

LAPLACE TRANSFORMS OF PROBABILITY DISTRIBUTIONS AND THEIR INVERSIONS ARE EASY ON LOGARITHMIC SCALES

A. G. ROSSBERG,* IIASA

Abstract

It is shown that, when expressing arguments in terms of their logarithms, the Laplace transform of a function is related to the antiderivative of this function by a simple convolution. This allows efficient numerical computations of moment generating functions of positive random variables and their inversion. The application of the method is straightforward, apart from the necessity to implement it using high-precision arithmetics. In numerical examples the approach is demonstrated to be particularly useful for distributions with heavy tails, such as lognormal, Weibull, or Pareto distributions, which are otherwise difficult to handle. The computational efficiency compared to other methods is demonstrated for an M/G/1 queueing problem.

Keywords: Moment generating function; Laplace transform; transform inversion; heavy tail

2000 Mathematics Subject Classification: Primary 44A10
Secondary 33F05; 44A35; 33B15

1. Introduction

Moment generating functions (MGFs) are frequently used in probability theory. However, computing an MGF from a given distribution, and, even more so, computing a distribution from a given MGF, can be challenging. Here, a new numerical method for these transformations is proposed. The method is particularly powerful for distributions that are well behaved on a logarithmic scale, e.g. for lognormal and other heavy-tailed distributions. Sums of such distributions are encountered in analyses of problems as diverse as radio communication [4], [6], [7], [24], tunnel junctions [21], turbulence [13], biophysics [18], [29], and finance [12], [19]. Sums of lognormals have so far been difficult to handle [4], [8], [19], [21]. Since the MGF is a variant of the Laplace transform, and since the theory applies to Laplace transforms in general, we shall first introduce it in this framework, and discuss its application in probability theory, including numerical examples, later on.

As shown for two examples in Figure 1(a) and (b), the graphs of the Laplace transform

$$[\mathcal{L}f](s) = \int_0^{\infty} e^{-st} f(t) dt = F(s)$$

of a function $f(t)$ and of its integral $\int_0^{1/s} f(t) dt$ with reciprocal upper bound $s > 0$ look quite similar. The similarity becomes even more striking when going over to logarithmic scales, i.e. when comparing $H(y) := F(e^y)$ with $h(y) := \int_0^{e^{-y}} f(t) dt$ (see Figure 1(c) and (d)). In

Received 21 August 2007; revision received 18 February 2008.

* Postal address: Evolution and Ecology Program, International Institute for Applied Systems Analysis (IIASA), Schlossplatz 1, 2361 Laxenburg, Austria. Email address: axel@rossberg.net

