ORSA Journal on Computing
Vol. 7, No. 1, Winter 1995

0899-1499/95/0700-0036 $01.25

Numerical Inversion of Laplace Transforms of Probability Distributions

JOSEPH ABATE / *900 Hammond Road, Ridgewood, NJ 07450-2908*

WARD WHITT / *AT & T Bell Laboratories, Room 2C-178, Murray Hill, NJ 07974-0636; Email: wow@research.att.com*

(Received: May 1992; revised: August 1992; accepted: September 1993)

We present a simple algorithm for numerically inverting Laplace transforms. The algorithm is designed especially for probability cumulative distribution functions, but it applies to other functions as well. Since it does not seem possible to provide effective methods with simple general error bounds, we simultaneously use two different methods to confirm the accuracy. Both methods are variants of the Fourier-series method. The first, building on Dubner and Abate[15] and Simon, Stroot, and Weiss,[32] uses the Bromwich integral, the Poisson summation formula and Euler summation; the second, building on Jagerman,[23, 24] uses the Post-Widder formula, the Poisson summation formula, and the Stehfest[33] enhancement. The resulting program is short and the computational experience is encouraging.

The recent emphasis on computational probability is increasing the value of stochastic models in operations research. It is becoming standard for modeling and analysis to include algorithms for computing probability distributions of interest. Many tools have been developed and are being developed for this purpose. Since probability distributions can often be characterized in terms of transforms (e.g., in queueing theory), it is natural to include numerical transform inversion among these tools.

Our purpose in this paper is to present a convenient algorithm for calculating probability cumulative distribution functions (cdf's) and other functions by numerically inverting Laplace transforms. We have described this algorithm as part of a rather extensive review of the Fourier-series method in [4]. Our purpose here is to give a concise account that will be easy to understand and apply.

We contend that numerical inversion of Laplace transforms is much easier than it is often made to seem. Nevertheless, it does not seem possible to provide effective methods with simple general error bounds that are independent of the function under consideration. Hence, as suggested by Davies and Martin,[13] we propose using two different methods, each without a complete error analysis. Assuming that the two methods agree to within desired precision, we can be confident of the computation.

The two methods we describe are both variants of the Fourier-series method, but they are dramatically different, so that they can be expected to serve as useful checks on each other. The Fourier-series method can be interpreted as numerically integrating a standard inversion integral by means of the trapezoidal rule (which turns out to be effective for the oscillating integrands under consideration). The same formula is obtained by using the Fourier series of an associated periodic function constructed by aliasing—hence the name. The key mathematical result is the Poisson summation formula, which identifies the discretization error associated with the trapezoidal rule and thus helps bound it. For more information about the Fourier-series method, including a survey of the literature and several numerical examples, see [4].

Our object is to calculate values of a real-valued function $f(t)$ of a positive real variable t for various t from the Laplace transform

$$\hat{f}(s) = \int_0^\infty e^{-st} f(t)\, dt, \tag{1}$$

where s is a complex variable with nonnegative real part. We think of $f(t)$ as being a complementary cdf (the probability of the interval (t, ∞)), but this is not essential. (In probability applications we typically know that we have a complementary cdf, so that there is nothing extra to verify.) However, we do use the fact that $|f(t)| \leq 1$ for all t in our error analyses. (This is sufficient for the Laplace transform (1) to be well defined.) The methods work better when f is suitably smooth. Indeed a different variant of the Fourier-series method for generating functions should be used for cdf's of lattice distributions, see [5] and Section 5 of [4]. When f is otherwise not sufficiently smooth (continuous and differentiable), it may help to perform convolution smoothing before doing the inversion. This lack of smoothness in f may be recognized from advanced knowledge about f or it may be revealed by insufficient precision (lack

Subject classifications: Probability, distributions: calculation by transform inversion. Queues, algorithms: Laplace transform inversion. Mathematics, functions: Laplace transforms.

