Probabilistic Scaling for the Numerical Inversion of Nonprobability Transforms

Gagan L. Choudhury / AT&T Laboratories, Room 1L-338, Holmdel, NJ 07733-5010. Email: gagan@bellcore.com

Ward Whitt / AT&T Laboratories, Room 3C-178, Murray Hill, NJ 07974-6036. Email: ward@research.att.com

(Received: April 1996; revised: October 1996; accepted: October 1996)

It is known that probability density functions and probability mass functions usually can be calculated quite easily by numerically inverting only transforms (Laplace transforms and generating functions, respectively) with the Fourier-series method. Other more general functions can be substantially more difficult to invert, because the aliasing and rounding errors tend to be more difficult to control. In this article we propose a simple new scaling procedure for nonprobability 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.

Numerical transform inversion is proving to be an effective tool for calculating quantities of interest in operations research models. It is especially useful for queuing models, because many probability distributions are readily available in the form of transforms; e.g., see Choudhury, Lucantoni, and Whitt.1 Abate and Whitt2 have shown that probability distributions can be computed remarkably easily from three transforms by numerical inversion of the Fourier-series method. This is especially true for cumulative distribution functions (cdfs) and probability mass functions (pmfs) because they are nonnegative and bounded above by one, but it also tends to be true for probability density functions (pdfs) 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 Lucantoni5 develop an algorithm for calculating moments, of high as well as low order, from a moment generating function, and Choudhury, Leung, and Whitt 4 calculate performance measures in product-form models by numerically inverting the generating function of the random variable. Both the moment normalization constants tend to grow (or decline) geometrically fast. Calculating these nonprobability functions by numerical inversion has proved to be substantially more difficult than calculating probability distributions. In these cases, the inversion algorithm requires developing an appropriate way to scale the transforms before performing the inversion. This scaling algorithms that have been developed are effective, but they are somewhat ad hoc.

The purpose of this article is to propose a systematic scaling algorithm for a large class of nonprobability 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 central 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 Lucantoni5 and Choudhury, Leung, and Whitt4 our scaling algorithm here provides an alternative to other methods proposed for developing general Fourier-series inversion algorithms, e.g., see Hui9 and Hirdes20 and Pinnier and Huyrnen.49

The scaling is also important because probability distributions themselves when we want to compute very small values at large arguments with controlled relative error. For example, we need to do this in order to calculate the asymptotic parameters describing the way tail probabilities decay. Choudhury and Lucantoni5 and Abate et al.20 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.

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 Whitt4, i.e., the G/G/1 queue. In Section 7 we describe the variants of the main

Numerical Inversion of Nonprobability Transforms

Gagan L. Choudhury / AT&T Laboratories, Room 1L-338, Holmdel, NJ 07733-5010. Email: gagan@bellcore.com

Ward Whitt / AT&T Laboratories, Room 3C-178, Murray Hill, NJ 07974-6036. Email: ward@research.att.com

(Received: April 1996; revised: October 1996; accepted: October 1996)
scaling algorithm for generating functions. In Section 8 we
give the scaling algorithm for multidimensional trans-
foms, which may be Laplace transforms in some dimen-
sions and generating functions in others. We apply the multi-
dimensional scaling to Section 9 as a two-dimensional example involving a closed square network. These
the scaling is an alternative to the scaling in [9]. Finally, we state our conclusions in Section 10.

1. The Fourier-Series Method

Given a Laplace transform

\[ \mathcal{L}(f) = \int_0^\infty e^{-yt} f(t) \, dt, \]

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

By this reasoning, we are able to write

\[ f(t) = \mathcal{L}^{-1}(f) - \mathcal{L}^{-1}(e^{-yf(t)}), \]

where the periodic approximation is

\[ f(t) = \mathcal{L}^{-1}(\hat{f}(t)) = \frac{b}{y} \sum_{n=1}^{\infty} \begin{cases} A_n e^{i n \omega t} & \text{for } n = 1, 2, 3, \ldots \end{cases} \]

and the aliasing error is

\[ e_n(t) = \frac{b}{y} \sum_{n=1}^{\infty} A_n e^{-i n \omega t} \left[ 1 + \begin{bmatrix} \omega(t) \end{bmatrix} \right] \]

(set 2.2) and (2.8) of Chaudhry, Laxenworth, and Whitt[25]. These are the one-dimensional versions of the two-dimen-
sional formulas given there.

Given (1)-(2), the idea is to choose the parameter \( A \) to make the aliasing errors (2) suitably small, and then choose the parameter \( b \) to make the roundoff error in calculating (2) suitably small. The roundoff error arises with limited preci-
sion (such as double precision) because the precision of (2) is a very large number leading to the multiplication of a very small number by a very large number. The overall proce-
dure is to find the parameters \( A, b \) to make the roundoff error and the aliasing error suitably small, and then calcu-
late (3) by approximately summing the infinite series in (2).