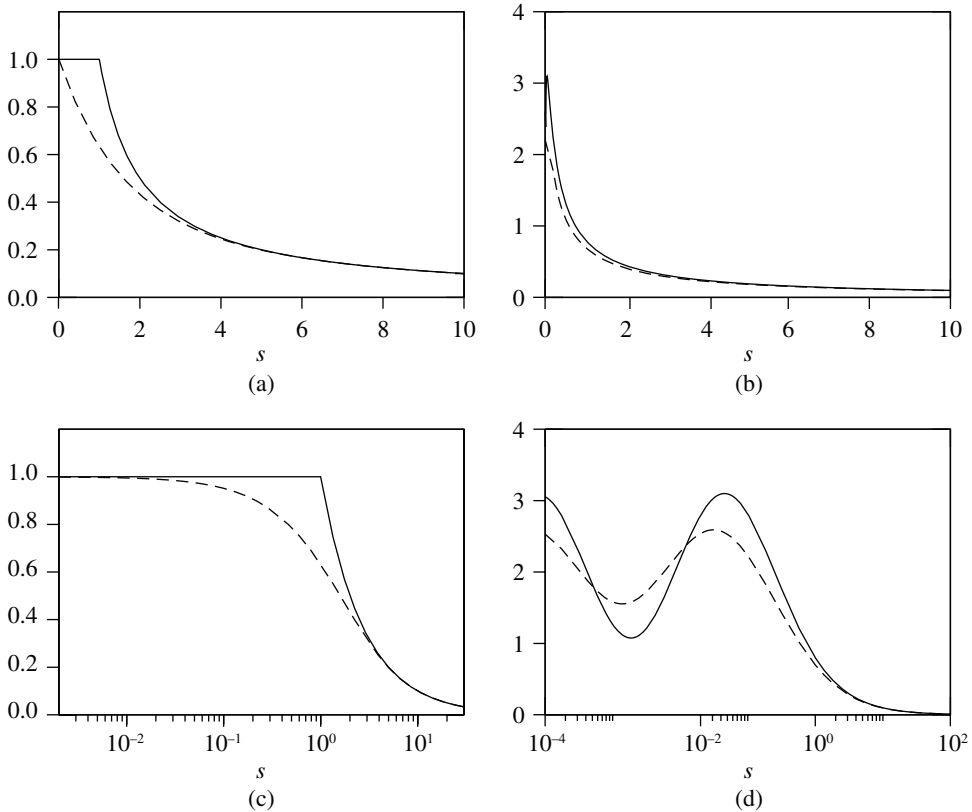


FIGURE 1: Examples comparing the integral $\int_0^{1/s} f(x) dx$ (solid line) with the Laplace transform (dashed line) of functions $f(x)$. (a) $f(x) = 1$ for $x < 1$, $f(x) = 0$ otherwise, (b) $f(x) = \sin[\ln(x + 1)]/x$, (c) $f(x)$ same as (a), but on semi-logarithmic axes, and (d) $f(x)$ same as (b), but on semi-logarithmic axes.

fact, as is shown below, $H(y)$ can be obtained from $h(y)$ by a simple convolution. Conversely, $h(y)$, and therefore the inverse of the Laplace transform, can be computed from $H(y)$ by a deconvolution, which can be efficiently implemented numerically using fast Fourier transforms (FFTs).

The problem of numerically inverting Laplace transforms has persistently attracted attention. Valkó and Vojta [28] compiled a list of over 1500 publications devoted to it dating from 1795 to 2003. Important methods used today are, for example, the Gaver–Stehfest method [14], [25] and its variants [27], the Euler algorithm [10], and the Talbot algorithm [26]. Abate and Whitt [3] recently compared these methods within a generalized formal framework.

2. General theory

2.1. Main theorem

To see that, on a logarithmic scale, the Laplace transform and integral of a function are related by a convolution, first recall the following simple fact not always stated precisely in the textbooks.

Lemma 1. Let $f: \mathbb{R}^{\geq 0} \rightarrow \mathbb{R}$ (or a corresponding linear functional) and assume that its Laplace transform $[\mathcal{L}f](s)$ is defined for all $s > 0$. Define $g(x) := \int_0^x f(t) dt$. Then

$$[\mathcal{L}f](s) = s[\mathcal{L}g](s) \quad \text{for all } s > 0. \tag{1}$$

Proof. Formally, this is, of course, just an integration by parts:

$$\int_0^T e^{-st} f(t) dt = e^{-sT} \int_0^T f(t) dt + s \int_0^T e^{-st} g(t) dt. \tag{2}$$

Subtle is only the question if the boundary term vanishes as $T \rightarrow \infty$. It does, because

$$\begin{aligned} \left| e^{-sT} \int_0^T f(t) dt \right| &= e^{-sT/2} \left| \int_0^T e^{-sT/2} f(t) dt \right| \\ &< e^{-sT/2} \left| \int_0^T e^{-st/2} f(t) dt \right|, \end{aligned} \tag{3}$$

and the last integral converges to $[\mathcal{L}f](s/2)$ as $T \rightarrow \infty$, which is finite by assumption, while the exponential factor goes to 0. Thus, (2) converges to (1) as $T \rightarrow \infty$.

The following theorem immediately follows.

Theorem 1. Under the conditions of Lemma 1, with $h(y) := g(e^{-y})$, $H(y) := [\mathcal{L}f](e^y)$, and $K(y) := \exp(-e^y)e^y$ for all $y \in \mathbb{R}$,

$$H = K * h, \tag{4}$$

where ‘*’ denotes the convolution operator.

Note that, in a less precise but more transparent notation, (4) reads as

$$[\mathcal{L}f](e^y) = [\exp(-e^y)e^y] * \int_0^{e^{-y}} f(z) dz.$$

Proof of Theorem 1. Apply Lemma 1 and then change variables by letting $T = e^{-x}$:

$$\begin{aligned} H(y) &= e^y [\mathcal{L}g](e^y) \\ &= e^y \int_0^\infty \exp(-e^y T) g(T) dT \\ &= \int_{-\infty}^\infty \exp(-e^{y-x}) e^{y-x} g(e^{-x}) dx \\ &= \int_{-\infty}^\infty K(y-x) h(x) dx \\ &= [K * h](y). \end{aligned}$$

2.2. Representation in Fourier space

The convolution in (4) and its inversion are efficiently computed in Fourier space. Define, for any function $g(x)$, its Fourier transform $\tilde{g}(k)$ such that

$$\tilde{g}(k) = \int_{-\infty}^\infty e^{-ikx} g(x) dx, \quad g(x) = \int_{-\infty}^\infty \frac{e^{ikx}}{2\pi} \tilde{g}(k) dk.$$

