Proceedings of the 2019 Winter Simulation Conference
N. Mustafee, K.-H.G. Bae, S. Lazarova-Molnar, M. Rabe, C. Szabo, P. Haas, and Y.-J. Son, eds.

THE METALOG DISTRIBUTIONS AND EXTREMELY ACCURATE SUMS OF LOGNORMALS IN CLOSED FORM

Thomas W. Keelin

Keelin Reeds Partners
565 Oakfield Lane
Menlo Park, CA 94025, USA

Lonnie Chrisman

Lumina Decision Systems
1350 Dell Ave., Suite 107
Campbell, CA 95008, USA

Sam L Savage

Probability Management.org
3507 Ross Road
Palo Alto, CA 94303, USA

ABSTRACT

The metalog probability distributions can represent virtually any continuous shape with a single family of equations, making them far more flexible for representing data than the Pearson and other distributions. Moreover, the metalogs are easy to parameterize with data without non-linear parameter estimation, have simple closed-form equations, and offer a choice of boundedness. Their closed-form quantile functions (F^{-1}) enable fast and convenient simulation. The previously unsolved problem of a closed-form analytical expression for the sum of lognormals is one application. Uses include simulating total impact of an uncertain number N of risk events (each with iid [independent, identically distributed] individual lognormal impact), noise in wireless communications networks and many others. Beyond sums of lognormals, the approach may be directly applied to represent and subsequently simulate sums of iid variables from virtually any continuous distribution, and, more broadly, to products, extreme values, or other many-to-one change of iid or correlated variables.

1 INTRODUCTION

1.1 The Problem and Contribution

The sum of lognormally distributed random variables occurs naturally in nearly every quantitative field. But "one of the most surprising facts of elementary probability theory: almost nothing is known about the sum of lognormals" (Dufresne 2008). There is no known closed-form expression for the sum-of-lognormals probability functions, and the efficient and accurate calculation via simple numeral or simulation methods remains elusive despite an extensive history of research in the area.

We introduce a new algorithm for obtaining a simple closed-form representation for the distribution of a sum of lognormally distributed random variables. Our algorithm is more accurate than existing efficiently-computed methods when summing between 2 and 100 iid (independent, identically distributed) random variables, each having a σ (the standard deviation of $\ln x$) ranging from 0.04 to 1.5. Our algorithm is especially notable for its simplicity and ease of implementation, speed and accuracy. We obtain the algorithm by representing the sum of lognormals with a metalog distribution.

1.2 Metalog Distributions

Since the metalogs can approximate the shape of nearly any distribution (Keelin 2016) and are easy to parameterize, they are a natural choice for representing a sum-of-lognormal distribution, which otherwise has no closed form expression. Figure 1 shows a range of metalog representations of lognormal distributions that are virtually indistinguishable from the lognormals themselves.

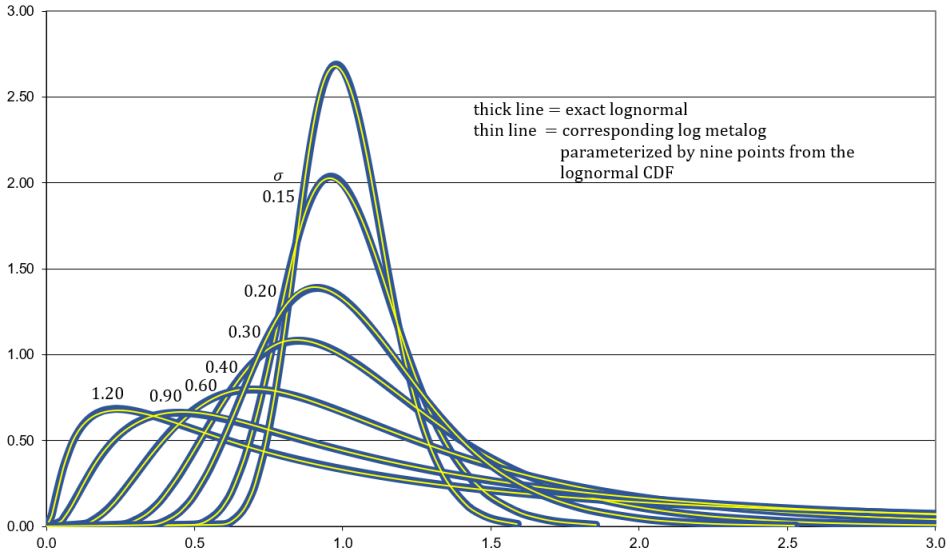

Figure 1: Exact lognormals vs. corresponding metalog distributions.

Metalogs are quantile-parameterized (Keelin and Powley 2010), with coefficients that can be determined by an ordinary least squares fit to CDF data, thus averting non-linear parameter-estimation procedures. For the lognormal distributions in Figure 1, we used nine points from the CDF to parameterize a nine-term semi-bounded metalog distribution. For *sums* of lognormal distributions, as discussed below, such CDF data may be predetermined by simulation.

Broadly, the metalog distributions offer significant improvements over the Pearson distributions (Pearson 1895, 1905, 1915) in terms of shape flexibility, ease of parameter estimation, simple closed-form quantile function for convenient simulation, simple closed-form PDF, and choice of boundedness (Keelin 2016). The sum of lognormals is one application.

1.3 Other Sum-of-Lognormal Approaches

The widespread need to sum lognormal distributions and the unsolved nature of this problem are widely documented. "The sum of correlated or even independent lognormal random variables, which is of wide interest in wireless communications, remains unsolved despite long-standing efforts" (Tellambura 2008). Moreover, "in finance, the most popular model of a stock price is the lognormal distribution"; in actuarial science (see also Zuanetti and Leite 2006), "individual claims are often well-represented by a lognormal distribution; what is the distribution of total claims?"; "The oldest and widest literature on the sum of lognormals is in engineering. Amplitudes of signals are modelled as lognormals. In telecommunications, engineers talk of the "logarithm of a sum of signals"; "the lognormal distributions has been used in many other fields: in economics, finance, reliability, biology, ecology, atmospheric sciences, geology" (Dufresne 2008).

