(Z < t) = \sum_k P(kY < t) \times P(X = k)$$

where the term with $X = 0$ is treated separately:

$$F_Z(t) = P(0 < t|X = 0) \times P(X = 0) + \sum_{k \neq 0} P\left(Y < \frac{t}{k}\right) \times P(X = k)$$

$$F_Z(t) = \sum_{k \neq 0} f_X(k)F_Y\left(\frac{t}{k}\right) + P(X = 0) \qquad \text{(Equation 1)}$$

The next step is the selection of the number of summands in Equation 1. As previously assumed, $f_X(k) = P(X = k)$ is a Poisson distribution where $P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}$ and the rate of convergence in the series depends solely on the rate of convergence to $0$ of $\frac{\lambda^k}{k!}$ and does not depend on $t$, whereas the second multiplier $P\left(Y < \frac{t}{k}\right) \leq 1!$ Therefore, for all values of $t$ and an arbitrary $\delta > 0$ there is a value of $n$ such that:

$$\sum_{k > n} \frac{\lambda^k e^{-\lambda}}{k!} F_Y\left(\frac{t}{k}\right) < \delta \qquad \text{(Equation 2)}$$

In our case, $\delta$ can be set, for example, to 1/1000. Thus, instead of solving the quantile equation for $t_p$ with an infinite series, on the left-hand side of the equation we have:

$$F_Z(t) = P(Z < t) = \sum_k P\left(Y < \frac{t}{k}\right)\frac{\lambda^k e^{-\lambda}}{k!} = p \qquad \text{(Equation 3)}$$

We can then solve the equation:

$$F_Z(t,n) = \sum_{k\leq n}\frac{\lambda^k e^{-\lambda}}{k!}F_Y\left(\frac{t}{k}\right) = p \qquad \text{(Equation 4)}$$

with only $n$ summands.

For example, if we choose $p = 0.95$, $\delta = 1/1000$, and $n$ such that Equation 2 takes place, then the solution $t_p(n)$ of Equation 4 is such that:

$$\left|F_Z\left(t_p(n)\right) - F_Z(t_p(n),n)\right| < \frac{1}{1000} \qquad \text{(Equation 5)}$$

In other words, a quantile found from Equation 4 is almost the true value, with a resulting error precision in probabilities less than 0.1%.

The only outstanding issue that remains is to find an estimate for $n$ given any level of $\delta$. We have:

$$\sum_{k>n}\frac{\lambda^k e^{-\lambda}}{k!}F_Y\left(\frac{t}{k}\right) < e^{-\lambda}\sum_{k>n}\frac{\lambda^k}{k!} \qquad \text{(Equation 6)}$$

The exponential series $R_n(\lambda) = \sum_{k>n}\frac{\lambda^k}{k!}$ in Equation 6 is bounded by $\frac{\lambda^{n+1}e^{\lambda}}{(n+1)!}$ by applying Taylor's Expansion Theorem, with the remainder of the function left for higher exponential function expansions. By substituting the upper bound for $R_n(\lambda)$ in Equation 6, we have:

$$\sum_{k>n}\frac{\lambda^k e^{-\lambda}}{k!}F_Y\left(\frac{t}{k}\right) < \frac{\lambda^{n+1}}{(n+1)!} \qquad \text{(Equation 7)}$$

Now we need to find the lower bound in $n$ for the solution of the inequality:

$$\frac{\lambda^{n+1}}{(n+1)!} < \delta \qquad \text{(Equation 8)}$$

Consider the following two cases:

1. If $\lambda \leq 1$, then $\frac{\lambda^{n+1}}{(n+1)!} \leq \frac{1}{(n+1)!} \leq (n+1)^{-(n+1)}e^n$. Consequently, we can solve the inequality $(n+1)^{-(n+1)}e^n < \delta$. Since $n^n$ grows quickly, we can simply take $n > -\ln\delta$. For example, for $\delta = \frac{1}{1000}$ it is sufficient to set $n = 7$ to satisfy Equation 8.

2. If $\lambda > 1$, then, in this case, using the same bounds for the factorial, we can choose $n$ such that:

$$(n+1)(\ln(n+1) - \ln\lambda - 1) > -\ln\delta - 1 \qquad \text{(Equation 9)}$$

To make the second multiplier greater than 1, we will need to choose $n > e^{2+\ln\lambda} - 1$.

*Approximation to the solution of the equation $F_Z(t) = p$ for a quantile value*

From the previous considerations, we found that instead of solving $F_Z(t) = p$ for $t$, we can solve $F_Z(t, n) = \sum_{k \leq n} \frac{\lambda^k e^{-\lambda}}{k!} F_Y\left(\frac{t}{k}\right) = p$ with $n$ set at the level indicated above. The value for $t_p$ resulting from such a substitution will satisfy the inequality $\left| F_Z\left(t_p(n)\right) - F_Z(t_p(n), n) \right| < \delta$.

*Solution of the equation $F_Z(t, n) = p$ given $n$ and $\delta$*

By moving $t$ to the left one unit at a time, we can find the first occurrence of the event $t = a$ such that $F_Z(a, n) \leq p$. Similarly, moving $t$ to the right we can find $b$ such that $F_Z(b, n) \geq p$. Now we can use a simple Bisection Method or other search algorithms to find the optimal solution to $F_Z(t, n) = p$.

## Empirical Results: Convolution versus Monte Carlo Risk Simulation for OPCAR

Based on the explanations and algorithms outlined above, the convolution approximation models are run, and the results are compared with Monte Carlo risk simulation outputs. These comparisons will serve as empirical evidence of the applicability of both approaches.

