Probabilistic Scaling for the Numerical Inversion of Non-Probability Transforms

by

Gagan L. Choudhury[1] and Ward Whitt[2]
AT&T Laboratories

April 22, 1996
Revision: October 11, 1996

[1]Room 1L-238, Holmdel, NJ 07733-3030 (gagan@buckaroo.att.com)
[2]Room 2C-178, Murray Hill, NJ 07974-0636 (wow@research.att.com)

Abstract

It is known that probability density functions and probability mass functions can usually be calculated quite easily by numerically inverting their transforms (Laplace transforms and generating functions, respectively) with the Fourier-series method, but other more general functions can be substantially more difficult to invert, because the aliasing and roundoff errors tend to be more difficult to control. In this paper we propose a simple new scaling procedure for non-probability functions that is based on transforming the given function into a probability density function or a probability mass function and transforming the point of inversion to the mean. This new scaling is even useful for probability functions, because it enables us to compute very small values at large arguments with controlled relative error.

Subject classifications: Mathematics, functions: scaling for numerical transform inversion. Queues, algorithms: scaling for numerical transform inversion.

Keywords: numerical inversion of transforms, Fourier-series method, Laplace transforms, generating functions, scaling.

Numerical transform inversion is proving to be an effective tool for calculating quantities of interest in operations research models. It is especially useful for queueing models, because many probability distributions are readily available in the form of transforms; e.g., see Choudhury, Lucantoni and Whitt [9]. Abate and Whitt [3] and Choudhury, Lucantoni and Whitt [8] have shown that probability distributions can be computed remarkably easily from their transforms by numerical inversion using the Fourier-series method. This is especially true for cumulative distribution functions (cdf's) and probability mass functions (pmf's), because they are nonnegative and bounded above by 1, but it also tends to be true for probability density functions (pdf's), because they are also nonnegative and typically are bounded above away from the origin as well (which suffices, see Section 2).

However, even for stochastic models, there is interest in calculating more general functions from transforms. For example, Choudhury and Lucantoni [7] develop an algorithm for calculating moments, of high as well as low order, from a moment generating function, and Choudhury, Leung and Whitt [5], [6] calculate performance measures in product-form models by numerically inverting the generating function of the normalization constant. Both moments and normalization constants tend to grow (or decline) geometrically fast. Calculating these non-probability functions by numerical inversion has proved to be substantially more difficult than calculating probability distributions. In these cases, the inversion algorithms required developing an appropriate way to scale the transforms before performing the inversion. The scaling algorithms that have been developed are effective, but they are somewhat ad hoc.

The purpose of this paper is to propose a systematic scaling algorithm for a large class of non-probability functions. Our main idea is to scale so that the original function is transformed into a pdf (with a Laplace transform) or a pmf (with a generating function), and the inversion point is transformed into the mean. It is actually not necessary to think probabilistically, but it can help intuition. More generally, the mean can be regarded as a center of gravity. The main point is that our scaling algorithm transforms a potentially difficult inversion problem into one that tends to be more manageable.

In addition to providing an alternative to the scaling algorithms in Choudhury and Lucantoni [7] and Choudhury, Leung and Whitt [5], [6], our scaling algorithm here provides an alternative to other methods proposed for developing general Fourier-series inversion algorithms, e.g., see Honig and Hirdes [12] and Piessens and Huysmans [16].

The scaling is also important for probability distributions themselves when we want to compute very small values at large arguments with controlled relative error. For example, we want to do this in order to calculate the asymptotic parameters describing the way tail probabilities decay. Choudhury and Lucantoni [7] and Abate, Choudhury, Lucantoni and Whitt [1] showed that it is possible to calculate these asymptotic parameters from high-order moments. The scaling here provides a way to calculate the asymptotic parameters directly from the tail probabilities themselves.

Here is how the rest of this paper is organized. In Section 1 we briefly review the Fourier-series method for one-dimensional Laplace transforms, and in Section 2 we develop the associated scaling algorithm. In Section 3 we develop an additional scaling algorithm to aid in computing functions at very large or small arguments. In Section 4 we present some simple examples (exponential and power functions) to illustrate the scaling concepts. In Section 5 we discuss the application of the scaling to calculate asymptotic parameters of probability distributions. In Section 6 we apply the new scaling algorithm to compute small tail probabilities in the statistical multiplexing model considered by Choudhury, Lucantoni and Whitt [10], i.e., the BMAP/G/1 queue. In Section 7 we describe the variant of the main scaling algorithm for generating functions. In Section 8 we present the scaling algorithm for multidimensional transforms, which may be Laplace transforms in some dimensions and generating functions in others. We apply the multidimensional scaling in Section 9 to a two-dimensional example involving a closed queueing network. There the scaling is an alternative to our scaling in [5]. Finally, we state our conclusions in Section 10.

1. The Fourier-Series Method

Given a Laplace transform

$$\hat{f}(s) = \int_0^\infty e^{-st} f(t)dt \ ,$$

the fourier-series method calculates the desired function $f(t)$ by constructing a periodic function by aliasing. The periodic function is calculated from its Fourier series, whose coefficients can be expressed in terms of the Laplace transform values. To ensure that the aliasing error is negligible, the original function $f(t)$ is replaced by the damped function $e^{-bt}f(t)$, $t \geq 0$, for $b > 0$. Then the damped function is extended to the whole line by letting it be 0 for $t < 0$.

By this reasoning, we are able to write

$$f(t) = f_a(t) - e_a(t) \ , \tag{1}$$

where the *periodic approximation* is

$$f_a(t) = \frac{e^{A/2l}}{2lt} \sum_{k=-\infty}^{\infty} \hat{f}\left(\frac{A}{2lt} + \frac{ik\pi}{lt}\right) e^{ik\pi/l} \tag{2}$$

and the *aliasing error* is

$$e_a(t) = \sum_{k=1}^{\infty} e^{-Ak} f((1+2kl)t) \ ; \tag{3}$$

see (2.7) and (2.8) of Choudhury, Lucantoni and Whitt [8]. (These are the one-dimensional versions of the two-dimensional formulas given there.)

Given (1)–(3), the idea is to choose the parameter A to make the aliasing error in (3) suitably small, and then choose the parameter l to make the roundoff error in calculating (2) suitably small. The roundoff error arises with limited precision (such as double precision) because the prefactor in (2) can be very large, leading to the multiplication of a very small number by a very large number. The overall procedure is to first choose the parameters l and A to make the roundoff error and the aliasing error small, and then calculate $f_a(t)$ by approximately summing the infinite series in (2), e.g., with an acceleration technique such as Euler summation. Euler summation tends to be effective provided that the function $f(t)$ is suitably smooth; e.g., see p. 46 of [3] and O'Cinneide [15]. If the function is not initially sufficiently smooth, then convolution smoothing can be considered; e.g., see p. 39 of [3] and Platzman, Ammons and Bartholdi [17].