(The function $\tilde{g}(k)$ might have to be interpreted as a linear functional.) Then

$$\tilde{H}(k) = \tilde{K}(k)\tilde{h}(k).$$

The Fourier transform of the kernel can be obtained by a change of variables $u = e^y$ as

$$\begin{aligned} \tilde{K}(k) &= \int_{-\infty}^{\infty} K(y)e^{-iky} dy \\ &= \int_{-\infty}^{\infty} \exp(-e^y)e^y e^{-iky} dy \\ &= \int_0^{\infty} e^{-u}u^{-ik} du \\ &= \Gamma(1 - ik). \end{aligned}$$

Thus,

$$\tilde{h}(k) = \frac{\tilde{H}(k)}{\Gamma(1 - ik)}.$$

At this point it is interesting to note that $1/\Gamma(1 + z)$ is an entire function [20] and that the coefficients d_n of its Taylor series

$$\frac{1}{\Gamma(1 + z)} = \sum_{n=0}^{\infty} d_n z^n$$

are known to be given by $d_0 = 1$ and the recursion [15]

$$(n + 1)d_{n+1} = \gamma_E d_n + \sum_{k=1}^n (-1)^k \zeta(k + 1)d_{n-k} \quad \text{for } n \geq 0,$$

where γ_E is Euler’s constant and $\zeta(x) = \sum_{n=1}^{\infty} n^{-x}$ denotes Riemann’s zeta function. The coefficients d_n have been shown to decay to 0 faster than $(n!)^{-(1-\varepsilon)}$ for any $\varepsilon > 0$ [17]. This might sometimes be sufficiently fast to obtain a convergent series representation of $h(x)$ as

$$\begin{aligned} h(x) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{ikx} \sum_{n=0}^{\infty} d_n (-ik)^n \tilde{H}(k) dx \\ &= \sum_{n=0}^{\infty} d_n \left(-\frac{d}{dx}\right)^n H(x). \end{aligned}$$

Numerically, this expansion has already been demonstrated to yield accurate results [22]. It would be interesting to understand under which conditions and at what computational cost this pointwise Laplace inversion formula will generally converge.

2.3. Application to probability theory

The MGF $M(t)$ of a real-valued random variable (RV) U with cumulative distribution function (CDF) $P(u) = P[U \leq u]$ is defined as the expectation value $M(t) = E e^{tU} = \int_{-\infty}^{\infty} e^{tu} dP(u)$. Let $p(u) = dP(u)/du$ denote the probability density of U (possibly defined in the functional sense), and assume that U attains only positive values, i.e. $p(u) = 0$ for $u \leq 0$.

Then, obviously, $M(-t) = [\mathcal{L}p](t)$. Note that $M(-t) = E e^{-tU} \leq 1$ for $t \geq 0$, even if all moments of U are undefined. Hence, the corresponding Laplace transform is always defined for positive arguments, and the above considerations apply with

$$f(u) = p(u), \quad h(y) = P(e^{-y}), \quad \text{and} \quad H(y) = M(-e^y).$$

Equation (4) then becomes

$$M(-e^y) = K(y) * P(e^{-y}). \tag{5}$$

Invertibility of the Laplace transform of $p(u)$ implies that knowledge of $M(t)$ for negative arguments is sufficient to recover $p(u)$. The standard procedure to compute moments of U directly from derivatives of $M(t)$ at $t = 0$ turns out to be numerically difficult when using (4), but efficient methods exist to compute moments of the logarithm of U directly from $M(t)$ [22]. These can be used, among others, to construct lognormal approximations of a RV from a given MGF.

The fact that a convolution of densities on logarithmic scales corresponds to a multiplication of RVs implies a close relationship between multiplication and Laplace transforms, which is reflected by the following theorem.

Theorem 2. *Let X and Λ be two independent RVs such that X is always positive and Λ is exponentially distributed with $E \Lambda = 1$. Denote by M_X the MGF of X and by P_Z the CDF of*

$$Z := \frac{\Lambda}{X}. \tag{6}$$

Then

$$M_X(-t) = 1 - P_Z(t) \quad \text{for any } t > 0. \tag{7}$$

Proof. Define $z = \ln Z$, $\lambda = \ln \Lambda$, and $x = \ln X$, and denote by $P_X(t)$ the CDF of X . Taking logarithms on both sides of (6) yields $z = \lambda + (-x)$. The density of λ equals $K(\lambda)$, since $e^{-\lambda} d\Lambda = \exp(-e^\lambda)e^\lambda d\lambda = K(\lambda) d\lambda$. The CDF of $-x$ is $1 - P_X(e^{-x})$. By the rule for the distribution of sums of independent RVs, the CDF of z , i.e. $P[z < y] = P_Z(e^y)$, equals $K(y) * (1 - P_X(e^{-y})) = 1 - K(y) * P_X(e^{-y})$. Thus, by (5), $1 - P_Z(e^y) = K(y) * P_X(e^{-y}) = M_X(-e^y)$, which proves (7) with $y = \ln t$.