Figure 2 shows the 10 most commonly used Severity distributions, namely, Exponential, Fréchet, Gamma, Logistic, Log Logistic, Lognormal (Arithmetic and Logarithmic inputs), Gumbel, Pareto, and Weibull. The Frequency of risk occurrences is set as Poisson, with Lambda ($\lambda$) or average frequency rate per period as its input. The input parameters for the 10 Severity distributions are typically Alpha ($\alpha$) and Beta ($\beta$), except for the Exponential distribution that uses a rate parameter, Rho ($\rho$), and Lognormal distribution that requires the mean ($\mu$) and standard deviation ($\sigma$) as inputs. For the first empirical test, we set $\lambda = 10$, $\alpha = 1.5$, $\beta = 2.5$, $\rho = 0.01$, $\mu = 1.8$, and $\sigma = 0.5$ for the Poisson frequency and 10 severity distributions. The Convolution Model row in Figure 2 was computed using the algorithms outlined above, and a set of Monte Carlo risk simulation assumptions were set with the same input parameters and simulated 100,000 trials with a prespecified seed value. The results from the simulation were pasted back into the model under the Simulated Results row and the Convolution Model was calculated based on these simulated outputs. Figure 2 shows 5 sets of simulation percentiles: 99.9%, 99.0%, 95.0%, 90.0%, and 50.0%. As can be seen, all of the simulation results and the convolution results on average agree to approximately within ±0.2%.

Figure 3 shows another empirical test whereby we select one specific distribution; in the illustration, we used the Poisson–Weibull compound function. The alpha and beta parameters in Weibull were changed, in concert with Poisson's lambda input. The first four columns show that the alpha and beta are being held steady while the lambda parameter is changing, whereas the last six columns show the same lambda with different alpha and beta input values (increasing alpha with beta constant and increasing beta with alpha constant). When the simulation results and the convolution results were compared, on average, they agree to approximately within ±0.2%.

Figure 4 shows the Credit, Market, Operational, and Liquidity (CMOL) risk software's operational risk module and how the simulation results agree with the convolution model. The

CMOL software uses the algorithms described above. The CMOL software settings are 100,000 Simulation Trials with a Seed Value of 1 with an OPCAR set to 99.90%.

Figures 5–8 show additional empirical tests where all 10 severity distributions were perturbed, convoluted, and compared with the simulation results. The results agree on average around ±0.3%.

| | | | | | |
|---|---|---|---|---|---|
| Lambda | 10.0000 | | Exponential's RHO | 0.0100 | |
| Alpha | 1.5000 | | Lognormal's MEAN | 1.8000 | |
| Beta | 2.5000 | | Lognormal's STDEV | 0.5000 | |
| Required Percentile | 0.9000 | | | | |

**Analytical Approximation Results Using Convolution Methods**

| Type | Poisson-Exponential | Poisson-Frechet | Poisson-Gamma | Poisson-Logistic | Poisson-LogLogistic | Poisson-Lognormal | Poisson-LognormalLog | Poisson-GumbelMax | Poisson-Pareto | Poisson-Weibull |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result 99.9% | 8194.65 | 2438.51 | 251.33 | 222.34 | 226.61 | 54.77 | 344.30 | 229.89 | 2702.52 | 117.35 |
| Convolution Model | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result 99.0% | 5056.14 | 562.08 | 160.38 | 145.78 | 98.20 | 41.49 | 225.87 | 149.77 | 538.58 | 82.23 |
| Convolution Model | 98.9% | 99.0% | 99.0% | 98.9% | 98.9% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result 95.0% | 3123.83 | 180.99 | 103.97 | 93.38 | 51.88 | 32.22 | 151.48 | 95.55 | 187.67 | 56.79 |
| Convolution Model | 94.9% | 94.8% | 94.9% | 94.9% | 95.1% | 94.9% | 94.9% | 95.0% | 95.0% | 95.0% |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result 90.0% | 2333.55 | 112.07 | 80.84 | 70.72 | 38.20 | 27.99 | 123.34 | 73.54 | 118.84 | 45.81 |
| Convolution Model | 89.9% | 89.8% | 90.0% | 89.8% | 90.2% | 89.7% | 90.0% | 90.0% | 90.0% | 90.0% |
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result 50.0% | 643.94 | 30.96 | 27.55 | 13.12 | 14.36 | 16.76 | 57.97 | 21.77 | 40.37 | 18.29 |
| Convolution Model | 49.8% | 49.8% | 49.8% | 49.7% | 50.1% | 49.7% | 50.0% | 49.5% | 49.9% | 50.0% |
| **Simulation Models** | | | | | | | | | | |
| Frequency | 10.00 | | | | | | | | | |
| Severity | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 | 10.00 |
| Multiplication | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |

**Figure 2: Comparing Convolution to Simulation Results I**

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| Lambda | 10.00 | 25.00 | 75.00 | 100.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 |
| Alpha | 1.50 | 1.50 | 1.50 | 1.50 | 1.50 | 15.00 | 45.00 | 1.50 | 45.00 | 1.50 |
| Beta | 2.50 | 2.50 | 2.50 | 2.50 | 2.50 | 2.50 | 2.50 | 15.00 | 50.00 | 75.00 |

**Analytical Approximation Results Using Convolution Methods: Weibull Distribution**

| Type | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull | Poisson-Weibull |
|---|---|---|---|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| Simulated Result | 117.35 | 251.52 | 716.62 | 943.55 | 488.72 | 186.23 | 183.36 | 2919.98 | 3651.15 | 14629.3 |
| Convolution Model | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% | 99.9% |
| Simulated Result | 82.23 | 185.40 | 531.75 | 708.24 | 362.16 | 169.24 | 167.17 | 2159.08 | 3339.56 | 10750.8 |
| Convolution Model | 99.0% | 98.9% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% |
| Simulated Result | 56.79 | 134.72 | 393.66 | 522.88 | 264.25 | 154.15 | 153.77 | 1589.67 | 3071.24 | 7909.56 |
| Convolution Model | 95.0% | 95.0% | 94.9% | 94.9% | 94.9% | 95.0% | 95.1% | 95.0% | 95.0% | 94.9% |
| Simulated Result | 45.81 | 110.63 | 327.60 | 437.82 | 219.00 | 146.39 | 146.76 | 1323.64 | 2932.39 | 6579.73 |
| Convolution Model | 90.0% | 89.8% | 89.8% | 90.0% | 89.8% | 90.0% | 90.2% | 90.1% | 90.0% | 89.9% |
| Simulated Result | 18.27 | 47.35 | 145.21 | 193.97 | 96.33 | 120.06 | 122.99 | 578.52 | 2461.52 | 2895.14 |
| Convolution Model | 50.0% | 49.8% | 49.9% | 49.9% | 49.9% | 49.8% | 50.0% | 50.0% | 50.2% | 50.0% |
| **Simulation Models** | | | | | | | | | | |
| Frequency | 10.00 | 25.00 | 75.00 | 100.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 | 50.00 |
| Severity | 2.26 | 2.26 | 2.26 | 2.26 | 2.26 | 2.41 | 2.47 | 13.54 | 49.38 | 67.71 |
| Multiplication | 22.57 | 56.42 | 169.26 | 225.69 | 112.84 | 120.71 | 123.46 | 677.06 | 2469.13 | 3385.29 |