Other key words: computational probability, numerical transform inversion, Laplace transforms, cumulative distribution functions, Fourier-series methods, Poisson summation formula, Bromwich inversion integral, Post-Widder inversion formula.

of agreement of the two procedures) when using the algorithm described here. If satisfactory precision cannot be obtained by adjusting the parameters of the algorithm, then it is natural to consider convolution smoothing, which is discussed in Section 6 of [4]. The Laplace transform inversion algorithm of Platzman, Ammons, and Bartholdi[30] is a variant of the Fourier-series method that exploits convolution smoothing from the start.

A feature of our algorithm is that it is intended for computing $f(t)$ at single values of t. This is suitable for many applications, but if values at many time points are desired, then it is natural to exploit the fast Fourier transform (FFT), as discussed at the end of Section 4 in [4]. See Dahlquist,[12] Embrechts, Grübel, and Pitts,[16] and Grübel[19] for recent work in this direction.

Our algorithm requires that we be able to evaluate the real part of the Laplace transform $\hat{f}(s)$ at any desired complex s. This is straightforward when the transform is given explicitly. However, often the Laplace transform is given implicitly via a functional equation, as with the busy-period distribution in the M/G/1 queue. We discuss methods for iteratively solving such functional equations in [6].

A significant application of the algorithm here is to compute the standard steady-state probability distributions in the BMAP/G/1 queue, which has a batch Markovian arrival process; see Lucantoni.[27] Choudhury, Lucantoni, and Whitt[11] have combined the numerical inversion algorithms here and in [4, 5] with the algorithms of Lucantoni[27] for this purpose. The overall algorithm has been applied to evaluate simple exponential approximations for tail probabilities in the BMAP/G/1 queue in Abate, Choudhury, and Whitt.[2, 3] Choudhury and Lucantoni[8] have developed a numerical inversion algorithm to calculate any number of moments plus the asymptotic decay rates and associated asymptotic constants of the steady-state distributions. Abate, Choudhury, and Whitt[1] have also applied the algorithm here to compute the steady-state waiting-time distribution in the GI/G/1 queue.

As indicated on p. 35 of [4], the inversion algorithm can be extended to higher dimensions. Multi-dimensional inversion algorithms are developed by Choudhury, Lucantoni, and Whitt[9] and applied to queueing problems in [10, 28].

Here is how the rest of this paper is organized. In Sections 1 and 2 we present the two methods. In Section 3 we discuss implementation and display sample versions of the computer programs. In Section 4 we present a new connection between the two methods (which does not prevent them from providing useful checks on each other). Finally, in Section 5 we draw conclusions.

1. The First Method: EULER

Our first method, which we call EULER because we employ Euler summation, is based on the Bromwich contour inversion integral, which can be expressed as the integral of a real-valued function of a real variable by choosing a specific contour. Letting the contour be any vertical line $s = a$ such that $\hat{f}(s)$ has no singularities on or to the right of it, we obtain

$$\begin{aligned} f(t) &= \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} e^{st}\hat{f}(s)\,ds = \frac{1}{2\pi}\int_{-\infty}^{\infty} e^{(a+iu)t}\hat{f}(a+iu)\,du \\ &= \frac{e^{at}}{2\pi}\int_{-\infty}^{\infty}(\cos ut + i\sin ut)\hat{f}(a+iu)\,du \\ &= \frac{e^{at}}{2\pi}\int_{-\infty}^{\infty}[\mathrm{Re}(\hat{f}(a+iu))\cos ut \\ &\qquad -\mathrm{Im}(\hat{f}(a+iu))\sin ut]\,du \\ &= \frac{2e^{at}}{\pi}\int_{0}^{\infty}\mathrm{Re}(\hat{f}(a+iu))\cos ut\,du, \end{aligned} \tag{2}$$

where $i = \sqrt{-1}$ and $\mathrm{Re}(s)$ and $\mathrm{Im}(s)$ are the real and imaginary parts of s; see pp. 4, 18 of Doetsch.[14] We calculate the integral (2) approximately. We use the Fourier-series method (the Poisson summation formula) to replace the integral by a series (which corresponds to the trapezoidal rule) with a specified discretization error. Since the series is nearly alternating, we apply Euler summation to accelerate convergence (approximately calculate the infinite sum). This last step seems to be very effective, but for it there is no error bound.