When f is a cumulative distribution function (cdf) or a complementary cdf (ccdf, i.e., one minus the cdf), $|f(t)| \leq C$ for $C = 1$ and all t, so that

$$|e_a(t)| \leq \frac{Ce^{-A}}{1-e^{-A}} \approx Ce^{-A} \ , \tag{4}$$

and the aliasing error is easily controlled. When f is a probability density function (pdf), the aliasing error is also usually easy to control. Then we assume that

$$|f(x)| < C \quad \text{for all} \quad x \geq (1+2l)t \ , \tag{5}$$

for some C, which is also sufficient to have the bound (4), as can be seen from (3). For pdf's, we may have $f(0) = \infty$, but the behavior below $(1+2l)t$ plays no role in the aliasing error. In general, pdf's need not satisfy (5) (because they could have arbitrarily high peaks, approximating point masses away from the origin), but they do in typical cases. Of course, the bound C is usually not known in advance, but reasonable estimates of C can usually be determined when performing the inversion, e.g., by starting with $C = 1$ and making adjustments as necessary from the observed accuracy. The accuracy can be estimated by performing the inversion with two different parameter pairs (A, l).

More general functions are difficult to invert for two reasons. First, (5) and, thus, (4) need not hold. Second, we may wish to calculate very small values $f(t)$. We assume that we are interested in controlling the relative aliasing error $|e_a(t)/f(t)|$ rather than the absolute aliasing error. Our scaling strategy is to transform the function into a pdf whose mean coincides with the desired inversion point t, so that we can invoke (5) and achieve (4). We aim to make the inversion point t coincide with the mean, so that the transformed function value should be $O(1)$. Then a small absolute aliasing error for the scaled function will translate into small relative error for the original function $f(t)$.

2. The Scaling Algorithm

We transform $f(t)$ into a pdf using the scaled function

$$f_{\alpha_0,\alpha_1}(t) = \alpha_0 e^{-\alpha_1 t} f(t), \quad t \geq 0, \tag{6}$$

which has Laplace transform

$$\hat{f}_{\alpha_0,\alpha_1}(s) = \alpha_0 \hat{f}(s + \alpha_1). \tag{7}$$

To compute $f(t)$, we first compute $f_{\alpha_0,\alpha_1}(t)$ by numerically inverting $\hat{f}_{\alpha_0,\alpha_1}(s)$ and then calculate $f(t)$ by letting

$$f(t) = \alpha_0^{-1} e^{\alpha_1 t} f_{\alpha_0,\alpha_1}(t). \tag{8}$$

Our choice of parameters α_0 and α_1 (discussed below) is intended to make $f_{\alpha_0,\alpha_1}(t)$ not too small or large, but $f(t), \alpha_0$ and $e^{\alpha_1 t}$ may be very small or large (even outside the floating point limit of the computer). Hence, we compute (8) using logarithms if necessary; i.e., we compute

$$\log f(t) = \alpha_1 t + \log f_{\alpha_0,\alpha_1}(t) - \log \alpha_0. \tag{9}$$

Now we turn to the choice of the scaling parameters α_0 and α_1 in (6) and (7). We choose the parameters α_0 and α_1 so that the function $f_{\alpha_0,\alpha_1}(t)$ is like a probability density function (pdf) and the desired inversion point t is near the mean. To achieve this property, we assume that the desired function f is nonnegative.

In the following, let $\hat{f}'(\alpha)$ be the derivative of $\hat{f}(s)$ at $s = \alpha$.

Theorem 2.1. *Suppose that the function f is nonnegative and let s^* be the rightmost singularity* *of $\hat{f}(s)$, with $s^* = -\infty$ if $\hat{f}(s)$ is analytic. For any m, $0 < m < \infty$, if the equation*

$$\frac{-\hat{f}'(\alpha_1)}{\hat{f}(\alpha_1)} = m \tag{10}$$

has a real root α_1 in the interval (s^*, ∞), then f_{α_0,α_1} in (6) is a bonafide probability density function with mean m for α_1 satisfying (10) and

$$\alpha_0 = 1/\hat{f}(\alpha_1) \ . \tag{11}$$

Moreover $-\hat{f}'(\alpha)/\hat{f}(\alpha)$ is decreasing in α for real α in (s^*, ∞), so that (10) has at most one real root.

Proof. Note that

$$\frac{-\hat{f}'(\alpha)}{\hat{f}(\alpha)} = \frac{\int_0^\infty te^{-\alpha t} f(t)dt}{\int_0^\infty e^{-\alpha t} f(t)dt} \ ,$$

so that it is indeed the mean of the pdf with density $f_\alpha(t) \equiv C_\alpha e^{-\alpha t} f(t)$, where C_α is chosen so that the total mass is 1. To establish the monotonicity of $-\hat{f}'(\alpha)/\hat{f}(\alpha)$, we use stochastic order concepts; these are discussed in the Appendix of Ross [19] and Shaked and Shanthikumar [20]. Note that the ratio of the pdf's satisfies

$$\frac{f_{\alpha_2}(t)}{f_{\alpha_1}(t)} = \frac{C_{\alpha_2} e^{-(\alpha_2 - \alpha_1)t}}{C_{\alpha_1}}, \quad t \geq 0,$$

which is decreasing in t for $\alpha_2 > \alpha_1$, which implies that f_{α_2} is smaller than f_{α_1} in the likelihood ratio ordering, which in turn implies that f_{α_2} is less than f_{α_1} in stochastic order, which in turn implies that the means are ordered, i.e.,

$$\frac{-\hat{f}'(\alpha_2)}{\hat{f}(\alpha_2)} < \frac{-\hat{f}'(\alpha_1)}{\hat{f}(\alpha_1)}.$$

Finally, given α_1, α_0 must satisfy (11) to make $f_{\alpha_0,\alpha_1}(t)$ a proper pdf. □

Hence, to compute $f(t)$, we at first scale the function f using (10) and (11) with $m = t$ and then use the inversion formula in Section 1. Since the function $-\hat{f}'(\alpha)/\hat{f}(\alpha)$ is monotone, it is relatively easy to find a root α_1 to equation (10) when it exists by a simple search algorithm. For example, we can start with $\alpha = 0$ and then consider $\alpha = +1$ or $\alpha = -1$. Afterwards, increase α in absolute value geometrically (e.g., $1, 2, 4, 8 \ldots$) until a finite interval containing the root is identified. Thereafter use bisection search. If no finite interval is identified after a large number of steps, we conclude that no root to (10) exists. Alternatively, we can use the Newton-Raphson root finding algorithm, which requires the derivative

$$\frac{d}{d\alpha} \frac{\hat{f}'(\alpha)}{\hat{f}(\alpha)} = \frac{\hat{f}''(\alpha)}{\hat{f}(\alpha)} - \frac{\hat{f}'(\alpha)^2}{\hat{f}(\alpha)^2} \ ,$$

which in turn requires the second derivative $\hat{f}''(\alpha)$. Typically, the second method will be much faster than the first, but neither requires significant computation.