3. Numerical examples

3.1. General considerations

For the application to probability theory described above, it is not difficult to see that $h(y), H(y) \rightarrow 0$ as $y \rightarrow \infty$ and $h(y), H(y) \rightarrow 1$ as $y \rightarrow -\infty$. Thus, for any numerical range of integration $[y_0, y_1]$, direct FFTs of $h(y)$ or $H(y)$ would lead to artifacts because these functions are not periodic.

With sufficiently small y_0 and sufficiently large y_1 , approximate periodicity can, for example, be achieved by splitting $h(y) = h_1(y) + h_0(y)$, with

$$h_0(y) = \frac{y - y_1}{y_0 - y_1}.$$

Then $H = K * h$ is obtained as the sum $H(y) = H_1(y) + H_0(y)$, where $H_1 = K * h_1$ is computed numerically using FFTs over $[y_0, y_1]$ and

$$H_0(y) = [K * h_0](y) = \frac{y - y_1 + \mathcal{Y}_E}{y_0 - y_1}.$$

The calculations used to evaluate the numerical examples hereafter are therefore characterized by four parameters: the lower y_0 and upper y_1 end of the interval taken into account, the number N of equally spaced mesh points in this interval at which $h(y)$ and $H(y)$ are computed, and the numerical accuracy ε at which computations are done. High-precision arithmetics are needed, just as for other methods of Laplace transform inversion [1].

While the implementation of the algorithm is straightforward when an arbitrary-precision FFT library is available, there is no systematic method, yet, for setting the parameters to achieve a desired accuracy of the output. Things to keep in mind when choosing the parameters are that $-y_0$ and y_1 need to be large enough to avoid aliasing and N must be sufficiently large to resolve $h(y)$ on the scale $(y_1 - y_0)/N$. For an appropriate choice of ε , note the following: after performing the desired manipulations of $H(y)$, we obtain another generating function H' for which the corresponding distribution is to be computed. The deconvolution of $H'(y)$ (or $H'_1(y)$) to obtain $h'(y)$ (or $h'_1(y)$) is, as any deconvolution, sensitive to numerical errors in $H'(y)$. A simple rule for suppressing artifacts from such errors is to set all the Fourier modes of $\tilde{h}'_1(k) = H'_1(k)/\tilde{K}(k)$ to 0 that lie beyond the absolute minimum of the power spectrum of $h'_1(y)$, estimated by some local averaging of $|\tilde{h}'_1(k)|^2$. This procedure is justified, because smoothness arguments suggest that $|\tilde{h}'_1(k)|^2 \rightarrow 0$ as $|k| \rightarrow \infty$. The numerical value of the minimum power gives a coarse estimate of the achievable accuracy (squared), and can be used to tune ε .

3.2. Sum of two lognormals

As a first simple example, the CDF of the sum of two lognormal RVs $\exp(\xi_1)$ and $\exp(2\xi_2)$ with independent standard normal ξ_1 and ξ_2 is computed. The MGF of the sum is obtained as the product of the MGF of the two distributions. Using the method described above with $-y_0 = y_1 = 40$, $N = 256$, and $\varepsilon = 10^{-20}$, the CDF of the sum can be computed to a numerical accuracy better than 2×10^{-12} , as is illustrated in Figure 2. The precise CDF used for comparison was obtained by a direct numerical convolution of the density of $\exp(\xi_1)$ with the CDF of $\exp(2\xi_2)$ using a high-precision integration algorithm. The comparison shows that