When \( t \) is a cdf of a complementary cdf (ccdf, i.e., one

\[ f(0) = 0 \text{ for } C \to 1 \] and \( \delta, \eta, \) so that

\[ \eta(t) < Ce^{-\alpha t} \]

and the aliasing error is easily controlled. Then if \( t \) is a pdf, the aliasing error is also usually easy to control. Then we assume that

\[ f(t) < C \text{ for all } t = (1 + 2t)^2 \]

for some \( C \), which is also sufficient to have the bound (4), as can be seen from (5). For pdf's we may have \( y \to \alpha \), but the behavior below \( (1 + 2t)^2 \) 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 accu-

To the accuracy of the aliasing error is sufficient for the sinusoidal function.

2. The Scaling Algorithm

We transform (3) into a pdf using the scaled function

\[ \int_{\Delta x} f(x) \, dx = \mathcal{L}^{-1}(\hat{f}(t)), \quad t > 0, \]

which has Laplace transform

\[ f(\Delta x) = \mathcal{L}^{-1}(\hat{f}(t) + \alpha \hat{f}(t)), \]

In general we may wish to calculate very small values \( f(\Delta x) \). We assume that we are interested in controlling the relative aliasing error \( |f(t) - f(\Delta x)|/f(\Delta x) \) rather than the absolute aliasing error. Our scaling strategy is to transform the function into a pdf whose mean

To compute (7), we first compute \( f(t) \) by numerically inverting \( f(\Delta x) \) and then calculate \( f(\Delta x) \) by letting

\[ f(t) = \mathcal{L}^{-1}(\hat{f}(t) + \alpha \hat{f}(t)) \]

Our choice of parameters \( \alpha_0 \) and \( \alpha_1 \) (discussed below) is intended to make \( f(t) \) not too small or too large, but \( f(t) \) and \( \mathcal{L}^{-1}(\hat{f}(t)) \) are very small or large (even outside the floating point limits of the computer). Hence, we compute (8) using logarithms if necessary; i.e., we compute

\[ \log f(t) = \alpha_0 + \log f(\Delta x) - \log \alpha_1 \]

Now we turn to the choice of the scaling parameters \( \alpha_0 \) and \( \alpha_1 \) (5) and (7). We choose the parameters \( \alpha_0 \) and \( \alpha_1 \) so that the function \( f(t) \) is like a pdf and the desired inver-
sion point \( t \) is near the mean. To achieve this property, we assume that the desired function \( f(t) \) is nonnegative.
Theorem 23. Suppose that the function \( f \) is non-negative and let \( s^* \) be the rightmost singularity of \( \tilde{f}(s) \) with \( s^* = \infty \) if \( \tilde{f}(s) \) is analytic. For any \( m, 0 < m < \infty \), if the equation
\[
\frac{\tilde{f}(s)}{\tilde{f}'(s)} = m
\]
(10)
has a real root \( a_m \) in the interval \((s^*, m)\), then \( f_{m}(x) \) is a bounded probability density function with mean \( m \) for \( a_m \) satisfying (10) and
\[
a_m = \frac{1}{\tilde{f}(a_m)}.
\]
Moreover, \( -\tilde{f}(s)/\tilde{f}'(s) \) is decreasing in \( s \) for real \( a \) in \((s^*, m)\), so that (10) has at most one real root.

Proof. Note that
\[
-\frac{\tilde{f}(s)}{\tilde{f}'(s)} = \frac{\delta_{0}^{1} \tilde{f}(s) \, ds}{\tilde{f}(s) \tilde{f}'(s) \, ds},
\]
so that it is indeed the mean of the pdf with density \( \tilde{f}(s) = C_0 e^{-s \tilde{f}(s)} \), where \( C_0 \) is chosen so that the total mass is 1. To establish the monotonicity of \( -\tilde{f}(s)/\tilde{f}'(s) \), we use stochastic order concepts; these are discussed in the Appendix of Ross and Spiral and Shaked and Shaked. Note that the ratio of the pdf’s satisfies
\[
\frac{f_m(t)}{f_n(t)} = \frac{C_{m} e^{-w \tau \tilde{f}(s)}}{C_{n} e^{-w \tau \tilde{f}(s)}}, \quad t > 0,
\]
which is decreasing in \( t \) for \( a_m > a_n \), which implies that \( f_m \) is smaller than \( f_n \), in the likelihood ratio ordering, which in turn implies that \( f_m \) is less than \( f_n \) in stochastic order; which in tum implies that the means are ordered, i.e.,
\[
-\frac{\tilde{f}(s)}{\tilde{f}'(s)} < \frac{\tilde{f}(s)}{\tilde{f}'(s)}.
\]
Finally, given \( a_m \), \( a_n \) must satisfy (11) to make \( f_{m,n}(t) \) a proper pdf.