However, it is important to be aware of two complications. First, the desired root α_1 in (10) must be to the right of all singularities of the Laplace transform $\hat{f}(s)$. This should be checked. Second, it is important to be aware that a root to equation (10) need not exist even if $\hat{f}(\alpha)$ is finite for one or more values of α. For example, suppose that

$$f(t) = (1 + t^c)^{-1} \tag{12}$$

for $c > 2$, so that

$$\tau = \int_0^\infty t f(t) dt < \infty \ . \tag{13}$$

Then $f_{\alpha_0,\alpha_1}(t)$ defined by (6) has mean $m < \tau$ in (13) for all $\alpha_1 \geq 0$, but has infinite mean for $\alpha_1 < 0$. Thus, it is possible to find a root to (10) for all $m \leq \tau$, but not for $m > \tau$.

3. Scaling For Large or Small Arguments

From (2) it is evident that there can be numerical difficulties if t is very small, because then the prefactor in (2) is very large. There can also be numerical difficulties with (2) if t is large and the transform $\hat{f}(s)$ has singularities on the line with $Re(s) = 0$, because the argument of $\hat{f}$ in (2) will be close to this line when t is large.

When t is very small or large, inversion can often be replaced by asymptotic analysis. We now show that it is also possible to avoid this difficulty by scaling the function so that the inversion is performed at $t = 1$. For this purpose, we use the scaled function

$$f^\tau(t) \equiv f(\tau t), \quad t \geq 0, \tag{14}$$

which has Laplace transform

$$\hat{f}^\tau(s) = \frac{1}{\tau} \hat{f}(s/\tau). \tag{15}$$

We compute $f(t)$ by calculating $f^t(1)$ by numerically inverting $\hat{f}^t(s)$. For this procedure, we exploit the fact that

$$f^t(1) = f^1(t) = f(t) \ .$$

Since the inversion point is shifted to $t = 1$ after scaling, from (2) we see that there should be no numerical difficulty even if the actual inversion point t before scaling is arbitrarily small or large. However, there can be numerical difficulty in computing $\hat{f}^t(s)$ using (15) with $\tau = t$ for very small t. This difficulty may be removed by the following key observation based on the initial value theorem for Laplace transforms. If $f(t) \to f(0)$ as $t \to 0$, then

$$\begin{aligned}\lim_{t\to 0} \hat{f}^t(s) &= \lim_{t\to 0} \frac{1}{t}\hat{f}\left(\frac{s}{t}\right) = \frac{1}{s}\lim_{\bar{s}\to\infty} \bar{s}\hat{f}(\bar{s}) \quad \text{for} \quad \bar{s} = s/t \\ &= \frac{1}{s}\lim_{t\to 0} f(t) = \frac{f(0)}{s} \ .\end{aligned}$$

The above states that if $\lim_{t\to 0} f(t)$ is finite, then so is $\lim_{t\to 0} \hat{f}^t(s)$, so that it should be possible to rewrite the righthand side of (15) with $\tau = t$ (basically by cancelling out t^{-1} terms) such that there is no computational difficulty for arbitrarily small t.

On the other hand, if $f(t)$ has a singularity at $t = 0$, there would be a corresponding singularity of $\hat{f}^t(s)$ at $t = 0$ and any inversion procedure would not work for obvious reasons. We illustrate this using a simple example. Let $f(t)$ represent the cdf of waiting time in an M/G/1 queue with utilization ρ, arrival rate λ and service-time LST $\hat{h}(s)$. Then the LST of $f(t)$ is

$$\hat{f}(s) \equiv \int_0^\infty e^{-st} df(t) = \frac{(1-\rho)}{s - \lambda + \lambda\hat{h}(s)} \ . \tag{16}$$

From (15),

$$\hat{f}^t(s) = \frac{1}{t} \cdot \frac{(1-\rho)}{(s/t) - \lambda + \lambda\hat{h}(s/t)} \ . \tag{17}$$

If we try to compute directly from (17), then there is numerical difficulty for small t. However, (17) can be rewritten as

$$\hat{f}^t(s) = \frac{(1-\rho)}{s - \lambda t + \lambda t\hat{h}(s/t)} \ , \tag{18}$$

and there is no numerical difficulty in computing from (18) for arbitrarily small t.

Next, if we do the same exercise on the pdf instead of the cdf, then we get

$$\hat{f}(s) = \frac{s(1-\rho)}{s - \lambda + \lambda\hat{h}(s)} \tag{19}$$

and

$$\hat{f}^t(s) = \frac{1}{t} \ \frac{(s/t)(1-\rho)}{(s/t) - \lambda + \lambda\hat{h}(s/t)} = \frac{(s/t)(1-\rho)}{s - \lambda t + \lambda t\hat{h}(s/t)} \ . \tag{20}$$

Note that (20) does have a numerical difficulty for small t, whereas (18) does not. This is because the pdf has a singularity at $t = 0$, while the cdf does not.

4. Simple Examples

In this section we discuss two simple examples to illustrate the scaling concepts.

Example 4.1. (an exponential function) Suppose that $f(t) = e^{\theta t}, t \geq 0$, with Laplace transform $\hat{f}(s) = (s-\theta)^{-1}$. Of course, no numerical inversion is needed in this case; this example is to illustrate the procedure. For $\theta > 0$, there does not exist a finite C such that $|f(x)| \leq C$ for $x \geq (1+2l)t$. For $\theta < 0$, $f(t)$ will be very small when t is suitably large.