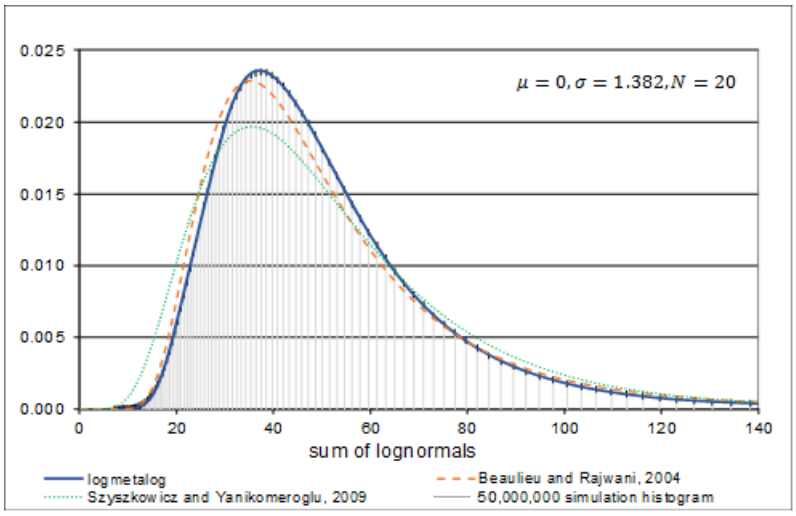

Figure 2. The sum-of-lognormals expressed as a metalog compared to previous approaches. N is the number of iid lognormal random variables each with $\ln x$ mean of μ and standard deviation σ.

There has been a wide range of prior efforts to characterize the sum of lognormals. These include using the Gram-Charlier series expansion (overviewed by Johnson et al. 1994); and numerous efforts to approximate the sum of lognormals with a lognormal itself (see Fenton 1960; Beaulieu and Xie 2004; Wu et al. 2005; Mehta et al. 2007; Szyszkowicz and Yanikomeroglu 2009; and Cobb et al. 2012, among others). Since the sum of lognormals is not actually lognormal, the accuracy of any such approximation is limited (Beaulieu and Rajwani 2004). In contrast, Beaulieu and Rajwani introduce a simple closed-form for the sum-of-lognormals that is "highly accurate" (their words) over a range of parameters commonly encountered in wireless communications. While we agree that their approximation is highly accurate, our approach is far more accurate. Figure 2 compares the histogram of a 50 million-trial simulation of sum of iid lognormals to three approximations: the metalog representation; the Beaulieu and Rajwani (2004) approximation; and Szyszkowicz and Yanikomeroglu (2009) lognormal approximation. The lognormal parameters used for this comparison are among those cited as relevant to wireless communications and used by Beaulieu and Rajwani to show the increased accuracy of their approximation relative to previous efforts. The Szyszkowicz approximation to the sum-of-lognormals in Figure 2 is the least accurate, followed by the Beaulieu approximation. In contrast, the metalog approximation is so extremely accurate that it is virtually indistinguishable from the histogram.

The metalog formulation has several additional advantages compared to Beaulieu and Rajwani. First, this approach may be applied to virtually any continuous distribution, not just lognormals. In addition, our formulation is a continuous function of the sum-of-lognormal parameters μ, σ, and N, whereas theirs works only for discrete values. Our formulation eliminates parameter-estimation by using a simple linear relationship between simulated data and coefficients, whereas theirs requires a non-linear optimization to find their coefficients. Our formulation has entirely algebraic closed form expressions for CDF and PDF, whereas their CDF requires the normal look-up table. Our formulation can be extended to any desired degree of accuracy, whereas theirs, while highly accurate over the ranges investigated, is a fixed formulation with no clear path for obtaining additional accuracy. Finally, our formulation can be directly extended to include certain correlation or other dependence relationships, whereas theirs assumes independence and offers no path for considering dependence.

2 EXPRESSING THE SUM OF LOGNORMALS AS A METALOG

We use the example of Figure 2 to illustrate how our algorithm expresses the sum of lognormals in closed form as a metalog distribution. For the sum-of-lognormals parameters shown in Figure 2, we interpolate into a pre-compiled table of highly-accurate sum-of-lognormal quantile data to obtain the

nine-points shown in yellow in Figure 3. The y-values for each yellow dot correspond to the probability vector $\boldsymbol{y} = (0.001, 0.020, 0.100, 0.250, 0.500, 0.750, 0.900, 0.980, 0.999)$, and the interpolated quantile vector is $\boldsymbol{x_s} = (14.5, 20.5, 27.5, 34.8, 45.8, 61.5, 82.2, 126, 151)$.

We then use the points $(\boldsymbol{x_s}, \boldsymbol{y})$ to parameterize a nine-term semi-bounded metalog distribution (with lower bound zero) to obtain the sum-of-lognormals curves shown in Figures 2 and 3. The sum-of-lognormals quantile function (inverse CDF) M_{sLN} is given by

$$M_{sLN}(y; \mu, \sigma, N) = Ne^{\boldsymbol{\mu + M(y)}} \text{ for } 0 < y < 1 \tag{1}$$

where y is probability and $M(y)$ is the nine-term unbounded metalog quantile function (Keelin 2016)

$$M(y) = a_1 + a_2 \ln\left(\frac{y}{1-y}\right) + a_3(y-0.5)\ln\left(\frac{y}{1-y}\right) + a_4(y-0.5) + a_5(y-0.5)^2 + a_6(y-0.5)^2 \ln\left(\frac{y}{1-y}\right) + a_7(y-0.5)^3 + a_8(y-0.5)^3 \ln\left(\frac{y}{1-y}\right) + a_9(y-0.5)^4 \tag{2}$$

The quantile function is fully described by the nine numbers $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9)$, and this vector of numbers is computed from the vector $\boldsymbol{x}$ as

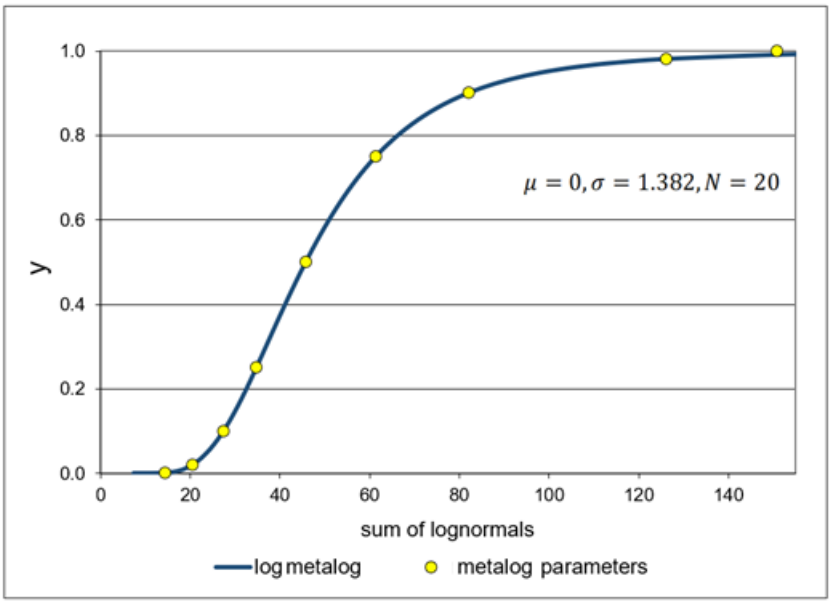

Figure 3. Sum-of-lognormals CDF and its quantile parameters.