**Figure 3: Comparing Convolution to Simulation Results II**

Lambda 10.00 25.00 75.00
Alpha 1.50 1.50 1.50
Beta 2.50 2.50 2.50

**Analytical Approxima...**

| Type | Poisson-Weibull 1 | Poisson-Weibull 2 | Poisson-Weibull 3 |
|---|---|---|---|
| Simulated Result | 117.35 | 251.52 | 716.62 |
| Convolution Model | 99.9% | 99.9% | 99.9% |
| Simulated Result | 82.23 | 185.40 | 531.75 |
| Convolution Model | 99.0% | 98.9% | 99.0% |
| Type ID | 1 | 2 | 3 |
| Simulated Result | 56.79 | 134.72 | 393.66 |
| Convolution Model | 95.0% | 95.0% | 94.9% |
| Type ID | 1 | 2 | 3 |
| Simulated Result | 45.81 | 110.63 | 327.60 |
| Convolution Model | 90.0% | 89.8% | 89.8% |
| Type ID | 1 | 2 | 3 |
| Simulated Result | 18.27 | 47.35 | 145.21 |
| Convolution Model | 50.0% | 49.8% | 49.9% |

**Simulation Models**

| | | | |
|---|---|---|---|
| Frequency | 10.00 | 25.00 | 75.00 |
| Severity | 2.26 | 2.26 | 2.26 |
| Multiplication | 22.57 | 56.42 | 169.26 |

ROV CREDIT, MARKET, LIQUIDITY RISK - [ C:\Users\Dr. Johnathan Mun\Desktop\ROV Credit Market and Liquidity Example.rovcml ]

Credit Risk (ERC)   Market Risk: Asset Liability Management   Analytical Models   **Operational Risk**

Basel OPRISK (BIA, TSA, ASA, RSA)   **Basel OPCAR (AMA)**   Loss Data & Fitting (AMA)   Fitted Loss Distribution (AMA)   Simulated Losses (AMA)

Frequency and Severity Assumptions   **Convoluted Simulation Results**   Loss Severity Fitting

Select simulated segment to view:  Compound Poisson-Weibull   Left Tail <=   Percentiles:   99.90 %   Simulated Value: 117.35

| | Frequency | Severity | Loss Distribution |
|---|---|---|---|
| Mean | 9.9710 | 2.2600 | 22.5545 |
| Median | 10.0000 | 1.9606 | 18.2431 |
| Stdev | 3.1569 | 1.5325 | 17.6265 |
| Skew | 0.3182 | 1.0614 | 1.9953 |
| Kurtosis | 0.1390 | 1.3313 | 4.0664 |
| 0.1% | 2.0000 | 0.0241 | 0.2083 |
| 90.0% | 14.0000 | 4.3672 | 45.8037 |
| 95.0% | 15.0000 | 5.1942 | 56.7921 |
| Left Tail 99.90% | 21.0000 | 8.9608 | 117.3459 |

Simulated Convolution of Expected Losses: 22.55
Simulated Convolution of Unexpected Losses: 94.79
Simulated Total Operational Capital at Risk: 117.35

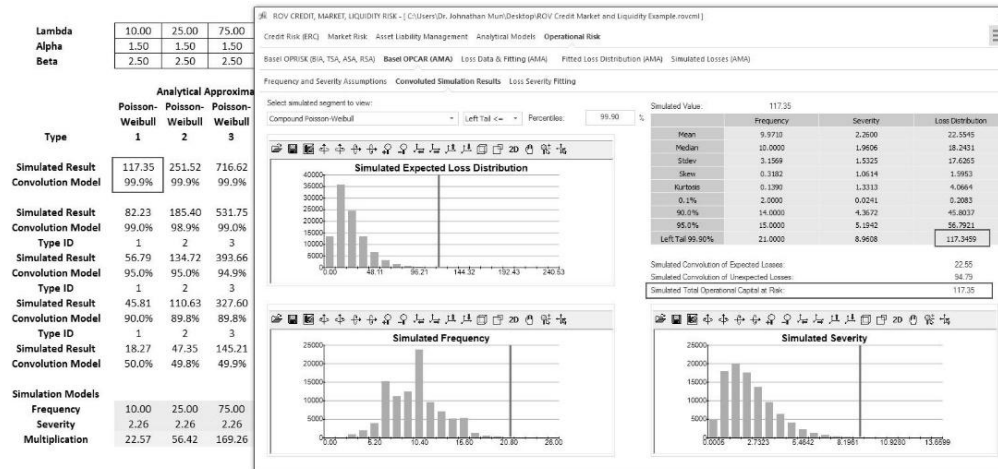Simulated Expected Loss Distribution

Simulated Frequency

Simulated Severity

**Figure 4: Comparing Convolution to Simulation Results III**

**Frequency: Poisson (λ=10) and Severity: 10 Distributions (α=1.5, β=2.5, ρ=0.01, μ=1.8, σ=0.5)**
SIMULATION RESULTS (APPROXIMATE PERCENTILE FROM SIMULATION GIVEN THE LEFT TAIL VALUE)