Hence, to compute \( f(t) \), we first scale the function \( f \) using (10) and (11) with \( m = 1 \), and then use the inversion formula in Section 1. Because the function \( -\tilde{f}(s)/\tilde{f}'(s) \) is monotone, it is relatively easy to find a root \( a_m \) to equation (10) when it exists by a simple search algorithm. For example, we can start with \( a = 0 \) and then consider \( a = a + 1 \) or \( a = a - 1 \). Afterward, increase \( a \) in absolute value geometrically (e.g., 1, 2, 4, 8, . . .) until an infinite 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
\[
\frac{d}{da} \frac{\tilde{f}(s)}{\tilde{f}'(s)} = \frac{\tilde{f}(a)}{\tilde{f}'(a)} - \frac{\tilde{f}'(a)}{\tilde{f}'(a)^2},
\]
which in turn requires the second derivative \( \tilde{f}'(s) \). 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 \( a_m \) in (10) must be to the right of all singularities of the Laplace transform \( \tilde{f} \). This should be checked. Second, it is important to be aware that a root to equation (10) need not exist even if \( \tilde{f}(s) \) is finite for one or more values of \( a \). For example, suppose that
\[
f(t) = (1 + t)^{-1},
\]
for \( c > 2 \), so that
\[
\tau = \int_{t}^{\infty} f(t) \, dt < \infty.
\]
Then \( f_{m,n}(t) \) defined by (6) has mean \( m < \tau \) in (13) for all \( a_m > 0 \), but has infinite mean for \( a_m < 0 \). Thus, it is possible to find a root to (10) for all \( m < \tau \), 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 predactor in (2) is very large. There can also be numerical difficulties with (2) if \( t \) is large and the transform \( \tilde{f} \) has singularities on the line with \( Re(s) = 0 \), because the arguments of \( \tilde{f} \) in (2) will be close to this line when \( t \) is large. When \( t \) is very small or very 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
\[
f(t) = (\tilde{f}(t)), \quad t > 0,
\]
which has Laplace transform
\[
f(\tau) = \frac{\tau}{\tau}(\tilde{f}(\tau)),
\]
(14)
we compute \( f(\tau) \) by calculating \( f(\tau)(\tilde{f}(\tau)) \) by numerically inverting \( \tilde{f}(\tau) \). For this procedure, we exploit the fact that
\[
f(1) = \tilde{f}(t) = (t).
\]
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 \( \tilde{f}(s) \) using (15) with \( t = \tau \) for very small \( t \). This difficulty may be removed by the following key observation based on the initial value theorem for Laplace transforms. If \( \tilde{f}(0) = \lim_{t \to 0} \tilde{f}(t) = \tilde{f}(0) \), then
\[
\lim_{t \to \infty} \tilde{f}(s) = \lim_{t \to \infty} \tilde{f}(1) = \frac{1}{s} \lim_{t \to \infty} \tilde{f}(t) = \frac{1}{s} \tilde{f}(0).
\]
for \( s > 0 \), \( \tilde{f}(0) = \lim_{t \to \infty} f(t) = (0) \).
The above states that if \( \lim_{\eta \to 0} f(\eta) \) is finite, then so is \( \lim_{\eta \to 0} F(\eta) \), so that \( \eta \) should be possible to rewrite the right-hand side of (15) with \( \eta = \tau \) (basically by canceling out \( t^n \) terms) such that there is no computational difficulty for arbitrarily small \( \tau \).

Alternatively, if \( f(\eta) \) has a singularity at \( \eta = 0 \), there would be a corresponding singularity of \( F(\eta) \) at \( \eta = 0 \) and any inversion procedure would not work for obvious reasons. We illustrate this using a simple example. Let \( f(\eta) \) represent the cdf of waiting time in an M/G/1 queue with utilization \( \mu \) and service time \( L(t) \). Then the LSF of \( f(\eta) \) is

\[
f(\eta) = \int_{0}^{\infty} e^{-t \cdot \eta} \cdot d\lambda(t) = \frac{(1 - \mu)}{\lambda + \mu L(t)}.
\]

From (15),

\[
\tilde{f}(\eta) = \frac{(1 - \mu)}{\lambda + \mu L(t)}
\]

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

\[
\tilde{f}(\eta) = \frac{(1 - \mu)}{\lambda + \mu L(t)}
\]

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

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

\[
\tilde{f}(\eta) = \frac{(1 - \mu)}{\lambda + \mu L(t)}
\]

and

\[
\tilde{f}(\eta) = \frac{(1 - \mu)}{\lambda + \mu L(t)}
\]

Note that (29) does have a numerical difficulty for small \( \tau \), whereas (18) does not. This is because the pdf has a singularity at \( \eta = 0 \), while the cdf does not.

4. Simple Examples

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

Example 4.1 (An exponential function). Suppose that \( f(\eta) = e^{-\lambda \eta}, \eta \geq 0 \), with Laplace transform \( \tilde{f}(\eta) = (\lambda - \eta)^{-1} \). Of course, no numerical inversion is needed in this case; this example is to illustrate the procedure. For \( \lambda > 0 \), there does not exist a finite \( C \) such that \( \tilde{f}(\eta) \geq C \) for \( \eta \geq (1 + 3\lambda) \). For \( \lambda < 0 \), \( f(\eta) \) will be very small when \( t \) is suitably large. In this case equation (10) becomes

\[
\tilde{f}(\eta) = \frac{1}{\eta - \lambda} \cdot \frac{\eta}{\lambda},
\]

so that

\[
a_0 = \lambda + \tau^n
\]

and, by (11),

\[a_0 = \frac{1}{\eta} \cdot (\lambda + \eta) = t^{-\lambda}.
\]

Hence,

\[
\tilde{f}(\eta) = \frac{1}{\tau} \cdot \eta^{-1},
\]

and

\[
\tilde{f}(\eta)(x) = \frac{1}{\tau} \cdot \eta^{-1} \cdot x \geq 0,
\]

which is the exponential pdf with mean \( \tau \), as could be predicted from Theorem 2.1.