In this case equation (10) becomes

$$\frac{-\hat{f}'(\alpha_1)}{\hat{f}(\alpha_1)} = \frac{1}{\alpha_1 - \theta} = t,$$

so that

$$\alpha_1 = \theta + t^{-1}$$

and, by (11),

$$\alpha_0 = \frac{1}{\hat{f}(\alpha_1)} = (\alpha_1 - \theta) = t^{-1}.$$

Hence,

$$\hat{f}_{\alpha_0,\alpha_1}(s) = \frac{t^{-1}}{s + t^{-1}},$$

and

$$f_{\alpha_0,\alpha_1}(x) = t^{-1} e^{-x/t}, \quad x \geq 0,$$

which is the exponential pdf with mean t, as could be predicted from Theorem 2.1.

Note that the scaled transform $\hat{f}_{\alpha_0,\alpha_1}(s)$ does not have the numerical difficulties of the original transform $\hat{f}(s)$. First, for all $x \geq (1+2l)t$, $|f_{\alpha_0,\alpha_1}(x)| \leq C$, where

$$C = \frac{1}{t} e^{-(1+2l)} \text{ and } \frac{C}{|f_{\alpha_0,\alpha_1}(t)|} = e^{-2l} < 1.$$

Second, $f_{\alpha_0,\alpha_1}(t) = 1/et$, which does not become too large or small unless t itself is very small or large.

We can address the problem of extremely large or small t by first using the scaling procedure in Section 3. As in (15), we let

$$\hat{f}^t(s) = \frac{1}{t}\hat{f}(s/t) = \frac{1}{s - \theta t}\ .$$

To get $f(t)$, we compute $f^t(1)$ by inverting $\hat{f}^t(s)$. Hence, the inversion point is shifted from t to 1. Now we apply the scaling procedure in Section 2 to get

$$1 = \frac{-\hat{f}^{t\prime}(\alpha_1)}{\hat{f}^t(\alpha_1)} = \frac{1}{\alpha_1 - \theta t}\ ,$$

so that $\alpha_1 = \theta t + 1$ and $\alpha_0 = 1$. As a consequence,

$$\hat{f}^t_{\alpha_0,\alpha_1}(s) = (1+s)^{-1}$$

and

$$f^t_{\alpha_0,\alpha_1}(x) = e^{-x}, \quad x \geq 0 \ .$$

Hence, we calculate $f^t_{\alpha_0,\alpha_1}(1) = e^{-1}$ by numerically inverting $(1+s)^{-1}$, which avoids all problems of small or large t. Then we calculate $f^t(1)$ by applying (8), i.e.,

$$f(t) = f^t(1) = \alpha_0^{-1} e^{\alpha_1} f_{\alpha_0,\alpha_1}(1) = e^{\theta t} \ .$$

Example 4.2. (a power) Now suppose that

$$f(t) = t^x, \quad t \geq 0,$$

for some $x > 0$, which has Laplace transform

$$\hat{f}(s) = \frac{\Gamma(x+1)}{s^{x+1}} \ .$$

There is a genuine difficulty in the inversion for t near 0 if x is negative, because then 0 is a singularity. Otherwise, we can apply the method of Section 3 to transform the inversion point to $t = 1$. So henceforth assume that the inversion point is $t = 1$.

We solve (10) to obtain α_1, obtaining

$$1 = \frac{-\hat{f}'(\alpha_1)}{\hat{f}(\alpha_1)} = \frac{x+1}{\alpha_1} \ ,$$

so that $\alpha_1 = x + 1$. Then, by (11),

$$\alpha_0 = \frac{1}{\hat{f}(\alpha_1)} = \frac{(x+1)^{x+1}}{\Gamma(x+1)} \ .$$

Hence,

$$\hat{f}_{\alpha_0,\alpha_1}(s) = \alpha_0 \hat{f}(s + \alpha_1) = \left(\frac{x+1}{s+x+1}\right)^{x+1} \tag{21}$$

and

$$f_{\alpha_0,\alpha_1}(t) = \frac{(x+1)^{x+1} t^x e^{-(x+1)t}}{\Gamma(x+1)}, \quad t \geq 0, \tag{22}$$

which we recognize as a gamma pdf with shape and scale parameter $x + 1$, and so mean 1 and variance $(1 + x)^{-1}$.

The scaled function-transform pair $(f_{\alpha_0,\alpha_1}(t), \hat{f}_{\alpha_0,\alpha_1}(s))$ in (21) and (22) is much better behaved than the original pair $(f(t), \hat{f}(s))$. First, the function $f_{\alpha_0,\alpha_1}(t)$ is strictly decreasing for $x > 1$, so that $C/|f_{\alpha_0,\alpha_1}(1)| < 1$. Second, the quantity $f_{\alpha_0,\alpha_1}(1)$ does not get too small or large for any x, even large x, as can be seen from Stirling's formula.

5. Asymptotic Parameters of Tail Probabilities

Suppose that we know or suspect that a complementary cdf (ccdf) $F^c(t) \equiv 1 - F(t)$ has the asymptotic form

$$F^c(t) \sim \alpha t^{\beta} e^{-\eta t} \text{ as } t \to \infty \tag{23}$$

for positive constants α and η and arbitrary constant β, where $f(t) \sim g(t)$ as $t \to \infty$ means that $f(t)/g(t) \to 1$ as $t \to \infty$. Choudhury and Lucantoni [7] and Abate et al. [1] showed how the asymptotic parameters α, η and β can be calculated numerically from the moments after they have been calculated by numerically inverting the moment generating function.

Now we show how the asymptotic parameters can be calculated from three values of $F^c(t)$ for large t. It suffices to solve the three equations

$$\log F^c(t_i) = \log \alpha + \beta \log t_i - \eta t_i \tag{24}$$

with $i = 1, 2, 3$ for $\log \alpha$, β and η. We use the scaling to compute $F^c(t_i)$ by numerical inversion from the Laplace transform $(1 - \hat{f}(s))/s$ for suitably large and separated t_i.

To illustrate, we consider the first-moment cdf (the time-dependent mean normalized to be a cdf) of reflected Brownian motion (RBM), which was analyzed in Abate and Whitt [2]. The associated RBM first-moment ccdf, denoted by $H^c(t)$, is known to have the asymptotic form

$$H^c(t) \sim 2\sqrt{\frac{2}{\pi t^3}} e^{-t/2} \quad \text{as} \quad t \to \infty \; ; \tag{25}$$

see Corollary 1.3.5 on p. 567 of [2]. We will verify that $\eta = 1/2$, $\alpha = 2\sqrt{2/\pi}$ and $\beta = -3/2$ by applying numerical transform inversion. The Laplace transform of $H^c(t)$ is

$$\hat{H}^c(s) = \frac{1 - \hat{h}_1(s)}{s} = \frac{s + 1 - \sqrt{1 + 2s}}{s^2} \; ; \tag{26}$$

see p. 568 of [2].