| Exponential | Frechet | Gamma | Logistic | Log Logistic | Lognormal | Lognormal | Gumbel | Pareto | Weibull |
|---|---|---|---|---|---|---|---|---|---|
| 99.8% | 99.8% | 99.8% | 99.9% | 99.9% | 99.8% | 99.9% | 99.9% | 99.8% | 99.8% |
| 0.06% | 0.08% | 0.07% | 0.03% | 0.04% | 0.09% | 0.04% | 0.03% | 0.10% | 0.08% |
| 98.9% | 99.0% | 99.4% | 99.0% | 98.9% | 98.9% | 99.0% | 98.9% | 98.9% | 98.9% |
| 0.10% | 0.03% | -0.38% | 0.00% | 0.07% | 0.08% | 0.05% | 0.06% | 0.10% | 0.10% |
| 94.9% | 95.1% | 94.7% | 95.0% | 95.2% | 95.0% | 94.8% | 95.0% | 94.9% | 95.0% |
| 0.06% | -0.06% | 0.35% | 0.03% | -0.15% | 0.00% | 0.21% | 0.00% | 0.10% | 0.01% |
| 90.0% | 90.2% | 90.1% | 90.1% | 90.2% | 90.0% | 90.1% | 90.0% | 89.9% | 90.1% |
| 0.00% | -0.17% | -0.08% | -0.08% | -0.17% | -0.02% | -0.07% | -0.01% | 0.10% | -0.12% |
| 50.0% | 50.2% | 50.1% | 50.1% | 50.1% | 50.1% | 50.1% | 50.1% | 50.0% | 50.1% |
| 0.05% | -0.15% | -0.05% | -0.08% | -0.13% | -0.10% | -0.05% | -0.06% | 0.01% | -0.11% |

**Figure 5: Empirical Results 1: Small Value Inputs**

**Frequency: Poisson (λ=50) and Severity: 10 Distributions (α=3, β=5, ρ=0.10, μ=5, σ=1)**
SIMULATION RESULTS (APPROXIMATE PERCENTILE FROM SIMULATION GIVEN THE LEFT TAIL VALUE)

| Exponential | Frechet | Gamma | Logistic | Log Logistic | Lognormal | Lognormal | Gumbel | Pareto | Weibull |
|---|---|---|---|---|---|---|---|---|---|
| 99.9% | 99.8% | 99.9% | 99.8% | 99.8% | 99.8% | 99.8% | 99.8% | 99.8% | 99.8% |
| 0.04% | 0.05% | -0.01% | 0.06% | 0.06% | 0.07% | 0.06% | 0.09% | 0.08% | 0.09% |
| 98.9% | 99.0% | 99.0% | 99.0% | 98.9% | 98.9% | 99.0% | 98.9% | 98.9% | 98.9% |
| 0.08% | -0.01% | -0.05% | -0.04% | 0.08% | 0.08% | 0.03% | 0.06% | 0.06% | 0.08% |
| 95.0% | 95.0% | 95.1% | 94.9% | 95.1% | 94.9% | 95.0% | 94.9% | 94.9% | 95.1% |
| -0.02% | -0.02% | -0.05% | 0.06% | -0.10% | 0.10% | 0.03% | 0.05% | 0.09% | -0.07% |
| 90.2% | 90.0% | 90.0% | 90.0% | 90.2% | 90.0% | 89.9% | 90.0% | 89.8% | 90.1% |
| -0.15% | -0.03% | 0.00% | 0.04% | -0.20% | 0.00% | 0.07% | 0.03% | 0.16% | -0.05% |
| 49.8% | 50.0% | 50.0% | 50.1% | 49.4% | 50.0% | 50.1% | 50.0% | 50.0% | 50.1% |
| 0.18% | 0.00% | 0.00% | -0.10% | 0.65% | 0.00% | -0.10% | 0.00% | 0.00% | -0.10% |

**Figure 6: Empirical Results 2: Average Value Inputs**

**Frequency: Poisson (λ=100) and Severity: 10 Distributions (α=25, β=35, ρ=0.025, μ=2.5, σ=0.9)**

SIMULATION RESULTS (APPROXIMATE PERCENTILE FROM SIMULATION GIVEN THE LEFT TAIL VALUE)

| Exponential | Frechet | Gamma | Logistic | Log Logistic | Lognormal | Lognormal | Gumbel | Pareto | Weibull |
|---|---|---|---|---|---|---|---|---|---|
| 99.9% | 99.9% | 99.8% | 99.9% | 99.8% | 99.8% | 99.8% | 99.8% | 99.9% | 99.8% |
| -0.05% | 0.04% | 0.06% | 0.04% | 0.07% | 0.07% | 0.05% | 0.07% | 0.03% | 0.08% |
| 99.5% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% | 98.9% | 98.9% | 98.9% | 99.0% |
| -0.54% | 0.03% | 0.01% | -0.02% | -0.02% | 0.05% | 0.10% | 0.11% | 0.06% | 0.01% |
| 97.2% | 95.0% | 95.2% | 95.0% | 95.1% | 95.0% | 95.0% | 94.9% | 94.9% | 95.0% |
| -2.20% | 0.00% | -0.17% | 0.00% | -0.10% | 0.05% | 0.00% | 0.15% | 0.13% | 0.00% |
| 93.7% | 90.0% | 90.0% | 90.0% | 90.0% | 90.1% | 89.9% | 90.0% | 90.0% | 90.3% |
| -3.70% | 0.00% | 0.05% | 0.00% | 0.00% | -0.05% | 0.15% | 0.00% | 0.00% | -0.25% |
| 56.2% | 50.0% | 50.1% | 50.0% | 50.0% | 50.0% | 50.0% | 50.1% | 50.0% | 50.0% |
| -6.21% | 0.00% | -0.05% | -0.01% | -0.01% | 0.00% | -0.01% | -0.06% | 0.05% | 0.00% |