$$\boldsymbol{a} = \boldsymbol{Y}^{-1} \ln \boldsymbol{x} \tag{3}$$

where, for scaling convenience, we define $\boldsymbol{x} = \boldsymbol{x_s}/\boldsymbol{N}$; $\boldsymbol{x}$ is the average-of-lognormal quantiles associated with $\boldsymbol{y}$; $\ln \boldsymbol{x}$ is the vector natural logarithms of each element of $\boldsymbol{x}$; and $\boldsymbol{Y}^{-1}$ is a constant 9x9 matrix shown in Table 1, which is the inverse of the matrix having the following entries applied to the above $\boldsymbol{y}$ vector

$$Y_{k,j} = \begin{cases} 1 & k = 1 \\ \ln\left(\frac{y_j}{1-y_j}\right) & k = 2 \\ (y_j - 0.5)^{\frac{k-1}{2}} \ln\left(\frac{y_j}{1-y_j}\right) & k = 3,5,7,9 \\ (y_j - 0.5)^{\frac{k}{2}} & k = 4,6,8 \end{cases} \tag{4}$$

Table 1: $\boldsymbol{Y^{-1}}$ for $\boldsymbol{y} = (.001, .020, .100, .250, .500, .750, .900, .980, .999)$.

0.000	0.000	0.000	0.000	1.000	0.000	0.000	0.000	0.000
-1.425	5.403	-7.763	4.893	0.000	-4.893	7.763	-5.403	1.425
2.856	-11.255	19.408	-19.572	17.126	-19.572	19.408	-11.255	2.856
5.789	-22.155	32.770	-23.449	0.000	23.449	-32.770	22.155	-5.789
-11.600	46.156	-81.925	93.797	-92.853	93.797	-81.925	46.156	-11.600
4.696	-19.684	29.471	-18.733	0.000	18.733	-29.471	19.684	-4.696
-13.042	61.114	-107.988	81.474	0.000	-81.474	107.988	-61.114	13.042
-9.411	41.009	-73.678	74.933	-65.706	74.933	-73.678	41.009	-9.411
26.136	-127.320	269.969	-325.896	314.222	-325.896	269.969	-127.320	26.136

Since $\boldsymbol{Y^{-1}}$is constant, an implementation can precompile or even hard code $\boldsymbol{Y^{-1}}$. While one could well use other choices for $\boldsymbol{y}$ (and by implication $\boldsymbol{Y^{-1}}$)**,** these $\boldsymbol{y}$ values provide a reasonable spread of points on the distribution. We do not consider other choices for $\boldsymbol{y}$ in this paper. In the example of Figure 3, the coefficients are $\boldsymbol{a} = (3.82, 0.109, 0.484, 0.620, -1.82, 0.282, -0.601, -1.27, 3.20)$. Since the sum-of-lognormals quantile function M_{sLN} has the same number of metalog terms as the number of data that parameterize it (nine in this case), the quantile function M_{sLN} is guaranteed to run through all of its CDF parameters $(\boldsymbol{x}, \boldsymbol{y})$ exactly.

By differentiating the quantile function M_{sLN} with respect to y, one obtains $\frac{dx}{dy}$. The reciprocal of this quantity is the sum-of-lognormals density function $m_{sLN} = \frac{dy}{dx}$.

$$\begin{aligned} m_{sLN}(y) &= \tfrac{1}{N} m(y) e^{-\mu - M(y)} && \text{for } 0 < y < 1, \\ &= 0 && \text{for } y = 0, \end{aligned} \tag{5}$$

where y is probability, $M(y)$ is (2), and $m(y)$ is the unbounded metalog density function (Keelin 2016).

$$\begin{aligned} m(y) = \Bigg[& \frac{a_2}{y(1-y)} + a_3\left(\frac{y-0.5}{y(1-y)} + \ln\left(\frac{y}{1-y}\right)\right) + a_4 \\ & + 2a_5(y-0.5) + a_6\left(\frac{(y-0.5)^2}{y(1-y)} + 2(y-0.5)\ln\left(\frac{y}{1-y}\right)\right) \\ & + 3a_7(y-0.5)^2 + a_8\left(\frac{(y-0.5)^3}{y(1-y)} + 3(y-0.5)^2\ln\left(\frac{y}{1-y}\right)\right) + 4a_9(y-0.5)^3 \Bigg]^{-1} \end{aligned}$$

The density function m_{sLN} is illustrated in Figure 2. To be a valid probability distribution, it must be the case that $m(y) > 0$ for all $y \in (0,1)$. While this condition is satisfied in Figure 2, there is no guarantee that it will be for an arbitrary $\boldsymbol{a}$-coefficient vector. When this condition is violated, we say that the $\boldsymbol{a}$ coefficients are *infeasible*.

Akin to Taylor series, metalogs with more terms provide more shape flexibility. We could have chosen other numbers of terms (e.g. 7, 10, 15) and achieved broadly similar results. We chose nine terms in this case because this number provides sufficient shape flexibility to capture sum-of-lognormals shape nuances without the unnecessary overhead of longer equations. We note that a nine-term semi-bounded metalog has eight shape parameters. By way of comparison, the beta distribution, widely regarded as among the most flexible of the Pearson distributions, has only two shape parameters.

3 GENERALIZATION TO A RANGE OF LOGNORMAL PARAMETERS

Applying the method of Section 2 to other values of sum-of-lognormal parameters σ and N yields equally satisfactory results. Figure 4 shows several more examples where the metalog-based sum-of-lognormal PDFs are visually indistinguishable from their respective 50-million-simulation histograms.