From (26), we obtain the derivative

$$\hat{H}^{c'}(s) = \frac{(2 + 3s)\sqrt{1 + 2s} - (2 + 5s + 2s^2)}{s^3(1 + 2s)} \; , \tag{27}$$

so that

$$\begin{aligned} r(s) &= \frac{-\hat{H}^{c'}(s)}{\hat{H}^c(s)} = \frac{1 + 2s - \sqrt{1 + 2s}}{2s(1 + 2s)} \\ &= (1 + (1 + 2s)^{-1/2})/2s \; . \end{aligned} \tag{28}$$

A partial check is obtained by noting that, by L'Hospital's rule, $r(0) = 0.5$, which agrees with the known formula for the mean; combine Corollaries 1.3.4 and 1.5.1 of [2].

We computed $H^c(t)$ for several values of t using the scaling and the algorithm in Section 1. We verified that all computations are correct up to the displayed number of places by doing independent computations with inversion parameters $l = 2$ and $l = 3$. The results are displayed below in Table 1. We get the first four values equally accurately even without scaling, but for $t \geq 50$ there are significant errors in the unscaled algorithm. (For the cases with $t \geq 100$, the unscaled algorithm using double precision cannot distinguish the exact values from 0.) Note that at $t = 2000$, $H^c(t)$ is even below the floating-point limit of the computer we used. For this example the scaling parameter α_1 approaches -0.5 and α_0 approaches 0.5 as t approaches infinity. (This is easy to show analytically as well.)

t	$H^c(t)$	asymptotic approximation
2.0	0.5679012E-01	0.2075537E+00
5.0	0.5634086E-02	0.1171599E-01
10.0	0.2186916E-03	0.3400147E-03
20.0	0.6303259E-06	0.8099911E-06
50.0	0.5611686E-13	0.6268347E-13
100.0	0.2905855E-24	0.3077839E-24
200.0	0.2038120E-46	0.2098828E-46
500.0	0.3764690E-112	0.3809733E-112
1000.0	0.3573839E-221	0.3595250E-221
2000.0	0.9029074E-439	0.9056141E-439

Table 1. Numerical results for the RBM first-moment ccdf $H^c(t)$

Table 1 also shows the asymptotic approximation using (25). Note that the exact value approaches the asymptote, but remarkably slowly. For example, there is only two-digit accuracy when the tail probability is 10^{-439}, which is already far outside the typical range of interest. This is unlike distributions with true exponential tails where the convergence is often spectacularly fast. We can compute estimates of the asymptotic parameters based on any three values of t, but we cannot expect them to be very accurate because of the accuracy in Table 1. If we use the last three values, then we get

$$\alpha = 1.4607588 \ , \quad \beta = -1.4872161 \ , \quad \eta = 0.5000059 \ ,$$

whereas the true values are

$$\alpha = 1.5957691 \ , \quad \beta = -1.5 \ , \quad \eta = 0.5 \ .$$

The accuracy for the asymptotic decay rate η is excellent, but the accuracy for α and β is not too good. However, overall the accuracy is good enough for many practical purposes. Just as in the context of moments [1], the accuracy may be greatly enhanced. Here we can assume a multi-term asymptotic expansion and get estimates based on many points. The main point here, however, is that the scaling enables us to accurately compute very small tail probabilities.

6. A Multiplexing Example

We now consider the MMPP/D/1 queueing model used to study the effectiveness of effective bandwidths to describe buffer overflow probabilities with statistical multiplexing in [10]. In that model there are N independent sources sending fixed-length cells to a buffer, which is drained by an output channel at a fixed rate whenever cells are present. The cell service-time distribution is thus deterministic and its value is set at 1 (by choosing the unit of time).

As in [10], we consider the special case of homogeneous on-off sources. For each source, the on and off periods have exponential distributions. The mean off-period ζ is 10 times the mean on-period ω. During the on period cells arrive according to a Poisson process at (peak) rate p. The mean number of arrivals in an on-period is $p\omega = 60$. The source rates are appropriately adjusted so that the long-term utilization of the output channel is 0.3 for each N.

We consider the cases of $N = 2$ and $N = 24$. The case $N = 24$ is the example in Section II of [10]. In that case $\omega = 436.6$, $\zeta = 4363.3$ and $p = 0.1375$. The case $N = 2$ is an alternative considered in Section IV of [10].

The Laplace-Stieltjes transform $\hat{W}(s)$ of the steady-state waiting-time distribution (used to approximate the buffer overflow probability) is given in (8.4) of [10]. The waiting-time tail probability $P(W > x)$ is calculated by numerically inverting the transform $\hat{W}^c(s) = [1 - \hat{W}(s)]/s$. However, there are two difficulties. First, the D service is not so easy to treat. Hence, we approximate the D service-time transform e^{-s} by an accurate non-probability transform that is a hybrid of Padé and Erlang approximations. In particular, we used the approximation

$$e^{-s} = \frac{a_0 + a_1 s}{(1 + \frac{bs}{128})^{128}} + 1 - a_0 \ , \tag{29}$$

where $a_0 = 0.9984796$, $a_1 = 0.0894316$ and $b = 0.9119549$. Note that the approximation is close to the Erlang E_{128} approximation, which is obtained by setting $a_0 = 1$, $a_1 = 0$ and $b = 1$. However, (29) also matches the first 3 moments of the deterministic distribution and can be shown to be very accurate in predicting the waiting-time tail probability. We intend to discuss the approximation of transforms such as e^{-s} in more detail elsewhere [11]; our approach is similar in spirit to Akar and Arikan [4].

The second difficulty is that the transform $\hat{W}(s)$ has an involved matrix expression, so that it is not easy to analytically calculate the derivative of $\hat{W}^c(s)$, as needed for the scaling algorithm in Section 3. Therefore, we use a numerical differentiation procedure. In particular, we use the formula

$$hf'(x) = \Delta_x^{(1)} - \frac{1}{24}\Delta_x^{(3)} + \frac{3}{128}\Delta_x^{(5)} - \frac{5}{7168}\Delta_x^{(7)} + \ldots \tag{30}$$

where $\Delta_x^{(1)} = f(x + (h/2)) - f(x - (h/2))$ and $\Delta_x^{(n+1)} = \Delta_{x+h/2}^{(n)} - \Delta_{x-h/2}^{(n)}$ for $n \geq 1$, which is based on Bessel's interpolation formula; see equation III-C-11 on p. 100 of Kopal [14]. Formula (30) is based on computing function values at $x \pm (2j+1)h/2$ for $j = 0, 1, \ldots, n$. We found that $h = \max\{0.01|x|, 0.001)$ and $n = 3$ (8 points) is often satisfactory. We also ensure (by reducing h if necessary) that each point in the derivative calculation is to the right of the rightmost singularity of the transform (which is easy to calculate accurately since it is the negative of the asymptotic exponential decay rate of the tail probability [7], [10]).