**Figure 7: Empirical Results 3: Medium Value Inputs**

**Frequency: Poisson (λ=15) and Severity: 10 Distributions (α=80, β=25, ρ=5, μ=25, σ=3)**

SIMULATION RESULTS (APPROXIMATE PERCENTILE FROM SIMULATION GIVEN THE LEFT TAIL VALUE)

| Exponential | Frechet | Gamma | Logistic | Log Logistic | Lognormal | Lognormal | Gumbel | Pareto | Weibull |
|---|---|---|---|---|---|---|---|---|---|
| 99.9% | 99.8% | 99.9% | 99.9% | 99.9% | 99.8% | 99.8% | 99.8% | 99.9% | 99.8% |
| 0.05% | 0.06% | 0.04% | 0.05% | 0.03% | 0.12% | 0.12% | 0.08% | 0.05% | 0.06% |
| 98.9% | 98.9% | 99.0% | 99.0% | 99.0% | 99.0% | 99.0% | 98.9% | 99.1% | 99.0% |
| 0.07% | 0.06% | -0.02% | 0.03% | 0.04% | 0.01% | 0.01% | 0.08% | -0.07% | 0.04% |
| 95.0% | 95.0% | 95.0% | 95.1% | 95.0% | 95.0% | 95.0% | 95.0% | 95.0% | 95.0% |
| 0.04% | 0.00% | 0.01% | -0.06% | 0.00% | 0.02% | 0.02% | 0.02% | 0.00% | 0.05% |
| 90.0% | 90.0% | 90.2% | 90.0% | 90.2% | 90.0% | 90.0% | 89.9% | 90.0% | 90.1% |
| 0.00% | 0.00% | -0.16% | 0.01% | -0.17% | 0.01% | 0.01% | 0.09% | 0.00% | -0.10% |
| 49.9% | 50.1% | 50.1% | 50.0% | 49.9% | 50.0% | 50.0% | 50.1% | 50.0% | 50.0% |
| 0.08% | -0.05% | -0.05% | 0.05% | 0.10% | 0.00% | 0.00% | -0.10% | 0.00% | 0.00% |

**Figure 8: Empirical Results 4: High-Value Inputs**

## High Lambda and Low Lambda Limitations

As seen in Equation 4, we have the $F_Z(t,n) = \sum_{k \le n} \frac{\lambda^k e^{-\lambda}}{k!} F_Y\left(\frac{t}{k}\right) = p$ convolution model. The results are accurate to as many decimal-points precision as desired as long as $n$ is sufficiently large, but this would mean that the convolution model is potentially mathematically intractable. When $\lambda$ and $k$ are high (the value $k$ depends on the Poisson rate $\lambda$), such as $\lambda = 10{,}000$, the summand cannot be easily computed. For instance, Microsoft Excel 2013 can only compute up to factorial 170! where 171! and above returns the #NUM! error. Banks whose operational risks have large $\lambda$ rate values (extremely high frequency of risk events when all risk types are lumped together into a comprehensive frequency count) have several options: Create a breakdown of the various risk types (broken down by risk categories, by department, by division, etc.) such that the $\lambda$ is more manageable; use a continuous distribution approximation as shown below; or use Monte Carlo risk simulation techniques, where large $\lambda$ values will not pose a problem whatsoever.

Poisson distributions with large $\lambda$ values approach the Normal distribution, and we can use this fact to generate an approximation model for the convolution method. The actual deviation between Poisson and Normal approximation can be estimated by the Berry–Esseen inequality. For a more

accurate and order of magnitude tighter estimation we can use the Wilson–Hilferty approximation instead. For the large lambda situation, we can compute the CDF of the compound of two continuous distributions whose PDFs are defined as $f(x)$ defined on the positive interval of $(a, b)$ for the random variable $X$, and $g(y)$ defined on the positive interval of $(c, d)$, for the random variable $Y$. In other words, we have $0 < a < b < \infty$ and $0 < c < d < \infty$. The joint distribution $Z = XY$ has the following characteristics:

$$f(v) = \int_i^j f(x) g\left(\frac{v}{x}\right) \frac{1}{x} dx$$

$$F(v) = \int_m^n f(v) = \iint_{m\,i}^{n\,j} f(x) g\left(\frac{v}{x}\right) \frac{1}{x} dx$$

The integration can be applied analytically using numerical integration methods, but the results will critically depend on the integration range of $x$ and $v$. The values of $a, b, c, d$ can be computed by taking the inverse CDF of the distributions at 0.01% and 99.99% respectively (e.g., in the Normal distribution, this allows us to obtain real values instead of relying on the theoretical tails of $-\infty$ and $+\infty$). The following table summarizes the integration ranges:

| When $AD < BC$ | When $AD = BC$ | When $AD > BC$ |
|---|---|---|
| $j = \frac{v}{c}$ if $ac < v < ad$ | $j = \frac{v}{c}$ if $ac < v < ad$ | $j = \frac{v}{c}$ if $ac < v < bc$ |
| $j = \frac{v}{c}$ if $ad < v < bc$ | $j = b$ if $ad < v < bd$ | $j = b$ if $bc < v < ad$ |
| $j = b$ if $bc < v < bd$ | $i = a$ if $ac < v < ad$ | $j = b$ if $ad < v < bd$ |
| $i = a$ if $ac < v < ad$ | $i = \frac{v}{d}$ if $ad < v < bd$ | $i = a$ if $ac < v < bc$ |
| $i = \frac{v}{d}$ if $ad < v < bc$ | | $i = a$ if $bc < v < ad$ |
| $i = \frac{v}{d}$ if $bc < v < bd$ | | $i = \frac{v}{d}$ if $ad < v < bd$ |

To obtain the values of $m$ and $n$, we can first run a Monte Carlo Risk Simulation of the two independent distributions, then multiply them to obtain the joint distribution, and from this joint distribution, we obtain the left tail 0.01% value and set this as $m$. The value of $n$ is the left tail VaR% (e.g., 99.95%) value. The second integral when run based on this range, will return the CDF percentile of the OPCAR VaR. Alternatively, as previously described, the Bisection Method can be used to obtain the lowest value of $m$ by performing iterative searches such that the CDF returns valid results at 0.01% and then a second search is performed to identify the upper range or $n$, where the resulting $n$ that makes the integral equal to the user-specified VaR%, i.e., the OPCAR value.