Note that sum-of-lognormals parameter μ does not affect shape but only scale of the x-axis. Since its effect is already accounted for in equations (1) and (5), there is no need to run various cases for μ. All that remains then to provide a general closed-form solution for the sum of lognormals is to specify $\boldsymbol{x} = \boldsymbol{x}(\sigma, N)$ in closed form as a function of σ and N.

We begin by pre-calculating the vector $\boldsymbol{x}(\sigma, N)$ for a range of discrete values of σ and N to within 10^{-4} of the true value for all entries. We devoted a lot of computation time to obtaining accurate entries for this table, available online (Analytica 2019; Keelin 2019) with a subset σ rows shown in Table 2 for illustration. An implementation of our algorithm does not need to repeat this work, but can simply import our pre-computed tables. The $\boldsymbol{x}$-vector of quantiles in each cell of this table was computed using a fast Fourier transform of 2^{14} points of the probability density graph of one log normal, raising each frequency component to the Nth power, and this using the fast inverse Fourier transform to obtain the PDF graph for the sum of lognormal distribution. This was then integrated to obtain the quantiles shown in this table. We also computed every cell using Monte Carlo simulation with 50 million trials as a potential approach and to validate the numbers, but settled on the Fourier approach since it seemed to be about 100 times more accurate with much less computation time.

What about values of σ and/or N that do not appear in the table but that are "in-between" values that do appear? One could of course set up and run a new Fourier transform analysis or Monte Carlo simulation for such values and apply the method of Section 2. However, these take considerable time, memory and effort. We would prefer an instant, closed-form expression for $\boldsymbol{x}(\sigma, N)$ that works for both values that appear in the table and for those in between. If the values of σ and N, respectively, that do appear are sufficiently "close to each other," a simple approach, which we will follow, is to use an interpolated $\boldsymbol{x}$ for the "in between" values. This approach however brings two new complexities into play. How can we ensure *a priori* that an interpolated $\boldsymbol{x}$ is feasible? How accurate is the interpolation relative to a new simulation? We address these questions in the subsequent sections.

4 FEASIBILITY OF INTERPOLATED QUANTILE PARAMETERS

As detailed in Keelin (2016), metalog distributions are subject to a feasibility condition which an arbitrary set of parameters $(\boldsymbol{x}, \boldsymbol{y})$ may or may not satisfy: PDF $m(y) > 0$ for all $y \in (0,1)$. Generally, this condition must be checked for any given $(\boldsymbol{x}, \boldsymbol{y})$ rather than assumed to be true. Taking vector $\boldsymbol{y} = (0.001, 0.020, 0.100, 0.250, 0.500, 0.750, 0.900, 0.980, 0.999)$ as fixed as discussed in Section 2, we have determined that all $\boldsymbol{x}$'s in our online table (Analytica 2019; Keelin 2019) are feasible. However, we cannot ensure the feasibility of interpolated quantile vectors without additional machinery. Since this additional machinery applies to all distributions in the metalog family, not just to those we are using for the sum of lognormals, we provide it as Appendix 1.

Summarizing Appendix 1 as it relates to the sum of lognormals, we first prove that for any given $\boldsymbol{y}$, the set of feasible $\boldsymbol{x}$'s is convex for unbounded metalog distributions. This implies that a linear interpolation among feasible $\boldsymbol{x}'s$ is feasible. For the *semibounded* metalogs we use for the sum of lognormals, $z(x) = \ln(x)$ is *unbounded*-metalog distributed (Keelin 2016). Thus, a linear interpolation among feasible $\ln(\boldsymbol{x})$'s is feasible, and a particular form of the interpolation equation that guarantees feasibility is implied.