Note that the scaled transform \( \tilde{f}(\eta) \) does not have the numerical difficulties of the original transform \( \tilde{f}(\eta) \). First, for all \( x \geq (1 + 3\lambda) \), \( \tilde{f}(\eta)(x) \leq C \), where

\[
C = \frac{\lambda}{\tau} \cdot \eta^{-1},
\]

but

\[
\frac{C}{\tilde{f}(\eta)(x)} = \tau^{-1} < 1.
\]

Second, \( \tilde{f}(\eta)(x) \) is \( 1/x \), which does not become too large or small unless \( x \) itself is very small or large.

We can address the problem of extremely large or small \( \tau \) by first using the scaling procedure in Section 3. As in (36), we let

\[
\tilde{f}(\eta) = \frac{1}{\tau} \cdot \tilde{f}(\eta)(x) = \frac{1}{\tau} \cdot \eta^{-1} = \eta^{-1}.
\]

To get \( \tilde{f}(\eta) \), we compute \( \tilde{f}(\eta) \) by inverting \( \tilde{f}(\eta) \). Hence, the inversion point is shifted from \( \tau \) to 1. Now we apply the scaling procedure in Section 2 to get

\[
1 = \frac{1}{\tau} \cdot \tilde{f}(\eta)(x) = \frac{1}{\tau} \cdot a_0 = \frac{1}{\tau} \cdot \lambda,
\]

so that \( a_1 = \lambda t \). This completes the solution, i.e.

\[
\tilde{f}(\eta)(x) = \frac{1}{\tau} \cdot \eta^{-1} \cdot x \geq 0.
\]

Hence, we calculate \( \tilde{f}(\eta)(x) \) by numerically inverting \( (1 + \lambda t)^{-1} \), which avoids all problems of small or large \( \tau \). Then we calculate \( \tilde{f}(\eta) \) by applying (8), i.e.

\[
\tilde{f}(\eta)(x) = \frac{1}{\tau} \cdot \tau^{-1} \cdot (1 + \lambda t)^{-1} \cdot \eta^{-1}.
\]

Example 4.2 (A power). Suppose that \( f(\eta) = \eta^n, \eta \geq 0 \), for some \( n > 0 \), which has Laplace transform

\[
\tilde{f}(\eta) = \frac{1}{\eta^{n+1}}.
\]

There is a genuine difficulty in the inversion for \( \tau \) near 0 if \( x \) is negative, because then \( \tilde{f}(\eta) \) is not in a singularity. Otherwise, we can apply the method of Section 3 to transform the inversion point to \( \lambda = 1 \). So hereafter assumes that the inversion point is \( \lambda = 1 \).
We solve (19) to obtain \( a_\lambda \), obtaining
\[
1 = \frac{-\ln(a_\lambda)}{\ln(\alpha)} = \frac{x + 1}{\ln(\alpha)}
\]
so that \( a_\lambda = x + 1 \). Then, by (11),
\[
a_\lambda = \left( \frac{x + 1}{\ln(\alpha)} \right) = \left( \frac{1}{\ln(\alpha)} \right) + \frac{x + 1}{\ln(\alpha)}.
\]

Hence,
\[
f_{\text{max}}(x) = a_\lambda f(x + a_\lambda) = \left( \frac{x + 1}{\ln(\alpha)} \right)^{x + 1}
\]
and
\[
f_{\text{max}}(x) \leq \left( \frac{x + 1}{\ln(\alpha)} \right)^{x + 1} = \left( \frac{x + 1}{\ln(\alpha)} \right)^{x + 1}, \quad \lambda > 0,
\]
which we recognize as a gamma pdf with shape and scale parameter \( x + 1 \), and so mean \( x + 1 \) and variance \( (x + 1)^2 \).