Numerical values of the tail probabilities and the scaling parameters α_1 and α_0 as a function of the buffer size are given for the two cases $N = 2$ and $N = 24$ in Table 2. With standard double precision and without scaling, the inversion algorithm would typically have errors of order 10^{-9} and hence all probabilities below 10^{-9} would have large relative errors. However, with the scaling, the algorithm maintains accuracy to 10^{-20}. (The accuracy was confirmed by independent computations with roundoff control parameters $\ell = 1$ and $\ell = 2$.)

	$N = 2$			$N = 24$		
buffer size	tail prob.	α_1	α_0	tail prob.	α_1	α_0
200	.13564e-1	-.01363	.00777	.63603e-5	-0.1777	1.152
400	.36103e-3	-.01577	.00428	.63129e-7	-.01788	1.116
600	.96600e-5	-.01652	.00298	.97442e-9	-.01793	1.091
800	.25848e-6	-.01690	.00229	.18351e-10	-.01795	1.071
1000	.69165e-8	-.01714	.00186	.38679e-12	-.01797	1.054
1200	.18507e-9	-.01729	.00156	.87597e-14	-.01798	1.039
1400	.49522e-11	-.01741	.00135	.20840e-15	-.01799	1.026
1600	.13251e-12	-.01749	.00119	.51345e-17	-.01800	1.014
1800	.35457e-14	-.01756	.00106	.12971e-18	-.01801	1.003
2000	.94876e-16	-.01761	.00096	.33352e-20	-.01801	0.992

Table 2. Tail probabilities and scaling parameters α_1 and α_0 for the MMPP/D/1 model in Section 6

as a function of the number N of sources and the buffer size. In these cases the rightmost singularity of the transform $\hat{W}(s)$ is at $-\eta$ for $\eta = 0.0181047$.

7. Scaling for Generating Functions

For generating functions, there is no analog of the small argument problem for Laplace transforms in Section 3, but there is an analog to the small-or-large function values problem in Section 2, and a minor modification of the same procedure applies.

Given the generating function $q^*(z) = \sum_{k=0}^{\infty} q_k z^k$, we construct the scaled sequence $\{q_{\alpha_0,\alpha_1}(k) : k \geq 0\}$ by setting

$$q_{\alpha_0,\alpha_1}(k) = \alpha_0 \alpha_1^k q_k, \quad k \geq 0, \tag{31}$$

which has generating function

$$q^*_{\alpha_0,\alpha_1}(z) = \alpha_0 q^*(\alpha_1 z) \ . \tag{32}$$

The following theorem is the discrete analog of Theorem 2.1. For its statement, let $q^{*'}(z)$ be the derivative of $q^*(z)$.

Theorem 7.1. *Suppose that the sequence q is nonnegative and let z^* be the radius of convergence of the generating function $q^*(z)$. For any λ, $0 < \lambda < \infty$, if the equation*

$$\frac{\alpha_1 q^{*'}(\alpha_1)}{q^*(\alpha_1)} = \lambda \tag{33}$$

has a real root α_1 in the interval $(0, z^*)$, then q_{α_0,α_1} in (31) is a bonafide probability mass function with mean λ for α_1 satisfying (33) and

$$\alpha_0 = 1/q^*(\alpha_1) \ . \tag{34}$$

Moreover, $\alpha q^{*'}(\alpha)/q^*(\alpha)$ is strictly increasing in α for positive α in $(0, z^*)$, so that (33) has at most one real root.

Proof. Note that

$$\frac{\alpha q^{*'}(\alpha)}{q^*(\alpha)} = \frac{\sum_{k=0}^{\infty} k \alpha^k q_k}{\sum_{k=0}^{\infty} \alpha^k q_k} \ ,$$

so that it is the mean of the pmf $q_\alpha(k) = C_\alpha \alpha^k q_k$, where the constant C_α is chosen to make the total mass 1. Note that the ratio of pmf's satisfies

$$\frac{q_{\alpha_2}(k)}{q_{\alpha_1}(k)} = \frac{C_{\alpha_2} \alpha_2^k}{C_{\alpha_1} \alpha_1^k}$$

which is increasing in k for $\alpha_2 > \alpha_1$, so that q_{α_2} is larger than q_{α_1} in the discrete likelihood ratio ordering, which implies stochastic order and the ordering of the means. $\square$

Hence, if we want to calculate q_k, then we would scale by (31) with α_1 chosen to satisfy (33) for $\lambda = k$.

8. Scaling for Multidimensional Transforms

The scaling for Laplace transforms in Section 2 and generating functions in Section 7 extends to multidimensional transforms, which may have some dimensions discrete (generating functions) and other dimensions continuous (Laplace transforms). We illustrate in this section by discussing the bivariate mixed case. Here we call the desired bivariate function that is a pdf in one dimension and a pmf in the other dimension simply a pdf.

Given a bivariate function $f(t,k)$ of a continuous variable t and a discrete variable k, let its transform be

$$\hat{f}(s,z) = \int_0^\infty \sum_{k=0}^\infty f(t,k) z^k e^{-st} dt \ . \tag{35}$$

We introduce the scaled function

$$f_\alpha(t,k) = \alpha_0 e^{-\alpha_1 t} \alpha_2^k f(t,k) \tag{36}$$

for $\alpha \equiv (\alpha_0, \alpha_1, \alpha_2)$, which has transform

$$\hat{f}_\alpha(s,z) = \alpha_0 \hat{f}(s+\alpha_1, \alpha_2 z) \ . \tag{37}$$

Theorem 8.1. *Suppose that the bivariate function f is nonnegative. For any m_1, $0 < m_1 < \infty$, and integer m_2, $0 < m_2 < \infty$, if the pair of equations*

$$-\frac{\partial}{\partial s} \log \hat{f}(s,z) \,|_{s=\alpha_1, z=\alpha_2} = m_1 \tag{38}$$

and

$$z \frac{\partial}{\partial z} \log \hat{f}(s,z) \,|_{s=\alpha_1, z=\alpha_2} = m_2 \tag{39}$$

has a solution (α_1, α_2) such that $\hat{f}_\alpha(s,z)$ is analytic for $|z| < 1$ and $Re(s) > 0$, then f_α in (36) is a bivariate pdf with means m_1 and m_2 in the two dimensions, provided that

$$\alpha_0 = \frac{1}{\hat{f}(\alpha_1, \alpha_2)} \ . \tag{40}$$

Proof. The arguments of Theorems 2.1 and 7.1 can be repeated. Recall that

$$\frac{-\partial}{\partial s} \log \hat{f}(s,z) = \frac{-\frac{\partial}{\partial s} \hat{f}(s,z)}{\hat{f}(s,z)}$$

and

$$z\frac{\partial}{\partial z}\log \hat{f}(s,z) = z\frac{\frac{\partial}{\partial z}\hat{f}(s,z)}{\hat{f}(s,z)} \ . \quad \square$$