Finally, for low lambda values, the algorithm still runs but will be a lot less accurate. Recall in Equation 2 that $\sum_{k>n} \frac{\lambda^k e^{-\lambda}}{k!} F_Y\left(\frac{t}{k}\right) < \delta$ where $\delta$ signifies the level of error precision (the lower the value, the higher the precision and accuracy of the results). The problem is, with low $\lambda$ values, both $k$ and $n$, which depend on $\lambda$, will also need to be a low value. This means that in the summand there

would be an insufficient number of integer intervals, making the summation function less accurate. For best results, $\lambda$ should be between 5 and 100.

## Convoluted Simulation vs. Simulation Multiplication

The next experiment is to look at the convolution simulation of two distributions, specifically, a Poisson discrete distribution to model the frequency of occurrence, and a Lognormal continuous distribution to model the impact of each occurrence. On the one hand, the test is applied by simulating these two independent distributions for 100,000 trials each, and in each trial, we multiply the results of the two simulated values. This will then generate a probability distribution of 100,000 losses. On the other hand, we will convolute these two probability distributions into a single simulation process to generate the sale 100,000 losses. Figures 9– 12 illustrate the results of these two competing simulation approaches.

In Figure 9, the same Lognormal distribution with a mean of 1,000 and standard deviation of 100 is used, while we vary the Poisson distribution, starting with a $\lambda$ (the average number of incidents) parameter of 10, 15, 20, all the way to 20,000. The analysis proceeds by the simple multiplication of the simulated distributions. A second parallel experiment was run with a convolution simulation. We then compare a few distributional moments of the results of these two methods, such as the mean, interquartile range (the first and third quartiles), and 90% confidence interval (5th and 95th percentiles). Their mean absolute deviations (MAD) were then computed. We see a heteroskedastic trend in the MAD values as $\lambda$ increases.

Conversely, Figure 10 shows the situation where the Poisson distribution is held constant, with a $\lambda$ of 100, while we vary the Lognormal distribution, where the mean is changed from 10, 50, 100, to 15,000,000 to model the impact of a loss distribution. We can see a more homoskedastic spread of MADs, regardless of the loss amounts.

Figure 11 shows what happens when we combine the two experiments, with varying Poisson and Lognormal distributional parameters. However, in all three experiments, we see that the first moments, the mean and median of all the combinations, tend to be similar, whether we ran a multiplicative simulation or convoluted simulation. We can quickly conclude that only the spreads of the total loss distributions differ, whereas the expected values remain the same.

Lognormal (1000,100) with Changing Poisson (Lambda)

Mean, Interquartile Range,
and 90% Confidence Interval of
the Mean Absolute Deviation (MAD)
between Convolution Simulation
and Multiplicative Simulation

|      | 10    | 15    | 20     | 40     | 80    | 100    | 200    | 500    | 1K     | 5K     | 10K    | 20K    |
|------|-------|-------|--------|--------|-------|--------|--------|--------|--------|--------|--------|--------|
| Max  | 0.0%  | -4.9% | -2.2%  | -10.2% | -7.7% | -13.8% | -12.4% | -16.2% | -18.4% | -16.8% | -21.5% | -21.3% |
| 95%  | 0.0%  | -2.3% | -2.7%  | -3.0%  | -3.9% | -4.3%  | -5.0%  | -6.1%  | -6.5%  | -7.8%  | -8.0%  | -8.2%  |
| 75%  | 0.0%  | -1.9% | -2.9%  | -3.8%  | -5.0% | -5.6%  | -7.0%  | -8.7%  | -9.9%  | -11.8% | -12.2% | -12.6% |
| 25%  | 0.0%  | -1.2% | -1.3%  | -2.2%  | -3.5% | -3.6%  | -5.0%  | -6.7%  | -7.8%  | -9.2%  | -9.8%  | -10.0% |
| 5%   | 0.0%  | -6.9% | 16.2%  | -16.1% | -2.9% | -1.5%  | -20.0% | -30.8% | -19.3% | -32.5% | -28.2% | -34.4% |
| Min  | -0.3% | 9.6%  | -12.7% | 20.9%  | 9.4%  | 8.6%   | 29.1%  | 42.4%  | 32.6%  | 48.3%  | 44.8%  | 51.5%  |
| Mean | -0.3% | 0.1%  | 0.1%   | 0.0%   | 0.0%  | 0.0%   | 0.0%   | 0.0%   | 0.0%   | 0.1%   | 0.0%   | 0.0%   |

**Figure 9: Different Poisson Assumptions with Identical Lognormal Distributions**

Poisson (Lambda 100) with Changing Lognormal (x, 0.1x)

Mean, Interquartile Range, and 90% Confidence Interval of the Mean Absolute Deviation (MAD) between Convolution Simulation and Multiplicative Simulation

|      | 10   | 50     | 100    | 500    | 1K    | 5K    | 10K    | 50K    | 100K   | 1M    | 10M    | 15M    |
|------|------|--------|--------|--------|-------|-------|--------|--------|--------|-------|--------|--------|
| Max  | 0.0% | -8.2%  | -12.5% | -11.7% | -8.0% | -8.2% | -11.7% | -13.0% | -13.5% | -7.6% | -13.6% | -10.5% |
| 95%  | 0.0% | -4.3%  | -4.3%  | -4.0%  | -4.1% | -4.2% | -4.1%  | -4.1%  | -4.2%  | -4.2% | -4.2%  | -4.1%  |
| 75%  | 0.0% | -5.5%  | -5.5%  | -5.6%  | -5.7% | -5.5% | -5.4%  | -5.7%  | -5.5%  | -5.3% | -5.4%  | -5.7%  |
| 25%  | 0.0% | -4.0%  | -3.8%  | -3.7%  | -3.8% | -3.8% | -3.8%  | -3.9%  | -3.5%  | -3.7% | -3.9%  | -3.7%  |
| 5%   | 0.0% | -10.2% | -15.1% | 3.2%   | -4.9% | -7.4% | -1.9%  | -6.1%  | -5.4%  | -7.1% | -22.6% | -12.8% |
| Min  | 0.0% | 17.7%  | 22.2%  | 3.9%   | 12.1% | 14.5% | 9.1%   | 13.4%  | 12.4%  | 14.1% | 29.8%  | 19.9%  |
| Mean | 0.0% | 0.1%   | -0.1%  | 0.0%   | -0.1% | 0.0%  | 0.1%   | 0.0%   | 0.0%   | 0.1%  | -0.1%  | -0.1%  |