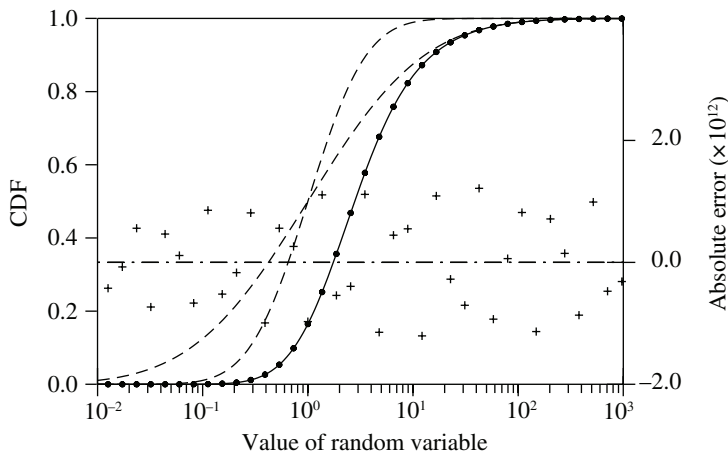


FIGURE 2: The CDF of the sum $\exp(\xi_1) + \exp(2\xi_2)$ with independent standard normal ξ_1 and ξ_2 , computed using the methods described here. The black circles correspond to the numerical grid, the solid line is a 5th order spline interpolation, and the dashed lines represent the CDF of the two addends. Absolute numerical errors are indicated by the crosses.

the numerical error is of the same order of magnitude at any point of the CDF. This observation suggests that the accuracy at which the CDF and the MGF were computed by the algorithm can be estimated a posteriori by the precision at which $H(y)$ respectively $h(y)$ converge to 0 or 1 in the tails.

3.3. Sums of 111 Weibull distributions

The next two examples are numerically more challenging. Consider sums $S := \sum_{i=1}^R X_i$ of $R = 111$ independent and identically distributed RVs X_i following a Weibull distribution:

$$W(x) := P[X_i > x] = \exp(x^\beta).$$

First, the CDF corresponding to a shape parameter $\beta = 0.5$ is computed. Parameters for the numerical Laplace transform are chosen as $-y_0 = y_1 = 250$, $N = 2^{13}$, and $\varepsilon = 10^{-50}$. The tail of the complementary CDF of the sum is shown in Figure 3. For $S > 2000$, numerical errors (aliasing) dominate the result. The size of these errors (less than 10^{-16}) indicates the numerical accuracy achieved. Near its expectation value 222, the distribution of the sum is well described by a normal distribution (standard deviation 47.1, dashed line in Figure 3). For large S , it approaches the asymptote $P[S > x] \sim RW(x)$ (dash-dotted line in Figure 3). To check if the result is also correct between these limits, the conditional Monte Carlo method proposed by Asmussen and Kroese [5] was used. Specifically, $P[S_i > x]$ was estimated by the average of 1000 pseudo-random realizations of

$$RW\left(\max\left(X_1, \dots, X_{R-1}, x - \sum_{i=1}^{R-1} X_i\right)\right). \tag{8}$$

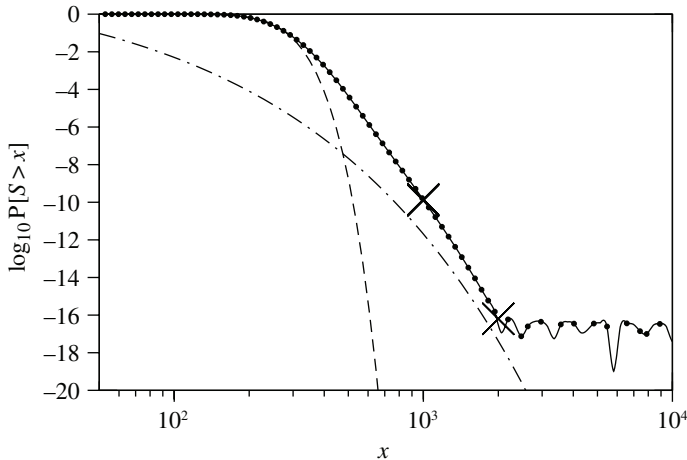


FIGURE 3: Complementary CDF of the sum S of 111 independent Weibull RVs with shape parameter $\beta = 0.5$, as computed by the method described here (for negative numerical results, the absolute value is shown). The black circles correspond to the numerical grid, the solid line is a 5th order spline interpolation on a logarithmic scale, the dashed line corresponds to a normal approximation, and the dash-dotted line corresponds to the tail approximation valid when the sum is dominated by the single largest term. Simulation results are marked by crosses. Five independent results were obtained at $x = 1000$ and $x = 2000$, which are visually indistinguishable here.