Unfortunately, however, it is not as easy to find the solution of the pair of equations (38) and (39) as it is to find the solution of the single equation arising in the one-dimensional cases in Sections 3 and 5. For practical purposes, we suggest using an iterative procedure. First fix α_2 and then find candidate values of α_0 and α_1, using the appropriate one-dimensional algorithm. Then fix α_1 and find new values of α_0 and α_2, again using the appropriate one-dimensional algorithm. Then repeat, fixing α_2, and so forth, stopping after a few iterations, since an exact solution is not required. To speed up convergence, after a few initial search steps, the Newton-Raphson root finding algorithm can be used.

The procedure just described seems often to be effective. In part, this is due to the function $f(\alpha_1, \alpha_2)$ here being monotone in each argument separately. However, to show that the multidimensional case is indeed more complicated than the one-dimensional case, we now give an example showing that a solution (to the mean equations) need not be unique in two dimensions.

Example 8.1. We consider a two-dimensional Laplace transform, i.e., the continuous-continuous case. To demonstrate lack of uniqueness, consider the four-point probability distribution assigning mass 1/4 to each of the points $(3,0), (2,1), (1,2)$ and $(0,3)$ in $\mathbb{R}^2$. Let the two means be $m_1 = 1$ and $m_2 = 2$. Then the two mean equations become

$$\frac{3x^3 + 2x^2y + xy^2}{x^3 + x^2y + xy^2 + y^3} = 1$$

and

$$\frac{3y^3 + 2y^2x + yx^2}{x^3 + x^2y + xy^2 + y^3} = 2 \ ,$$

where $x = e^{-\alpha_1}$ and $y = e^{-\alpha_2}$. These equations both reduce to the single equation

$$2x^3 + x^2y - y^3 = 0 \ . \tag{41}$$

Dividing through (41) by y^3, we see that $z \equiv x/y$ satisfies the equation

$$2z^3 + z^2 - 1 = 0 \ ,$$

which has one positive real root $z = 0.65730$. Hence (zy, y) is a solution to (41) for all $y > 0$. $\square$

It still remains to better understand the behavior of the system of equations (38) and (39) in the multidimensional case. We conclude this section by giving a condition under which the bivariate function is monotone in the arguments α_1 and α_2, but even this leaves open the questions of existence, uniqueness and convergence. For our monotonicity result, we exploit the notions of total positivity and multivariate likelihood ratio ordering, see Karlin and Rinott [13] and Whitt [21]. One bivariate pdf $f_1(t_1, t_2)$ is said to be less than or equal to another $f_2(t_1, t_2)$ in the *multivariate likelihood ratio* (MLR) order if

$$f_1(x) f_2(y) \le f_1(x \wedge y) f_2(x \vee y)$$

for all vectors $x \equiv (x_1, x_2)$ and $y \equiv (y_1, y_2)$, where $x \wedge y = (x_1 \wedge y_1, x_2 \wedge y_2)$, $x_1 \wedge y_1 = \min\{x_1, y_1\}$ $x \vee y = (x_1 \vee y_1, x_2 \vee y_2)$ and $x_1 \vee y_1 = \max\{x_1, y_1\}$, and we write $f_1 \le_{lr} f_2$. A single bivariate pdf f is said to be *totally positive of order* 2 (TP_2) if $f \le_{lr} f$.

Theorem 8.2. *Consider a nonnegative bivariate function $f(t_1, t_2)$ of two continuous variables t_1 and t_2. If, in addition, f is TP_2, then the bivariate function*

$$\left(\frac{-\partial}{\partial s_2} \log \hat{f}(s_1, s_2) \ , \quad \frac{-\partial}{\partial s_2} \log \hat{f}(s_1, s_2) \right) |_{s_1=\alpha_1, s_2=\alpha_2}$$

is increasing in α_1 and α_2.

Proof. Given the extra conditions, the proof is the two-dimensional generalization of the proof of Theorem 2.1. Given that f is TP_2, MLR order is equivalent to the ratio $f_{\alpha_2}(x)/f_{\alpha_1}(x)$ being decreasing in the vector x; see Theorem 3 of [21]. The MLR order implies stochastic order, which in turn implies an ordering of the means. $\square$

9. A Two-Dimensional Queueing Network Example

Consider the two-dimensional transform

$$\tilde{g}(z_1, z_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} g(n_1, n_2) z_1^{n_1} z_2^{n_2} = \frac{\exp\left(\sum_{j=1}^{2} \rho_{j0} z_j\right)}{\Pi_{i=1}^{q} \left(1 - \sum_{j=1}^{2} \rho_{ji} z_j\right)} \ . \tag{42}$$

This is the generating function of the normalization constant in a closed queueing network with two chains; see [5] for details. Obtaining the normalization constant can be computationally very intensive and many algorithms have been proposed; e.g., the convolution algorithm. However, in [5] it was shown that under many conditions numerical transform inversion is the most efficient procedure. But a difficulty is scaling. We show below how our scaling procedure in this paper works in that context.

We work with the scaled generating function

$$\tilde{g}_\alpha(z_1, z_2) = \alpha_0 \tilde{g}(\alpha_1 z_1, \alpha_2 z_2) \ . \tag{43}$$

The scaling parameters α_1 and α_2 are obtained from the two equations

$$\begin{aligned} n_i &= z_i \frac{\partial}{\partial z_i} \log \tilde{g}(z_1, z_2)|_{z_1=\alpha_1, z_2=\alpha_2} \\ &= \alpha_i \left[\rho_{i0} + \sum_{j=1}^{q} \frac{\rho_{ij}}{1 - \sum_{k=1}^{2} \rho_{kj} \alpha_k} \right] \quad \text{for} \quad i = 1, 2 \ . \end{aligned} \tag{44}$$

We must solve the pair of nonlinear equations in (44). As suggested earlier, we can fix α_2 and search for the value α_1 that satisfies the equation for $i = 1$. Next, fixing α_1 at the value obtained, we search for the value α_2 that satisfies the equation for $i = 2$. We do this repeatedly until convergence is achieved based on some prescribed error criterion. We observed that this procedure indeed converges, but the rate of convergence becomes slow as n_1 and n_2 increases. By contrast, the two-dimensional Newton-Raphson method (see Press et al. [18], Chapter 9) converges very fast (less than 10 steps), provided that we start not too far from the root. So we initially use the search procedure a few times and then the Newton-Raphson method.