**Figure 10: Different Lognormal Assumptions with Identical Poisson Distributions**



Changing Poisson (Lambda) with Changing Lognormal (Mean, 0.1 Mean)

Mean, Interquartile Range, and 90% Confidence Interval of the Mean Absolute Deviation (MAD) between Convolution Simulation and Multiplicative Simulation

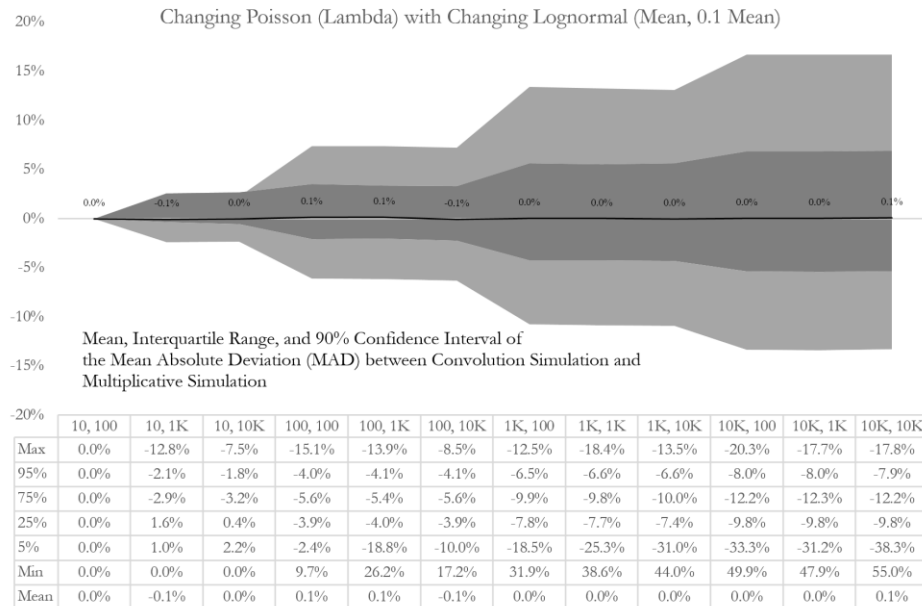|      | 10, 100 | 10, 1K | 10, 10K | 100, 100 | 100, 1K | 100, 10K | 1K, 100 | 1K, 1K | 1K, 10K | 10K, 100 | 10K, 1K | 10K, 10K |
|------|---------|--------|---------|----------|---------|----------|---------|--------|---------|----------|---------|----------|
| Max  | 0.0%    | -12.8% | -7.5%   | -15.1%   | -13.9%  | -8.5%    | -12.5%  | -18.4% | -13.5%  | -20.3%   | -17.7%  | -17.8%   |
| 95%  | 0.0%    | -2.1%  | -1.8%   | -4.0%    | -4.1%   | -4.1%    | -6.5%   | -6.6%  | -6.6%   | -8.0%    | -8.0%   | -7.9%    |
| 75%  | 0.0%    | -2.9%  | -3.2%   | -5.6%    | -5.4%   | -5.6%    | -9.9%   | -9.8%  | -10.0%  | -12.2%   | -12.3%  | -12.2%   |
| 25%  | 0.0%    | 1.6%   | 0.4%    | -3.9%    | -4.0%   | -3.9%    | -7.8%   | -7.7%  | -7.4%   | -9.8%    | -9.8%   | -9.8%    |
| 5%   | 0.0%    | 1.0%   | 2.2%    | -2.4%    | -18.8%  | -10.0%   | -18.5%  | -25.3% | -31.0%  | -33.3%   | -31.2%  | -38.3%   |
| Min  | 0.0%    | 0.0%   | 0.0%    | 9.7%     | 26.2%   | 17.2%    | 31.9%   | 38.6%  | 44.0%   | 49.9%    | 47.9%   | 55.0%    |
| Mean | 0.0%    | -0.1%  | 0.0%    | 0.1%     | 0.1%    | -0.1%    | 0.0%    | 0.0%   | 0.0%    | 0.0%     | 0.0%    | 0.1%     |

**Figure 11: Different Poisson and Different Lognormal Assumptions**

Figure 12 more clearly shows a visual of the spread. Specifically, we see the left column charts where the Poisson $\lambda$ parameter is allowed to increase. In all cases, the expected values are the same, but when convolution simulation is applied, the spread of the convoluted distribution is usually less wide and more conservative than the multiplicative simulation results. The difference is most obvious when the $\lambda$ is large.