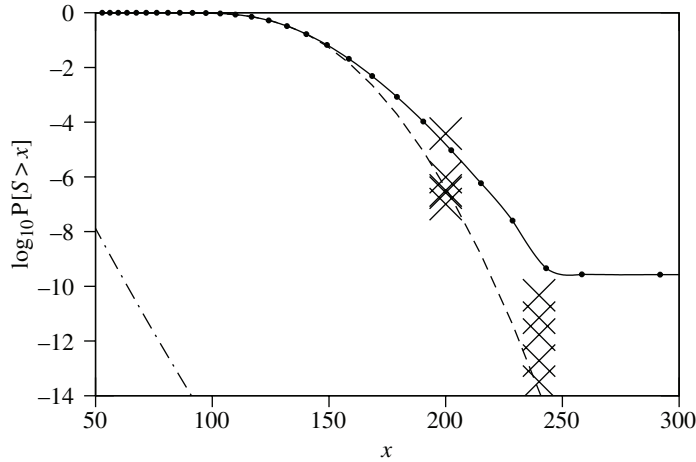


FIGURE 4: Complementary CDF of the sum S of 111 Weibull RVs with shape parameter $\beta = 0.8$. Other details are as in Figure 3. Compared to Figure 3, the simulation results (crosses at $x = 200$ and $x = 240$) become inaccurate here.

As a consistency check, five realizations of this average were sampled. At $x = 1000$, values ranged from 1.17×10^{-10} to 1.47×10^{-10} , while the Laplace method yields 1.35×10^{-10} . Similarly, good agreement was achieved at $x = 2000$ (Figure 3), thus, confirming that the complementary CDF is accurate down to probabilities of 10^{-16} .

Next, the CDF for shape parameter $\beta = 0.8$ was evaluated. With the same values for y_0, y_1, N , and ε as above, errors from aliasing are now approximately 10^{-9} . In the range $x = 150$ to $x = 240$, as shown in Figure 4, neither the normal approximation nor the asymptotic tail formula are reliable. Additionally, Monte Carlo simulations based on (8) (again averaged over 1000 samples) now scatter strongly. Equation (8) is known to become inefficient for large x when $\beta > 0.585$ [5]. An independent verification of the CDF found here for $\beta = 0.8$ might therefore be difficult.

3.4. Waiting times in M/P/1 queues

Finally, we take up one of the numerical examples used recently to demonstrate a new method for computing waiting time distributions in M/G/1 queues with heavy-tailed service time distributions [23]. In the example considered here, the service times X_i follow a Pareto distribution of the form

$$\int_0^x B(t) dt = P[X_i < x] = 1 - (x + 1)^{-\alpha}.$$

This case is, for example, important for questions of network design [11]. The Laplace transforms of the distributions of waiting time $W(t)$ and service time $B(t)$ are known to be related as (see, e.g. [16])

$$[\mathcal{L}W](s) = \frac{(1 - \rho)s}{s - (1 - [\mathcal{L}B](s))\lambda}, \tag{9}$$

where λ is the arrival rate and $\rho = \lambda E X$ is the load. Formally, (9) corresponds to a mixture of sums of X_i with a geometric distribution of the number of terms. Equation (9) can be used

TABLE 1: Parameters, numerical results, and computation times for the CDF $F(t)$ of waiting times in M/P/1 queues. Parameters λ and α were chosen such that $\rho \approx 0.8$ [23]. Computation times are given in seconds normalized to a 1GHz clock frequency. Note that in [23] $F(t)$ was computed on 25 points linearly spaced between 0 and 250, while here 128 grid points geometrically spaced between 3.1×10^{-7} and 3.3×10^6 are used.

	Problem 1	Problem 2	Problem 3
λ	1.00	0.866666	0.8161616
α	2.25	2.083333	2.0202020
Current work			
$F(30)$	0.904	0.854	0.830
$F(100)$	0.983	0.964	0.952
CPU time $F(t)$	0.6	0.6	0.6
Reference [23]			
$F(30)$	0.904	0.854	–
$F(100)$	0.984	0.964	–
CPU time $F(30)$	4	9	–
CPU time $F(100)$	2.5	5.5	–
CPU time $F(t)$	4.5	22	35

to compute $W(t)$. The challenge [23] is to compute the cumulative waiting time distribution $F(t) = \int_0^t W(\tau) d\tau$ in the interval $0 \leq t \leq 250$ to an accuracy of 0.0005 for parameters α and λ as given in Table 1.

Two points need attention before applying the method introduced here to this problem. First, since $[\mathcal{L}B](s) \rightarrow 1$ as $s \rightarrow 0$, small inaccuracies in $\mathcal{L}B$ will produce artifacts when computing $\mathcal{L}W$ from (9). These artifacts were removed here by correcting the values of $[\mathcal{L}W](s)$, obtained numerically on the grid $s = s_k$ (with $s_k < s_{k+1}$ for all k) as follows: iterating from large to small s , $[\mathcal{L}W](s_i)$ was set to $[\mathcal{L}W](s_{i+1})$ if it was smaller than $[\mathcal{L}W](s_{i+1})$ and to 1 if it was larger than 1.