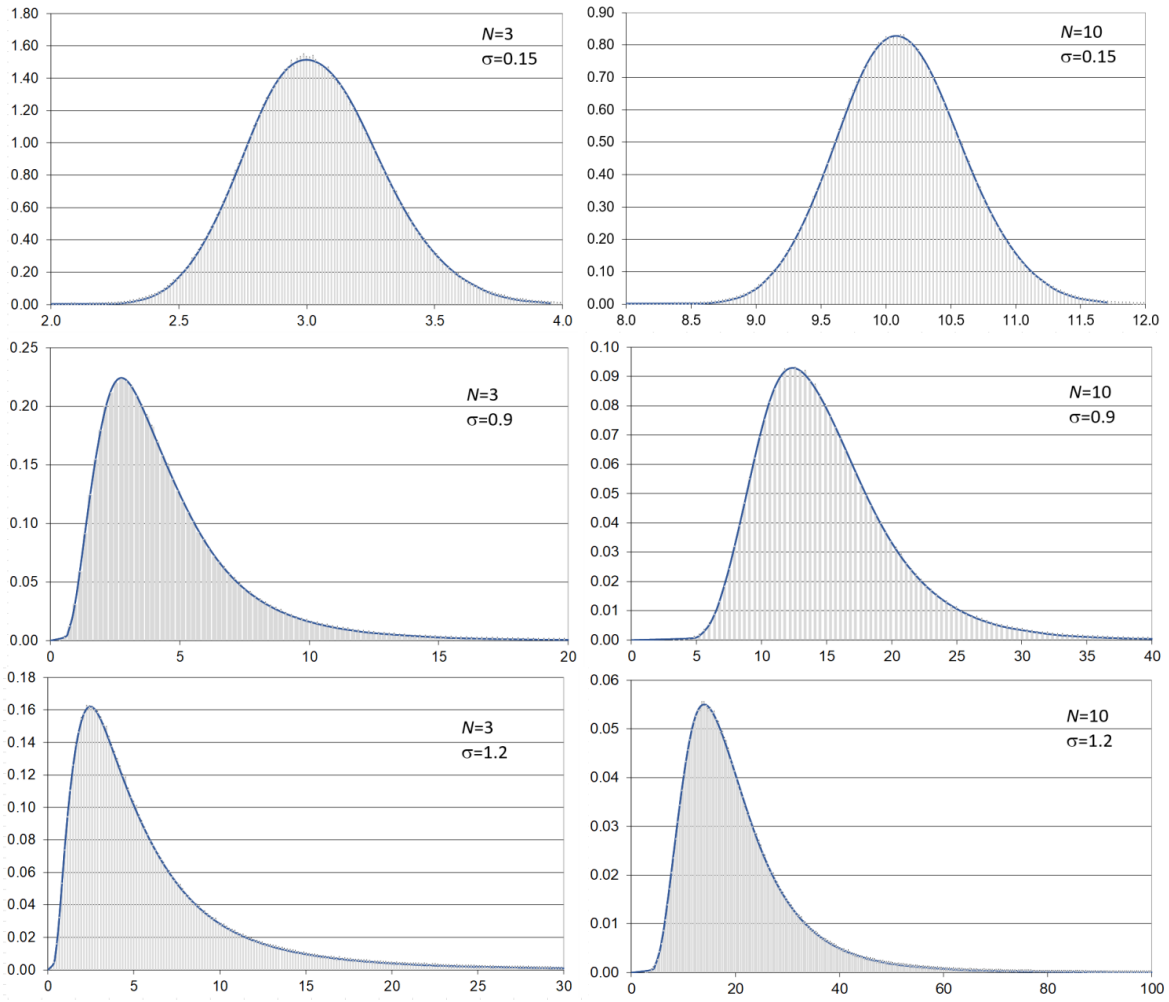

Figure 4: Metalog-based sum-of-lognormal PDFs vs. 50-million simulation histograms.

For "in between" values of σ and N, we calculate an interpolated $\boldsymbol{x}$ that is guaranteed to be feasible as follows. Let i and j denote a row and column of our online table such that $\sigma_{i-1} \le \sigma \le \sigma_i$ and $N_{j-1} \le N \le N_j$. Then, with interpolation weights $w_\sigma = (\sigma - \sigma_{i-1})/(\sigma_i - \sigma_{i-1})$ and $w_N = (N - N_{j-1})/(N_j - N_{j-1})$, the guaranteed-feasible interpolated $\boldsymbol{x}$ is a weighted product of feasible quantile vectors

$$\boldsymbol{x} = \boldsymbol{x}_{\boldsymbol{i-1,j-1}}^{(\mathbf{1}-\boldsymbol{w}_\sigma)(\mathbf{1}-\boldsymbol{w}_N)} \boldsymbol{x}_{\boldsymbol{i-1,j}}^{(\mathbf{1}-\boldsymbol{w}_\sigma)\boldsymbol{w}_N} \boldsymbol{x}_{\boldsymbol{i,j-1}}^{\boldsymbol{w}_\sigma(\mathbf{1}-\boldsymbol{w}_N)} \boldsymbol{x}_{\boldsymbol{i,j}}^{\boldsymbol{w}_\sigma \boldsymbol{w}_N} \tag{6}$$

where the $\boldsymbol{x}_{\boldsymbol{i,j}}$ denotes the nine-number quantile vector in row i, column j. With feasibility guaranteed, we now summarize the above algorithm and then discuss the accuracy of such interpolations.

		N																	
σ	y	1	2	3	4	5	6	12	14	16	18	20	60	70	80	90	100
	0.001	0.884	0.917	0.932	0.941	0.947	0.951	0.966	0.968	0.970	0.972	0.973	0.985	0.986	0.987	0.988	0.988
	0.020	0.921	0.944	0.954	0.960	0.965	0.968	0.977	0.979	0.980	0.982	0.983	0.990	0.991	0.992	0.992	0.993
	0.100	0.950	0.965	0.971	0.975	0.978	0.980	0.986	0.987	0.988	0.989	0.989	0.994	0.995	0.995	0.995	0.996
	0.250	0.973	0.981	0.985	0.987	0.989	0.990	0.993	0.994	0.994	0.994	0.995	0.997	0.998	0.998	0.998	0.998
0.04	0.500	1.000	1.000	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001	1.001
	0.750	1.027	1.020	1.016	1.014	1.013	1.012	1.009	1.008	1.008	1.007	1.007	1.004	1.004	1.004	1.004	1.003
	0.900	1.053	1.037	1.031	1.027	1.024	1.022	1.016	1.015	1.014	1.013	1.012	1.007	1.007	1.007	1.006	1.006
	0.980	1.086	1.060	1.049	1.043	1.038	1.035	1.025	1.023	1.022	1.020	1.019	1.011	1.011	1.010	1.009	1.009
	0.999	1.132	1.092	1.075	1.064	1.058	1.052	1.037	1.034	1.032	1.030	1.029	1.017	1.016	1.015	1.014	1.013
	0.001	0.201	0.335	0.421	0.482	0.528	0.565	0.695	0.722	0.743	0.762	0.778	0.916	0.932	0.944	0.955	0.964
	0.020	0.344	0.493	0.578	0.634	0.676	0.709	0.817	0.838	0.855	0.870	0.883	0.986	0.997	1.006	1.014	1.020
	0.100	0.514	0.658	0.732	0.780	0.814	0.839	0.922	0.938	0.950	0.961	0.970	1.042	1.049	1.055	1.060	1.064
	0.250	0.704	0.826	0.883	0.918	0.942	0.960	1.015	1.025	1.033	1.039	1.045	1.088	1.092	1.096	1.098	1.101
0.52	0.500	1.000	1.064	1.088	1.101	1.109	1.115	1.129	1.131	1.133	1.134	1.135	1.142	1.142	1.142	1.143	1.143
	0.750	1.420	1.373	1.343	1.323	1.308	1.297	1.258	1.250	1.244	1.239	1.234	1.198	1.194	1.191	1.189	1.187
	0.900	1.947	1.730	1.626	1.563	1.520	1.487	1.387	1.369	1.354	1.342	1.332	1.252	1.244	1.237	1.232	1.228
	0.980	2.909	2.328	2.082	1.939	1.845	1.776	1.573	1.538	1.510	1.487	1.468	1.324	1.310	1.299	1.290	1.282
	0.999	4.985	3.503	2.930	2.615	2.413	2.271	1.872	1.806	1.755	1.713	1.679	1.429	1.406	1.387	1.372	1.360
	0.001	0.147	0.274	0.360	0.423	0.473	0.513	0.660	0.690	0.716	0.738	0.757	0.923	0.942	0.957	0.970	0.981
	0.020	0.280	0.434	0.526	0.589	0.637	0.674	0.802	0.827	0.848	0.866	0.881	1.009	1.023	1.034	1.044	1.052
	0.100	0.452	0.612	0.699	0.755	0.795	0.827	0.928	0.948	0.963	0.976	0.988	1.079	1.088	1.096	1.103	1.108
	0.250	0.658	0.804	0.875	0.919	0.949	0.972	1.042	1.055	1.065	1.074	1.081	1.137	1.143	1.148	1.151	1.154
0.62	0.500	1.000	1.090	1.125	1.144	1.156	1.165	1.187	1.190	1.193	1.195	1.196	1.207	1.207	1.208	1.208	1.209
	0.750	1.519	1.480	1.451	1.429	1.412	1.399	1.354	1.345	1.337	1.331	1.326	1.281	1.276	1.272	1.269	1.266
	0.900	2.213	1.955	1.829	1.751	1.696	1.655	1.526	1.503	1.484	1.469	1.455	1.352	1.341	1.333	1.326	1.320
	0.980	3.572	2.803	2.473	2.281	2.153	2.060	1.785	1.737	1.700	1.669	1.643	1.449	1.430	1.415	1.403	1.393
	0.999	6.786	4.624	3.780	3.316	3.019	2.810	2.225	2.131	2.056	1.996	1.947	1.595	1.562	1.537	1.516	1.499
	0.001	0.010	0.047	0.097	0.148	0.198	0.247	0.486	0.550	0.608	0.661	0.709	1.255	1.333	1.399	1.457	1.509
	0.020	0.046	0.148	0.249	0.341	0.422	0.495	0.814	0.891	0.959	1.019	1.073	1.627	1.699	1.760	1.812	1.858
	0.100	0.146	0.347	0.510	0.642	0.752	0.845	1.212	1.294	1.364	1.425	1.480	1.990	2.053	2.104	2.148	2.186
	0.250	0.364	0.686	0.906	1.069	1.197	1.300	1.676	1.754	1.819	1.875	1.924	2.350	2.398	2.438	2.471	2.499
1.5	0.500	1.000	1.478	1.743	1.917	2.042	2.138	2.445	2.501	2.547	2.584	2.616	2.856	2.879	2.897	2.912	2.924
	0.750	2.750	3.243	3.434	3.529	3.583	3.615	3.662	3.660	3.655	3.649	3.643	3.530	3.511	3.495	3.480	3.467
	0.900	6.835	6.728	6.512	6.318	6.152	6.010	5.457	5.337	5.235	5.146	5.069	4.370	4.288	4.221	4.165	4.116
	0.980	21.750	17.753	15.589	14.173	13.151	12.368	9.809	9.327	8.933	8.604	8.323	6.072	5.840	5.653	5.498	5.368
	0.999	101.578	71.090	57.176	48.886	43.265	39.149	26.812	24.671	22.965	21.567	20.396	11.741	10.928	10.285	9.760	9.323