The right column shows the results when the Poisson distribution is held constant while the Lognormal parameters are allowed to change. Just like in the homoskedastic chart, we see that the differential spreads between the multiplicative simulation and convolution simulation tend to be relatively stable (typically when the parameter is such that $5 \leq \lambda \leq 100$).
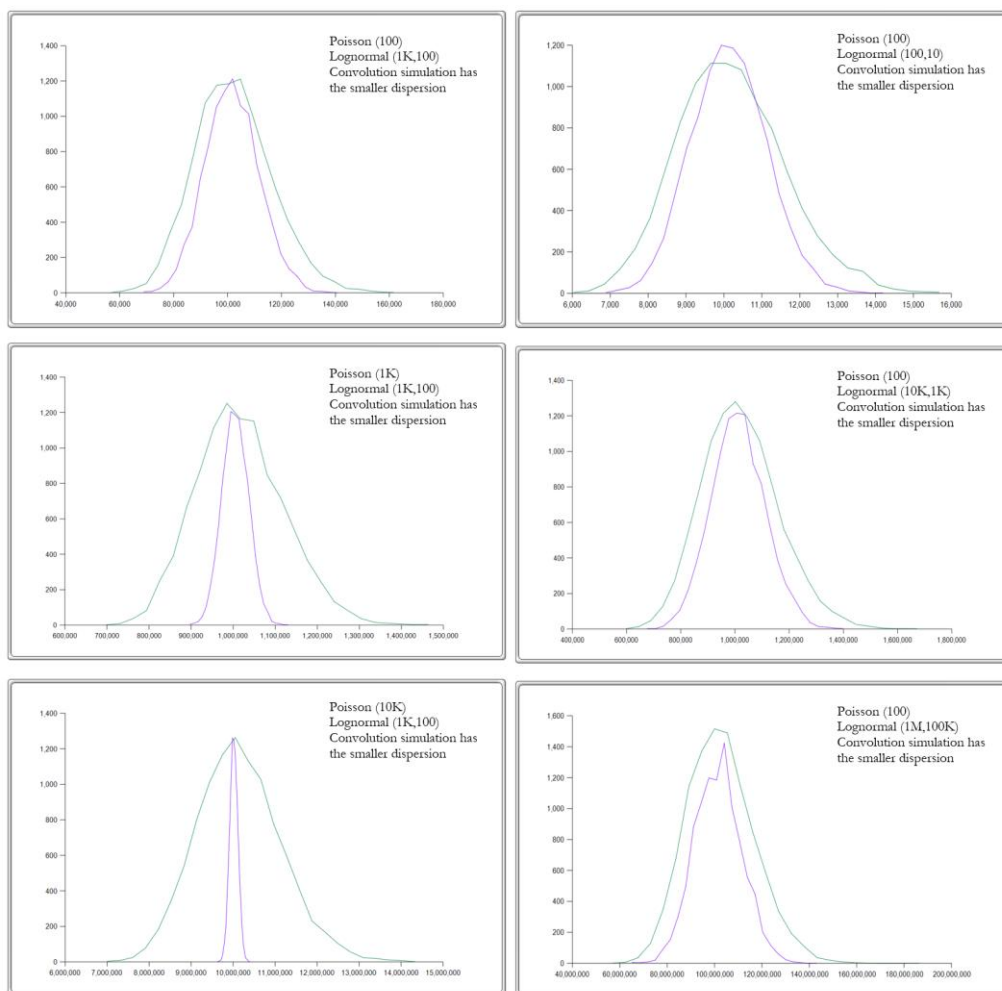


**Figure 12: Simulated Multiplication vs. Convolution Simulation**

## CAVEATS, CONCLUSIONS, AND RECOMMENDATIONS

Based on the theory, application, and empirical evidence above, one can conclude that the convolution of Frequency × Severity independent stochastic random probability distributions can be modeled using the algorithms outlined above as well as using Monte Carlo simulation methods. On average, the results from these two methods tend to converge with some slight percentage variation due to randomness in the simulation process and the precision depending on the number of intervals in the summand or numerical integration techniques employed. However, as noted, the algorithms described above are only applicable when the $\lambda$ parameter $5 \leq \lambda \leq 100$, else the approximation using numerical integration approach is required.

In contrast, Monte Carlo risk simulation methods using a standard multiplicative model vs. a convoluted simulation approach tend to yield the same expected value or central tendency results, i.e., the same mean values. But as $\lambda$ increases, the distributional width for the convoluted simulation approach tends to be more conservative and smaller, as compared to the multiplicative model. Monte Carlo risk simulation methods are the recommended path when it comes to modeling OPCAR and depending on whether a more conservative distribution is needed or a wider distribution is required, the relevant method can be applied.

## REFERENCES

Basel Committee on Banking Supervision (2014). Consultative Document on Operational Risk. Accessed: https://www.bis.org/publ/bcbs291.pdf

Gajic, Z. (2003). Linear Dynamic Systems and Signals. Prentice-Hall.

Georgiev, S.G. (2021). Convolutions. Theory of Distributions. DOI: 10.1007/978-3-030-81265-2_6.

Kaur, S. and Agrawal, R. (2022). Convolutional Networks. *Deep Learning Approaches to Cloud Security*. DOI: 10.1002/9781119760542

Luchko, Y.L. (2022). Convolution series and the generalized convolution Taylor formula. Fractional Calculus and Applied Analysis. DOI: 10.1007/s13540-021-00009-9.

Mun, J. (2016a). Advanced Analytical Models. ROV Press.

Mun, J. (2016b). Readings in Certified Risk Management. IIPER Press.

Oppenheim, A. and Willsky, A. (1996). Signals and Systems (Second Edition). PHI Press.

Peters, G., Targino, R., and Shevchenko, P. (2013). Understanding Operational Risk Capital Approximations: First and Second Orders. Journal of Governance and Regulation (print). 2. 58-78. 10.22495/jgr_v2_i3_p6.

Ramaré, O. (2022). The Convolution Method. Excursions in Multiplicative Number Theory. DOI: 10.1007/978-3-030-73169-4_8.

Sayeed, A. et al. (2021). A Deep Convolutional Neural Network Model for Improving WRF Simulations. Transactions on Neural Networks and Learning Systems. DOI: 10.1109/TNNLS.2021.3100902.

**Formatted:** Font: Italic

SimpliLearn (2022). AI and Machine Learning:: Convolutional Neural Network. Accessed: https://www.simplilearn.com/tutorials/deep-learning-tutorial/convolutional-neural-network

Siu, Y.M. (2022). Convolution Approach for Value at Risk Estimation. Review of Pacific Basin Financial Markets and Policies. DOI: 10.1142/S0219091522500059.