Second, the transformations suggested in Section 3.1 to obtain periodic functions for the FFTs fail for the back transformation if applied directly to $[\mathcal{L}W](s)$, because this function converges to $(1 - \rho)$ for $-y = \log(s) \rightarrow \infty$, and not to 0. One way to fix this is to follow [23] in going over to the waiting time distribution of the nonidle queue $W_b(t) = W(t)/\rho$ (for $t > 0$). This yields $[\mathcal{L}W_b](s) = ([\mathcal{L}W](s) - (1 - \rho))/\rho$, which has the desired limits. A corresponding relation holds for the CDF.

The required accuracy is comfortably reached by setting $-y_0 = y_1 = 15$, $N = 2^7 = 128$, and $\varepsilon = 10^{-30}$ for all three cases in Table 1. A 5th order spline was used to interpolate between grid points. The numerical results mostly coincide with [23]. For $F(100) = 0.983$ in problem 1, only the last digit differs from that of [23]. In calculations with more ambitious parameter settings, we obtain $F(100) = 0.98316\dots$, confirming the present result.

Marked differences can be found when comparing computation times. Calculations here were performed using a general-purpose mathematical command language on an UltraSPARC IV+ 1.5GHz CPU. The times given in Table 1 are adjusted to a 1GHz clock rate. Contrary to [23], no attempts were made to invert the Laplace transform for individual values of t . Instead, FFTs were used to compute convolutions and deconvolutions as above, which yields $F(t)$ over the full range of $t = e^y$. Depending on the problem, the improvement in computation

time over the already highly efficient methods of [23] was seven to 58 fold (Table 1). However, to be fair, we have to point out that the algorithms used there automatically determined the parameters required to reach a specified accuracy, while this was done by trial and error here. Refinements of the methods of [23], for example by using continued fraction representations of the forward transform [2], might also be possible.

4. Concluding discussion

The results of the foregoing section demonstrate that the method to compute and invert Laplace transforms proposed here can efficiently be applied to problems in probability theory. The method is particularly powerful in situations involving distributions with heavy tails, which are naturally represented on a logarithmic scale.

The most important application of the method is probably the computation of the CDF or sums, or mixtures of sums, of independent RVs. But correlated lognormal variables can be handled as well, provided that the correlation can be factored out of the sum. This is, for example, the case for the sum $S = \sum_{i=1}^R \exp(\xi_i)$, where the ξ_i follow a multivariate normal distribution such that $c = \text{cov}(\xi_i, \xi_j)$ is the same for all $i \neq j$ and $\text{var } \xi_i \geq c$ for all i . In an insurance setting this would correspond to the plausible assumption that, when the claim of client i was K times larger than usual, the claims of other clients can be predicted to be typically a factor $cK/\text{var } \xi_i$ higher than usual. To evaluate the CDF of S , note that S has the same distribution as $e^Y \sum_{i=1}^R \exp(X_i)$ for independent normally distributed X_i and Y such that $E X_i = E \xi_i$ and $E Y = 0$, and $\text{var } X_i = \text{var } \xi_i - c$ and $\text{var } Y = c$. The CDF of S can be obtained by first computing the CDF of $\sum_{i=1}^R \exp(X_i)$ (where R may be an RV or constant) and then incorporating the multiplication with e^Y as an addition on logarithmic scales, i.e. by a convolution of the distributions of the logarithms. Another type of sum of correlated lognormals with a factorizable structure is $\sum_{i=1}^R \exp(\sum_{j=1}^i \xi_j) = (\cdots ((\exp(\xi_1) + 1) \exp(\xi_2) + 1) \exp(\xi_3) \cdots)$ with independent, normal ξ_i . Such expressions occur in certain applications in finance, where the ξ_i represent relative changes in the value of an asset over subsequent time intervals.

Since computationally the most expensive steps of the method proposed here are computing the two FFTs for the Laplace transform and the two FFTs for the backward transform, the computational complexity of the algorithm increases as $N \log N$ with the number of mesh points. Most Laplace inversion algorithms working on linear scales appear to be at least of order N^2 (but see [9]). Open problems are how to choose optimal parameters for the numerical scheme and how this affects the complexity of the algorithm.