Table 2: Quantile parameters $\boldsymbol{x}$ for a range of σ and $(\mu = 0)$. The full table (Analytica 2019; Keelin 2019) contains 16 σ spaced geometrically from 0.04 to 1.5 (i.e., σ=0.04, 0.07, 0.11, 0.16, 0.215, 0.27, 0.34, 0.42, 0.52, 0.62, 0.74, 0.88, 1.04, 1.22, 1.44 and 1.5).

5 ALGORITHM SUMMARY

Pulling this all together, our algorithm for obtaining a closed-form metalog representation that approximates a sum-of-lognormal distribution is as follows.

Given:

- N = The number of independent iid lognormal random variables to sum. An integer.
- σ = The standard deviation of $\ln x_i$ for each iid lognormal component. A positive scalar.
- μ = The mean of $\ln x_i$ for each iid lognormal component. A positive scalar.

Output: (The parameters for final representation)

- $\boldsymbol{a} = (a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9)$ = a vector of metalog coefficients

Steps:

1. Use (N, σ) to lookup the 9-number **x** vector in our online table (Analytica 2019; Keelin 2019). When your N or σ lands between entries listed, interpolate using Equation (6).
2. Compute the metalog coefficient vector $\boldsymbol{a}$ using Equation (3) where $\boldsymbol{Y}^{-1}$ is the matrix in Table 1.
3. Use **a** in Equations (1) and (5) to determine any point on the quantile or probability density curves for this sum of lognormal distribution.

6 ACCURACY OF INTERPOLATED METALOGS

Following (Keelin 2016), we characterize the overall accuracy of our algorithm by the Kolmorogov-Smirnoff (K-S) deviation between the interpolated metalog and the true distribution (maximum difference in CDF across all x values). We measured this accuracy by computing an estimate of the true CDF to very high accuracy, using the same computationally intensive Fourier transform technique used to compute the quantile tables, then comparing the x-values at 1000 equally spaced quantiles to the cumulative probability of the interpolated metalog at these 1000 points. We tested at 10,889 combinations of N and σ, at every N from 2 to 100 and 110 different values of σ, including the midpoints of the σ values used in the quantile table. The maximum K-S deviation measured across all N and σ was 0.0098, which occurred at $N = 100$, $\sigma = 0.46$. The average K-S over the 10,890 CDFs measured was 0.0038.

The algorithm's limitations in accuracy (although impressively low) stem from three sources:

- Representation error: How accurately does the metalog approximate a sum-of-lognormal distribution given exact values for the 9 quantiles?
- Interpolation error: How accurate are the interpolated quantiles?
- Precompiled quantile table errors: How accurate are the pre-computed values in our online table?

To study how each of these contribute the overall accuracy, we examine each in turn.

To measure how accurately the 9-term metalog fits the shapes of sum-of-lognormals over our target range of $2 \le N \le 100$ and $0.04 \le \sigma \le 1.5$, we examined the K-S deviation for each combination of N and σ in our online table, since these are the test cases that involve no interpolation. The worst-case deviation (at $\sigma = 1.3, N = 2$) was 0.0014, with an average deviation over all cells of 0.00035. This confirms that the representation error due to our choice of a 9-term metalog is impressively low, accounting for about one-tenth the overall K-S deviation. We note that had this been a dominant source of error, it could be reduced by increasing the number of terms of the metalog.

The next source of error comes from interpolation error – how closely do the interpolated quantiles match the true quantiles at the "in between" values of σ and N. Since the K-S deviation of representation error is roughly one-tenth of the overall K-S deviation, it is reasonable to attribute 90% to interpolation error. Since this originates from inaccuracies in the nine interpolated quantiles, we measured the accuracy of this interpolation. The interpolation would be exact if the quantiles values were a geometric function of N and σ combinations. From our highly accurate Fourier calculations, we extracted the 9 quantile levels, $\boldsymbol{x}_{true}$, at each of the 10,890 interpolated (N, σ) combinations. Then from the interpolated quantiles, $\boldsymbol{x}_{\boldsymbol{interp}}$, we computed the interpolation accuracy element-by-element as one minus the relative error

$$accuracy = 1 - \frac{Abs\left(x_{interp} - x_{true}\right)}{x_{true}}$$

The minimum accuracy was 98.2%, average accuracy was 99.87%. The interpolation accuracy can, of course, be increased by increasing the resolution of the pre-calculated table.