The scaled function in terms of \( f_{\text{max}}(x) \) in (21) and (22) is much better behaved than the original pair \( f(x), \beta \). First, the function \( f_{\text{max}}(x) \) is strictly decreasing for \( x > 1 \), so that \( C_{\text{max}}(x) < 1 \). Second, the quantity \( f_{\text{max}}(x) \) 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 cdf \( F(x) \equiv 1 - F(0) \)
has the asymptotic form
\[
F(x) \approx a x^\lambda e^{-bx} \quad \text{as} \quad t \to \infty
\]
for positive constants \( a \) and \( \lambda \) and actuary constant \( \beta \), where \( F(0) \rightarrow 0 \) as \( t \to \infty \) means that \( F(0)/g(t) \rightarrow 1 \) as \( t \to \infty \). Chouluck and Lucyricn17 and Abate and Whitt18 showed how the asymptotic parameters \( a, \lambda, \) and \( \beta \) 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 \( F(x) \) for large \( x \). It suffices to solve the three equations
\[
\log(F(t_i)) = \log(a) + \lambda \log(t_i) - \beta t_i,
\]
with \( i = 1, 2, 3 \) for \( a, \lambda, \) and \( \beta \). We use the scaling to compute \( F(t_i) \) by numerical inversion from the Laplace transform \( (1 - F(t))/\sigma \) for suitable 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, which was analyzed in Abate and Whitt.18 The associated reflected Brownian motion first-moment cdf, denoted by \( H(x) \), is known to have the asymptotic form
\[
F(x) \sim \frac{1}{\sqrt{2\pi}} \frac{1}{\sqrt{t+2\beta}} e^{-\frac{x^2}{2t+4\beta}} \text{as} \quad t \to \infty.
\]
see Corollary 3.3.5 on p. 567 of [2]. We will verify that \( \eta = \frac{1}{\alpha} \), \( a = \frac{2t+2\beta}{\sqrt{2\pi}} \), and \( \beta = -\frac{1}{4t} \) by applying numerical transmformation. The Laplace transform of \( H(x) \) is
\[
\hat{H}(s) = \frac{1 - \frac{\beta}{s}}{s + 1} \left( 1 + \sqrt{1 + 2s} \right); \quad \text{see p. 568 of [2].}
\]
From (26), we obtain the derivative
\[
\hat{H}'(s) = \left( 2 + 3s \right) \left( 2 + 3s + (2 + 5s + 2s^2) \right) / \left( 2 + 2s \right)^2
\]
so that
\[
r(s) = \frac{\hat{H}'(s)}{\hat{H}(s)} = \frac{1 + 2s}{2(2 + 2s + s^2)} = \frac{1}{2s} \left( 1 + \frac{1}{2s} \right)^{1/2} \text{as} \quad s \to \infty.
\]
A partial check is obtained by noting that, by L'Hôpital's rule, \( r(s) = 0.5 \), which agrees with the known formula for the mean, combine Corollaries 13.4 and 13.51 of [2].

We computed \( r(s) \) for several values of \( s \) 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 parameter \( t = 2 \) and \( t = 3 \). The results are displayed in Table I. We get the first four values equally accurately \( r(s) \) with out scaling, but for \( t = 50 \) there are significant errors in the unscaled algorithm. (For the cases with \( t > 100 \), the unscaled algorithm using double precision cannot distinguish the exact values from 0.) Note that at \( t = 2000 \), \( r(s) \) is even below the floating-point limit of the computer we used. For the example the scaling parameter \( \alpha_\lambda \) approaches \(-0.5 \) and \( a_\lambda \) approaches 0.5 as \( s \) approaches infinity. (This is easy to show analytically as well.)

Table I. Numerical Results for the First-Moment cdf \( F(x) \) of Reflected Brownian Motion

<table>
<thead>
<tr>
<th>( t )</th>
<th>( \alpha_\lambda )</th>
<th>( \beta_\lambda )</th>
<th>( \lambda_\lambda )</th>
<th>( \eta_\lambda )</th>
<th>( \eta_\lambda )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0</td>
<td>0.56769016</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>5.0</td>
<td>0.36493865</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>10.0</td>
<td>0.21869605</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>20.0</td>
<td>0.43035989</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>50.0</td>
<td>0.96140693</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>100.0</td>
<td>0.59086554</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>200.0</td>
<td>0.23013006</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>500.0</td>
<td>0.97649683</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>1000.0</td>
<td>0.39759892</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
<tr>
<td>2000.0</td>
<td>0.95020748</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
<td>0.01173956</td>
</tr>
</tbody>
</table>
is often spectacularly fast. We can compute estimates of the asymptotic parameter based on any three values of \( t \), but we cannot expect them to be very accurate because of the accuracy in Table I. If we use the last three values, then we get

\[
\alpha = 1.46075568, \quad \beta = -1.4871261, \quad \eta = 0.50000599,
\]

where the true values are

\[
\alpha = 1.59556591, \quad \beta = -1.5, \quad \eta = 0.5.
\]

The accuracy for the asymptotic decay rate is excellent, but the accuracy for \( \alpha \) and \( \beta \) is not too good. However, overall the accuracy is good enough for many practical purposes. Just as in the context of moments,\(^\text{11} \) the accuracy may be greatly enhanced. Here we can announce a multitemp asymptotic expansion and get estimates based on many points. The main point hereinafter, however, is that the scaling enables us to accurately compute very small tail probabilities.

5. A Multiplying Example

We now consider the MAPPP/D/1/growing model used in [10] to study the effectiveness of effective bandwidths to describe buffer overflow probabilities with statistical multiplexing. 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 periods and the off periods have exponential distributions. The mean off-period \( \delta \) in 10 times the mean on-period \( \omega \). During the on period cells arrive according to a Poisson process at (peak) rate \( \rho \). The mean number of arrivals in an on period is \( \rho \delta = \omega \). The source rates are approximately adjusted so that the long-term utilization of the output channel is 0.3 for each \( N \).