References

- [1] ABATE, J. AND VALKÓ, P. P. (2004). Multi-precision Laplace transform inversion. *Internat. J. Numerical Meth. Eng.* **60**, 979–993.
- [2] ABATE, J. AND WHITT, W. (1999). Computing Laplace transforms for numerical inversion via continued fractions. *INFORMS J. Comput.* **11**, 394–405.
- [3] ABATE, J. AND WHITT, W. (2006). A unified framework for numerically inverting Laplace transforms. *INFORMS J. Comput.* **18**, 408–421.
- [4] ABU-DAYYA, A. AND BEAULIEU, N. C. (1994). Outage probabilities in the presence of correlated lognormal interference. *IEEE Trans. Veh. Technol.* **43**, 164–173.
- [5] ASMUSSEN, S. AND KROESE, D. P. (2006). Improved algorithms for rare event simulation with heavy tails. *Adv. Appl. Prob.* **38**, 545–558.
- [6] BEAULIEU, N. C. AND RAJWANI, F. (2004). Highly accurate simple closed-form approximations to lognormal sum distributions and densities. *IEEE Commun. Lett.* **8**, 709–711.
- [7] BEAULIEU, N. C. AND XIE, Q. (2004). An optimal lognormal approximation to lognormal sum distributions. *IEEE Trans. Veh. Technol.* **53**, 479–489.

- [8] BEAULIEU, N. C., ABU-DAYYA, A. A. AND McLANE, P. J. (1995). Estimating the distribution of a sum of independent lognormal random variables. *IEEE Trans. Commun.* **43**, 2869–2873.
- [9] DEN ISEGER, P. (2006). Numerical transform inversion using Gaussian quadrature. *Prob. Eng. Inf. Sci.* **20**, 1–44.
- [10] DUBNER, H. AND ABATE, J. (1968). Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *J. Assoc. Comput. Mach.* **15**, 115–123.
- [11] DUFFIELD, N. G. AND WHITT, W. (2000). Network design and control using on/off and multilevel source traffic models with heavy-tailed distributions. In *Self-Similar Network Traffic and Performance Evaluation*, eds K. Park and W. Willinger, John Wiley, New York, pp. 421–445.
- [12] DUFRESNE, D. (2004). The log-normal approximation in financial and other computations. *Adv. Appl. Prob.* **36**, 747–773.
- [13] FRISCH, U. AND SORNETTE, D. (1997). Extreme deviations and applications. *J. Physique II* **7**, 1155–1171.
- [14] GAVER, D. P. (1966). Observing stochastic processes and approximate transform inversion. *Operat. Res.* **14**, 444–459.
- [15] GRADSHTEIN, I. AND RYZHIK, I. (1980). *Tables of Integrals, Series and Products*, 2nd edn. Academic Press, New York.
- [16] GROSS, D. AND HARRIS, C. M. (1998). *Fundamentals of Queueing Theory*, 3rd edn. John Wiley, New York.
- [17] LEIPNIK, R. R. (1991). On lognormal random variables. I. The characteristic function. *J. Austral. Math. Soc. Ser. B* **32**, 327–347.
- [18] LOPEZ-FIDALGO, J. AND SANCHEZ, G. (2005). Statistical criteria to establish bioassay programs. *Health Phys.* **89**, 333–338.
- [19] MILEVSKY, M. A. AND POSNER, S. E. (1998). Asian options, the sum of lognormals, and the reciprocal gamma distribution. *J. Financial Quant. Anal.* **33**, 409–422.
- [20] NIELSEN, N. (1906). *Handbuch der Theorie der Gammafunktion*. Teubner, Leipzig.
- [21] ROMEO, M., COSTA, V. D. AND BARDOU, F. (2003). Broad distribution effects in sums of lognormal random variables. *Europ. Phys. J. B* **32**, 513–525.
- [22] ROSSBERG, A. G., AMEMIYA, T. AND ITOH, K. (2008). Accurate and fast approximations of moment-generating functions and their inversion for log-normal and similar distributions. Submitted.
- [23] SHORTLE, J. F. *et al.* (2004). An algorithm to compute the waiting time distribution for the M/G/1 queue. *INFORMS J. Comput.* **16**, 152–161.
- [24] SLIMANE, S. B. (2001). Bounds on the distribution of a sum of independent lognormal random variables. *IEEE Trans. Commun.* **49**, 975–978.
- [25] STEHFEST, H. (1970). Algorithm 368: numerical inversion of Laplace transforms. *Commun. ACM* **13**, 47–49.
- [26] TALBOT, A. (1979). The accurate inversion of Laplace transforms. *J. Inst. Maths. Appl.* **23**, 97–120.
- [27] VALKÓ, P. AND ABATE, J. (2004). Comparison of sequence accelerators for the Gaver method of numerical Laplace transform inversion. *Comput. Math. Appl.* **48**, 629–636.
- [28] VALKÓ, P. P. AND VOJTA, B. L. (2003). The list. <http://www.pe.tamu.edu/valko/Nil/LapLit.pdf>.
- [29] VALLADE, M. AND HOUCHEMANDZADEH, B. (2003). Analytical solution of a neutral model of biodiversity. *Phys. Rev. E* **68**, 061902.