Here is an example with generating function (42). It corresponds to a closed queueing network with two single-server queues, one infinite-server queue and two chains. The parameters are:

$$\begin{aligned} \rho_{1,0} &= 1 \ , \quad \rho_{2,0} = 1 \ , \quad \rho_{1,1} = 1 \ , \quad \rho_{2,1} = 2 \ , \\ \rho_{1,2} &= 2 \ , \quad \rho_{2,2} = 3 \ . \end{aligned}$$

The results for several values of the chain populations n_1 and n_2 are displayed in Table 3.

n_1	n_2	g_{n_1,n_2}	α_1	α_2	α_0
3	2	0.243883E+04	0.240070	0.104428	0.806627E-01
30	20	0.627741E+33	0.294397	0.130311	0.589928E-02
300	200	0.973460E+331	0.299451	0.133032	0.564701E-03
3000	2000	0.235196E+3318	0.299945	0.133303	0.562179E-04

Table 3. Numerical results for the normalization constant g_{n_1,n_2} in a closed queueing network with two chains.

The accurate computation in the last case would be challenging by any alternative algorithm. Our algorithm uses Euler summation in each dimension and took only seconds. Accuracy was checked by performing two independent computations with two sets of inversion parameters.

10. Conclusions

We have shown how to scale one-dimensional Laplace transforms (Sectons 2 and 3), one-dimensional generating functions (Section 7) and multidimensional transforms (Section 8) of non-probability functions in order to control the aliasing and round-off errors in applications of the Fourier-series method of numerical transform inversion. The scaling also applies to compute very small values of probability functions. The strategy is to transform the original function into a pdf or pmf and transform the inversion point to the mean. The required equation in one dimension is usually easy to solve, but as noted at the end of Section 3 a solution does not always exist. Moreover, for pathological examples (e.g., a bimodal function with the mean located in a deep troff) the mean may not be a good inversion point. However, examples in Sections 4–6 show that the procedure in one dimension is typically very effective.

As shown in Section 8, the scaling extends to multidimensional transforms, but the resulting scaling equations are more complicated. The scaling equations seem easy to solve in examples, as illustrated by the queueing network example in Section 9, but the multidimensional scaling equations still need to be better understood.

References

[1] J. Abate, G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Asymptotic analysis of tail probabilities based on the computation of moments," *Ann. Appl. Prob.* 5, 983–1007 (1995).

[2] J. Abate and W. Whitt, "Transient behavior of regulated Brownian motion, I: starting at the origin," *Adv. Appl. Prob.* 19, 560–598 (1987).

[3] J. Abate and W. Whitt, "The Fourier-series method for inverting transforms of probability distributions," *Queueing Systems* 10, 5–88 (1992).

[4] N. Akar and E. Arikan, "A numerically efficient method for the MAP/D/1/K queue via rational approximations, *Queueing Systems* 22, 97–120 (1996).

[5] G. L. Choudhury, K. K. Leung and W. Whitt, "Calculating normalization constants of closed queueing networks by numerically inverting their generating functions," *J. ACM* 42, 935–970. (1995).

[6] G. L. Choudhury, K. K. Leung and W. Whitt, "An inversion algorithm to calculate blocking probabilities in loss networks with state-dependent rates," *IEEE/ACM Trans. Networking* 3, 585–601 (1995).

[7] G. L. Choudhury and D. M. Lucantoni, "Numerical computation of the moments of a probability distribution from its transform, *Oper. Res.* 44, 368–381 (1996).

[8] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Multidimensional transform inversion with applications to the transient M/G/1 queue," *Ann. Appl. Prob.* 4, 719–740 (1994).

[9] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Numerical transform inversion to analyze teletraffic models," in *The fundamental Role of Teletraffic in the Evolution of Telecommunication Networks, Proceedings of the 14^{th} International Teletraffic Congress*, J. Labetoulle and J. W. Roberts (eds.), Elsevier, Amsterdam, 1b, 1043–1052 (1994).

[10] G. L. Choudhury, D. M. Lucantoni and W. Whitt, "Squeezing the most out of ATM," *IEEE Trans. Commun.* 44, 203–217 (1996).

[11] G. L. Choudhury and W. Whitt, "Non-probability approximations to a deterministic service-time distribution with applications to traffic modeling in high-speed networks," AT&T Laboratories, 1996.

[12] G. Honig and U. Hirdes, "A method for numerical inversion of Laplace transforms," *J. Comp. Appl. Math.* 10, 113–132 (1984).

[13] S. Karlin and Y. Rinott, "Classes of orderings of measures and related correlation inequalities: I. multivariate totally positive distributions," *J. Multivariate Anal.* 10, 467–498 (1980).

[14] Z. Kopal, *Numerical Analysis*, second ed., Wiley, New York, 1961.

[15] C. A. O'Cinneide, "Euler summation for Fourier series and Laplace transform inversion," *Stochastic Models*, to appear.

[16] R. Piessens and R. Huysmans, "Algorithm 619, automatic numerical inversion of the Laplace transform," *ACM Trans. Math. Software* 10, 348–353 (1984).

[17] L. K. Platzman, J. C. Ammons and J. J. Bartholdi, III, "A simple and efficient algorithm to compute tail probabilities from transforms," *Oper. Res.* 36, 137–144 (1988).

[18] W. H. Press, B. P. Flannery, S. A. Teukolsky and W. T. Vetterling, *Numerical Recipes, FORTRAN version*, Cambridge University Press, Cambridge, England, 1988.

[19] S. M. Ross, *Introduction to Stochastic Dynamic Programming*, Academic Press, New York, 1983.

[20] M. Shaked and J. G. Shanthikumar (1994) *Stochastic Orders and Their Applications*, Academic Press, New York.

[21] W. Whitt, "Multivariate monotone likelihood ratio and uniform conditional stochastic order," *J. Appl. Prob.* 19, 695–701 (1982).