We consider the case of \( N = 2 \) and \( N = 24 \). The case \( N = 24 \) is the example in Section II of [10]. In that case \( \omega = 436.6 \) \( \times \) \( 436.6 \) and \( \rho = 0.1375 \). The case \( N = 2 \) is an alternative considered in Section IV of [10].

The Laplace-Shifts transform \( \Phi(s) \) of the steady-state waiting-time distribution (used to approximate the buffer overflow probability) is given in Eq. 8.4 of [10]. The waiting-time tail probability \( P(V > x) \) is calculated by numerically inverting the transform \( \Phi(s) = \int_{0}^{\infty} e^{-sx} \Phi(s) \) for \( x \). However, there are two difficulties. First, the \( D \) service is not so easy to treat. Hence, we approximate the \( D \) service-time transform \( e^{-x} \) by an accurate asymptotic probability transform that is a hybrid of Padé and Erlang approximations. In particular, we use the approximation\(^\text{12} \)

\[
e^{-x} \approx \left( 1 + \frac{x}{N} + \frac{1}{(N+1)!} \right) e^{-x}, \quad x > 0.
\]

5.1 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 value problem. In Section 2, and a minor modification of the same procedure applies.

Given the generating function \( q(x) = \sum_{k=0}^{\infty} q_{k} \), we construct the scaled sequence \( q_{k}(x) = q_{k}(\lambda x) \) by setting

\[
q_{k}(x) = \lambda q_{k}(x), \quad k > 0,
\]

which has generating function

\[
q_{k}(x) = \left( \frac{q_{k}(x)}{q_{0}(x)} \right).
\]

The following theorem is the discrete analog of Theorem 2.1. For its statement, let \( q_{k}(x) \) be the derivative of \( q(x) \). Then

\[
\sum_{k=0}^{\infty} q_{k}(x) = \lambda
\]

has a real root \( x_{k} \) in the interval \( (0, \pi) \), then \( q_{k}(x) \) is

\[
\text{very accurate in predicting the waiting-time tail probability. We intend to discuss the approximation of transforms such as } e^{-x}\text{ in more detail elsewhere.}^{\text{11}} \text{ Our approach is similar in spirit to Aker and Arikan.}^{\text{12}}
\]

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

\[
h'(x) = \frac{h(x + \delta/2) - h(x - \delta/2)}{\delta} + \frac{3}{128} h''(x) + \frac{5}{768} h'''(x) + \ldots
\]

where \( h''(x) = (x + \delta/2) - (x - \delta/2) \) and \( h''(x) = \delta \). This is based on Bessel's interpolation formulae; see equation III-C-11 on p. 100 of Kopal.\(^{\text{14}} \) The formula \( \Phi(s) \) is based on computing function values at \( x = 1/2 \) and \( x = 1/2 + j/2 \) for \( j = 0, 1, 2, \ldots, n \). We found that \( h \max(0.001, 0.001) \) 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 because it is the negative of the asymptotic exponential decay rate of the tail probability).\(^{\text{15}} \)

Numerical values of the tail probabilities and the scaling parameters \( a_{0} \) and \( a_{1} \) as a function of the buffer size are given for the two cases \( N = 2 \) and \( N = 24 \) in Table II. With standard double precision and without scaling, the inversion algorithm would typically have error of order \( 10^{-14} \) and hence all probabilities below \( 10^{-10} \) would have large relative errors. However, with the scaling, the algorithm maintains accuracy to \( 10^{-16} \). The accuracy was confirmed by independent computations with roundoff control parameters \( e = 3 \) and \( \varepsilon = 2 \).
Table II. Tail Probabilities and Scaling Parameters $\alpha$ and $\delta_0$ for the MMPP/D/1 Model in Section 6 as a Function of the Number $N$ of Sources and the Buffer Size