Even though we tested at a very large number of (N, σ) combinations, we note that testing at only a finite number of interpolated points does not cover all possible cases. In principle, there could be less-accurate cases that wouldn't show up unless they were tested. Nevertheless, given the practical limitations, we believe this a is very reasonable estimate of interpolation accuracy.

To estimate the accuracy of our precompiled table quantiles, we repeated the Fourier transform calculations using both 2^{13} and 2^{14} points in each graph (with the same number of points in the frequency spectrum) and used the difference in the 9 computed quantiles to estimate the accuracy. We repeated this for each of the cells appearing in Table 2. We believe this to be a credible measure of the accuracy of the 2^{13}-point Fourier technique, but since our estimates were all computed with a 2^{14}-point Fourier, our estimate should be on the conservative side. We found that in all cases, the absolute error came out to be less than 10^{-4} (one digit more than printed in our online table) and better than 6 significant digits in over 98% of the cells. So this accounts for between 0.1% and 0.001% of the overall divergence.

7 CONCLUSIONS

We have presented an extremely accurate method for summing iid lognormal distributions. Their CDF's and PDF's, as shown in Figure 5 for a given μ and σ and a range of N, can be instantly calculated in closed form without simulation. We have made available a library with functions that implement the algorithm, the pre-computed quantile tables, and other supporting materials (Analytica 2019; Keelin 2019).

We are optimistic that our metalog-based approach should be applicable to other problems where a hard-to-characterize distribution results from a transformation on a continuous base distribution. Beyond sums of lognormals, the approach may be directly applied to represent and subsequently simulate sums of iid variables from virtually any continuous distribution, and, more broadly, to products, extreme values, or other many-to-one change of iid or correlated variables. Potential applications must have a small number of input parameters (we had just 2, N and σ) so that tables can be reasonably pre-compiled. To be within the metalog's scope, the metalog approximation at each target cell must be feasible.

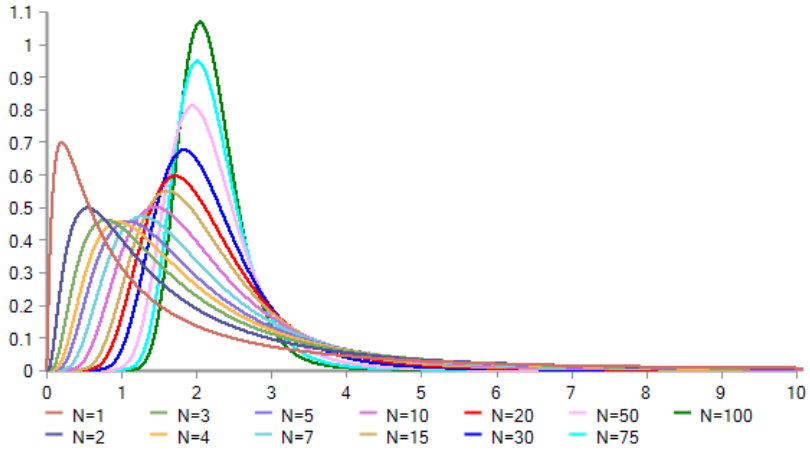

Figure 5: Average of N Lognormals for a Range of N ($\mu = 0,\ \sigma = 1.256$).

A APPENDIX I. PROOF THAT THE SET OF FEASIBLE METALOG QUANTILE PARAMETERS IS CONVEX

First, we prove the convexity of feasible parameters of the unbounded metalog distribution and thus that linearly interpolated quantile vectors are feasible. Then we show the implications of this theorem for feasible interpolation methods for semi-bounded and bounded metalog distributions.

Theorem 1 Convexity of feasible parameters of unbounded metalog distributions. Let $M_n(y; \boldsymbol{x}, \boldsymbol{y})$ be a n-term metalog distribution with quantile parameters $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} = (x_1, \ldots, x_n)$ and $\boldsymbol{y} = (y_1, \ldots, y_n)$. For any given $\boldsymbol{y}$, the set of feasible $\boldsymbol{x}$ is convex.

Proof of Theorem 1 The metalog distribution $M_n(y; \boldsymbol{x}, \boldsymbol{y})$ is a quantile-parameterized distribution (QPD) as defined Keelin and Powley (2011). Keelin and Powley (in a proof due to Brad Powley) proved that, for any QPD, the set of feasible $\boldsymbol{a}$ coefficients is convex. For any given $\boldsymbol{y}$, $\boldsymbol{x}$ is a linear transformation of $\boldsymbol{a}$: $\boldsymbol{x} = \boldsymbol{Y}\mathbf{a}$, where $\boldsymbol{Y}$ is Equation (4) (Keelin 2016). Convexity is preserved under a linear transformation. □

The importance of Theorem 1 is that if quantile vectors $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are feasible, then any convex combination of them (such as $\boldsymbol{x}_3 = \alpha\boldsymbol{x}_1 + (1-\alpha)\boldsymbol{x}_2$, where $0 \le \alpha \le 1$) is feasible. No further feasibility check is required. This result also implies that feasible parameters of metalog-distributed transforms may be interpolated to yield other feasible parameters of that transform.

Corollary 1 Feasible interpolation of parameters of metalog transforms. Let $z(x)$ be metalog distributed, where z is strictly increasing and invertible. If $\boldsymbol{x_1}$ and $\boldsymbol{x_2}$ are feasible parameterizations of the z-metalog, then for any $0 \le \alpha \le 1$, $\boldsymbol{x}_3$ is feasible, where

$$\boldsymbol{x}_3 = z^{-1}[\alpha z(\boldsymbol{x_1}) + (1-\alpha) z(\boldsymbol{x_2})]$$

Proof of Corollary 1 Since z is invertible, the above equation can be rewritten, $z(\boldsymbol{x}_3) = \alpha z(\boldsymbol{x_1}) + (1-\alpha) z(\boldsymbol{x_2})$. Since $z(\boldsymbol{x_1})$ and $z(\boldsymbol{x_2})$ are feasible, $z(\boldsymbol{x_3})$ is feasible by Theorem 1 and thus $\boldsymbol{x_3}$ is feasible. □

The z functions for bounded and semi-bounded metalog are shown in Table 3, where b_l and b_u are upper and lower bounds.