<table>
<thead>
<tr>
<th>Buffer Size</th>
<th>Tail Prob. $N = 2$</th>
<th>$m_1$</th>
<th>$\alpha_1$</th>
<th>Tail Prob. $N = 24$</th>
<th>$m_0$</th>
<th>$\alpha_0$</th>
</tr>
</thead>
<tbody>
<tr>
<td>200</td>
<td>0.23564e-1</td>
<td>-0.01363</td>
<td>0.00077</td>
<td>0.63602e-5</td>
<td>-0.01777</td>
<td>1.152</td>
</tr>
<tr>
<td>400</td>
<td>0.36150e-3</td>
<td>-0.01577</td>
<td>0.00428</td>
<td>0.6319e-7</td>
<td>-0.01788</td>
<td>1.116</td>
</tr>
<tr>
<td>600</td>
<td>0.5660e-5</td>
<td>-0.01652</td>
<td>0.00398</td>
<td>0.974e-9</td>
<td>-0.01793</td>
<td>1.091</td>
</tr>
<tr>
<td>800</td>
<td>0.23864e-6</td>
<td>-0.01690</td>
<td>0.00329</td>
<td>0.10530e-10</td>
<td>-0.01795</td>
<td>1.071</td>
</tr>
<tr>
<td>1000</td>
<td>0.6971e-8</td>
<td>-0.01714</td>
<td>0.00186</td>
<td>0.396e-12</td>
<td>-0.01797</td>
<td>1.064</td>
</tr>
<tr>
<td>1200</td>
<td>0.8569e-10</td>
<td>-0.01729</td>
<td>0.00125</td>
<td>0.8795e-14</td>
<td>-0.01798</td>
<td>1.008</td>
</tr>
<tr>
<td>1400</td>
<td>0.0037e-13</td>
<td>-0.01741</td>
<td>0.00135</td>
<td>0.2084e-15</td>
<td>-0.01799</td>
<td>1.026</td>
</tr>
<tr>
<td>1600</td>
<td>0.0331e-12</td>
<td>-0.01756</td>
<td>0.00119</td>
<td>0.5134e-17</td>
<td>-0.01800</td>
<td>1.014</td>
</tr>
<tr>
<td>1800</td>
<td>0.50347e-14</td>
<td>-0.01766</td>
<td>0.00106</td>
<td>0.12971e-18</td>
<td>-0.01801</td>
<td>1.003</td>
</tr>
<tr>
<td>2000</td>
<td>0.6437e-16</td>
<td>-0.01768</td>
<td>0.00096</td>
<td>0.3355e-20</td>
<td>-0.01802</td>
<td>0.992</td>
</tr>
</tbody>
</table>

**transferring probability mass function with mean $\lambda$ for $\alpha_1$ satisfying (33) and**

\[ \alpha_0 = 1 / q^*(\alpha_0). \]  

(34)

Moreover, $q^*(\alpha_1) / q^*(\alpha_0)$ is strictly increasing in $\alpha$ for positive $\alpha$ in $(0, \infty)$, so that (33) has at most one real root.

**Proof.** Note that

\[ q^*(\alpha_1) / q^*(\alpha_0) = \frac{\sum_{k=1}^{\infty} k \alpha_1^k}{\sum_{k=1}^{\infty} k \alpha_0^k}, \]

so that it is the mean of the pmf $q_\alpha(k) = C \alpha^k \delta_0^k$, where the constant $C$ is chosen to make the total mass 1. Note that the ratio of pmf's satisfies

\[ q_\alpha(k) / q_{\alpha_0}(k) = C \alpha^k \delta_0^k, \]

which is increasing in $k$ for $\alpha > \alpha_0$, so that $q_{\alpha_0}$ is larger than $q_{\alpha}$ in the discrete likelihood ratio ordering, which implies stochastic order and the ordering of the means. ■

Hence, if we want to calculate $q_\alpha$, then we would scale by (31) with $\alpha_0$ chosen to satisfy (30) 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 dimension discrete (generating functions) and other dimensions continuous (Laplace transforms). We illustrate this section by discussing the bivariate mixed case. Here we call the desired bivariate function that is in one dimension and a pdf in the other dimension simply a pdf.

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

\[ F(s, z) = \int_0^\infty \int_0^{\infty} f(t, k) z^k e^{-st} dt. \]  

(35)

We introduce the scaled function

\[ f_k(s, z) = a_k^{-\alpha_k} F(s, k), \]

for $a = (a_0, a_1, a_2)$, which has transform

\[ F_k(s, z) = a_k^{\alpha_k} (1 + \alpha_k a_2). \]

(36)

**Theorem 8.1.** Suppose that the bivariate function $f$ is nonnegative. For any $m_0 < m_2 < m_1$, and integer $m_0 < m_2 < m_1$ if the pair of equations

\[ \frac{d}{dz} \log f(s, z) \bigg|_{z=m_1} = m_1 \]

and

\[ \frac{d}{dz} \log f(s, z) \bigg|_{z=m_2} = m_2 \]

(38)

(39)

has a solution $(\alpha_1, \alpha_0)$ such that $f_k(s, z)$ is analytic for $|s| < 1$ and $f_k(z) > 0$, then $f_k$ in (36) is a bivariate pdf with means $m_0$ and $m_2$ in the two dimensions, provided that

\[ \alpha_0 = \frac{1}{f(m_0, m_2)}. \]

(40)

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

\[ \frac{d}{ds} \log f(s, z) = \frac{-(f(s, z)/f(s, z))}{f(s, z)}. \]

(41)
and

\[
\frac{3}{2} \log f(s, x) = \frac{3}{2} \frac{f(x, z)}{f(z, x)}. 
\]

Unfortunately, however, it is not as easy to find the solution to the pair of equations (38) and (39) as it is to find the solution to 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 \(a_x\) and \(a_y\), and then find candidate values of \(a_x\) and \(a_y\), again using the appropriate one-dimensional algorithm. Then repeat, fixing \(a_x\) and \(a_y\), and so forth, stopping after a few iterations because 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 \(f(x, y)\) 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 \(\frac{1}{4}\) to each of the points \((3, 0), (0, 3), (1, 2),\) and \((2, 1)\) in \(\mathbb{R}^2\). Let the two means be \(m_1 = 1\) and \(m_2 = 2\). Then the two mean equations become

\[
3x + 2y + x + 2y = 1
\]

and

\[
3x + 2y + x + 2y = 1
\]

where \(x = e^{-y}\) and \(y = e^{-x}\). These equations both reduce to the simple equation

\[
x + y = 1 - e^{x+y} = 0.
\]

Dividing through (41) by \(y^2\), we see that \(x = y\) satisfies the equation.

\[
x + y = 1 - 0.
\]

which has one positive root \(x = 0.65730\). Hence \((x, y) = (0.65730, y)\) is a solution to (41) for all \(y > 0\).

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 \(a_x\) and \(a_y\), but even this leaves open the questions of existence, uniqueness, and convergence. For our monotonicity result, we exploit the notion of total positivity and multivariate likelihood ratio ordering, see Karlin and Krickeberg[12] and Whitt[13]. One bivariate pdf \(f(x, y)\) is said to be less than or equal to another \(f'(x, y)\) in the multivariate likelihood ratio order if

\[
f(x, y) < f(x, y)\quad \text{for all vectors } \mathbf{x} = (x_1, x_2) \text{ and } \mathbf{y} = (y_1, y_2), \text{ where } x_1 < x_2 \text{ and } y_1 < y_2,
\]

and \(x_1 < y_1 \text{ and } x_2 < y_2\). Let \(x_1 = \min(x, y_1)\) and \(x_2 = \max(x, y_1)\), and we write \(x_1 < y_1 < y_2 < x_2\). A single bivariate pdf \(f\) is said to be totally positive of order 2 (TP2) if

\[
f_{x,y}(x_2, y_2) < f_{x,y}(x_1, y_2) \text{ is increasing in } a_x \text{ and } a_y.
\]

Proof. Given the \(x\)-to-\(a\) conditions, the proof is the two-dimensional generalization of the proof of Theorem 2.1. Given that \(f\) is TP2, the multivariate likelihood ratio order is equivalent to the ratio \(f_{x,y}(x_2, y_2) / f_{x,y}(x_1, y_2)\) being decreasing in the vector \(x\); see Theorem 3 of [13]. The multivariate likelihood ratio order implies stochastic order, which in turn implies an ordering of the means.

9. A Two-dimensional Queuing Network Example

Consider the two-dimensional transform

\[
g(x, y) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} g(n, m) x^n y^m e^{\beta m}
\]

\[
= \frac{\exp(-\beta x)}{1 - \frac{\beta}{\mu_1} x \mu_2 y}
\]

This is the generating function of the normalization constant in a closed queuing 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 is this article works in that context.

We work with the scaled generating function

\[
g_k(x, y) = \frac{g(n, m)}{n! m!}
\]

The scaling parameters \(n_k\) and \(m_k\) are obtained from the two equations

\[
\begin{align*}
\nu_1 &= \frac{3}{2} \log g_k(x, y) \\

\nu_2 &= \frac{3}{2} \log g_k(x, y)
\end{align*}
\]

We must solve the pair of nonlinear equations in (44). As suggested earlier, we can fix \(a_2\) and search for the value \(a_1\) that satisfies the equation for \(i = 1, 2\). Next, fixing \(a_1\) at the value obtained, we search for the value of \(a_2\) that satisfies the
Table III. Numerical Results for the Normalization Constant $g_{n_1}$ in a Closed Queueing Network with Two Chains

<table>
<thead>
<tr>
<th>$n_1$</th>
<th>$n_2$</th>
<th>$g_{n_1}$</th>
<th>$g_{n_2}$</th>
<th>$g_{n_3}$</th>
<th>$g_{n_4}$</th>
<th>$g_{n_5}$</th>
<th>$g_{n_6}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>2</td>
<td>0.243883R+04</td>
<td>0.240070</td>
<td>0.104028</td>
<td>0.86627E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>30</td>
<td>20</td>
<td>0.82774R+03</td>
<td>0.294997</td>
<td>0.130311</td>
<td>0.589928E-02</td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>200</td>
<td>0.979460E+31</td>
<td>0.299851</td>
<td>0.133032</td>
<td>0.364701E-03</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3000</td>
<td>2000</td>
<td>0.335159E+31</td>
<td>0.299945</td>
<td>0.133303</td>
<td>0.562179E-04</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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 $n_2$ increases. By contrast, the two-dimensional Newton-Raphson method (see Press et al., Chapter 5) 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:

- $\lambda_1 = 1$, $\mu_2 = 1$, $\mu_1 = 2$, $\mu_2 = 3.$

The results for several values of the chain populations $n_1$ and $n_2$ are displayed in Table III. 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 (Sections 2 and 3), one-dimensional generating functions (Section 7) and multidimensional transforms (Section 6) of nonprobability functions in order to control the scaling 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 pdf and transform the inversion point to the mean. The required equation in one dimension is usually easy to solve, but solved 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 trough) 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 9, 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 quasiparticle network example in Section 9, but the multidimensional scaling equations still need to be better understood.

References


18. W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetter-