Table 3: Transformation functions for interpolated quantile vectors.

distribution:	semi-bounded-lower metalog	semi-bounded-upper metalog	bounded metalog
z:	$\ln(x - \mathrm{b}_l)$	$-\ln(\mathrm{b}_u - x)$	$\ln\left(\frac{x - \mathrm{b}_l}{\mathrm{b}_u - x}\right)$

REFERENCES

Analytica. 2019. SoLN Supporting Materials. https://wiki.analytica.com/SoLN_paper_supporting_materials, accessed 29th April.

Beaulieu, N.C and Q. Xie, 2004. "An Optimal Lognormal Approximation to Lognormal Sum Distributions." Institute of Electrical and Electronic Engineers Transactions on Vehicular Technology, 53(2):479-489.

Beaulieu, N.C. and F. Rajwani. 2004. "Highly Accurate Simple Closed-Form Approximations to Lognormal Sum Distributions and Densities. Institute of Electrical and Electronic Engineers Communications Letters, 8(12):709-711.

Cobb, B.R., R. Rumi, and A. Salmerón. 2012. "Approximating the Distribution of a Sum of Log-Normal Random Variables." Statistics and Computing, 16(3):293-308.

Dufresne, D. 2008. "Sums of Lognormals." In Actuarial Research Conference, edited by C. Fuhrer and A. Shapiro, 1-6. Schaumburg, Illinois: Society of Actuaries.

Fenton, L. 1960. "The Sum of Log-Normal Probability Distributions in Scatter Transmission Systems." *Institute of Radio Engineers Transactions on Communications Systems*, 8(1):57-67.

Johnson, N.L., S. Kotz, and N. Balakrishnan. 1994. *Continuous Univariate Distributions, Vols. 1 and 2*. New York: John Wiley & Sons.

Keelin, T.W. 2016. "The Metalog Distributions." *Decision Analysis*, 13(4):243-277.

Keelin, T.W. 2019. The Metalog Distributions. http://metalogs.org/, accessed 28th April.

Keelin, T.W. and B.W. Powley. 2011. "Quantile-Parameterized Distributions." *Decision Analysis*, 8(3):206-219.

Mehta, N.B., J. Wu, A.F. Molisch, and J. Zhang. 2007. "Approximating a Sum of Random Variables with a Lognormal." *Institute of Electrical and Electronic Engineers Transactions on Wireless Communications*, 6(7).

Pearson, K. 1895. "Contributions to the Mathematical Theory of Evolution. II. Skew Variation in Homogeneous Material." *Philosophical Transactions of the Royal Society of London. A*, 186:343-414.

Pearson, K. 1901. "Mathematical Contributions to the Theory of Evolution. X. Supplement to a Memoir on Skew Variation. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 197:443-459.

Pearson, K. 1916. "Mathematical Contributions to the Theory of Evolution. XIX. Second Supplement to a Memoir on Skew Variation." *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 216:429-457.

Szyszkowicz, S.S. and H. Yanikomeroglu. 2009. "Limit Theorem on the Sum of Identically Distributed Equally and Positively Correlated Joint Lognormals." *Institute of Electrical and Electronic Engineers Transactions on Communications*, 57(12).

Tellambura, C. 2008. "Bounds on the Distribution of a Sum of Correlated Lognormal Random Variables and Their Application." *Institute of Electrical and Electronic Engineers Transactions on Communications*, 56(8).

Wu, J., N.B. Mehta, and J. Zhang. 2005. "Flexible Lognormal Sum Approximation Method. In *GLOBECOM '05. Institute of Electrical and Electronic Engineers Global Telecommunications Conference, 2005* 6:3413-3417. Institute of Electrical and Electronic Engineers

Zuanetti, D., C. Diniz, and J. Leite. 2006. "A Lognormal Model for Insurance Claims Data." *REVSTAT-Statistical Journal*, 4(2):131-142.

AUTHOR BIOGRAPHIES

TOM KEELIN is founder and Managing Partner of Keelin Reeds Partners, providing strategic decision consulting services based on decision-analysic modeling and simulation. As Chairman of Millennial Capital, LLC, he has serves as general partner for multiple successful real estate funds. In addition, Tom volunteers as Chief Research Scientist for ProbabilityManagement.org. Previously, as Worldwide Managing Director of the Strategic Decision Group, he led the client work for and co-authored the Harvard Business Review article "How SmithKline Beecham Makes Better Resource Allocation Decisions" (Mar-Apr '98). Through that work, he and his colleagues invented the portfolio-management standard which subsequently was adopted widely across life-sciences industry. Having recognized over his carrer in practice that tradiitonal probability distributions are poorly suited to modern needs, Tom developed The Metalog Distributions (Keelin, 2016). Tom is a Fellow of the Society of Decision Professionals, and a founder and director of the Decision Education Foundation. Tom holds three degrees from Stanford University: BA in Economics and MS and PhD in Engineering-Economic Systems. He can be reached at tomk@keelinreeds.com.

LONNIE CHRISMAN is the Chief Technology Officer at Lumina Decision Systems and leads the design and development of Analytica. He has a PhD in Machine Learning and Computer Science from Carnegie Mellon University and a BSEE from UC Berkeley. He has published in areas of Artificial Intelligence planning, neural network learing, robotics, Bayesian networks and computational biology. He can be reached at LChrisman@Lumina.com.

SAM L SAVAGE is Executive Director of ProbabilityManagement.org, a 501(c)(3) nonprofit devoted to the communication and calculation of uncertainty. The organization has received funding from Chevron, Lockheed Martin, PG&E, and others, and he is joined on the board by Harry Markowitz, Nobel Laureate in Economics. Dr. Savage is author of The Flaw of Averages: Why We Underestimate Risk in the Face of Uncertainty (John Wiley & Sons, 2009, 2012), and is an Adjunct Professor in Civil and Environmental Engineering at Stanford University. He is the inventor of the Stochastic Information Packet (SIP), an auditable data array for conveying uncertainty. He received his Ph.D. in computational complexity from Yale University. He can be reached at sam@probabilitymanagement.org.