The Fourier-series method for numerically inverting Laplace transforms (and identifying the discretization error) was first proposed by Dubner and Abate.[15] The use of Euler summation in this context was proposed by Simon, Stroot, and Weiss,[32] but the approach was not widely adopted. An essentially equivalent algorithm was developed by Hosono[20, 21] and popularized in Japan as the FILT (fast inversion of Laplace transforms) in [22]; it can be based on the complementary form of (2),

$$f(t) = \frac{-2e^{at}}{\pi}\int_{0}^{\infty}\mathrm{Im}(\hat{f}(a+iu))\sin ut\,du, \tag{3}$$

but Hosono actually derives it in a different way. Hosono's variant of EULER was applied by Bertsimas and Nakazato.[7]

As indicated above, we numerically evaluate the integral (2) by means of the trapezoidal rule. If we use a step size h, then this gives

$$\begin{aligned} f(t) \approx f_h(t) \equiv{}& \frac{he^{at}}{\pi}\mathrm{Re}(\hat{f})(a) \\ &+ \frac{2he^{at}}{\pi}\sum_{k=1}^{\infty}\mathrm{Re}(\hat{f})(a+ikh)\cos(kht). \end{aligned} \tag{4}$$

Letting $h = \pi/2t$ and $a = A/2t$, we obtain the nearly alternating series

$$\begin{aligned} f_h(t) ={}& \frac{e^{A/2}}{2t}\mathrm{Re}(\hat{f})\left(\frac{A}{2t}\right) \\ &+ \frac{e^{A/2}}{t}\sum_{k=1}^{\infty}(-1)^k\mathrm{Re}(\hat{f})\left(\frac{A+2k\pi i}{2t}\right). \end{aligned} \tag{5}$$

(This is (21) of Dubner and Abate.[15])

We now use the Poisson summation formula to identify the discretization error associated with (5). The essential idea is to replace the damped function $g(t) \equiv e^{-bt}f(t)$ for $b > 0$ by the *periodic function*

$$g_p(t) = \sum_{k=-\infty}^{\infty} g\left(t + \frac{2\pi k}{h}\right) \tag{6}$$

of period $2\pi/h$ (which we assume can be done; it clearly can when $|f(t)| \leqslant 1$ for all t). We then represent the periodic function g_p by its *complex Fourier series*

$$g_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{ikht}, \tag{7}$$

where c_k is the kth *Fourier coefficient* of g_p, i.e.,

$$\begin{aligned} c_k &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} g_p(t)e^{-ikht}\,dt \\ &= \frac{h}{2\pi}\int_{-\pi/h}^{\pi/h} \sum_{k=-\infty}^{\infty} g\left(t + \frac{2k\pi}{h}\right)e^{-kht}\,dt \\ &= \frac{h}{2\pi}\int_{-\infty}^{\infty} g(t)e^{-ikht}\,dt \\ &= \frac{h}{2\pi}\int_{0}^{\infty} e^{-bt}f(t)e^{-ikht}\,dt \\ &= \frac{h}{2\pi}\hat{f}(b + ikh). \end{aligned} \tag{8}$$

Combining (6)–(8), we obtain a version of the *Poisson summation formula*

$$\begin{aligned} g_p(t) &= \sum_{k=-\infty}^{\infty} g\left(t + \frac{2\pi k}{h}\right) \\ &= \sum_{k=-\infty}^{\infty} f\left(t + \frac{2\pi k}{h}\right)e^{-b(t+2\pi k/h)} \\ &= \frac{h}{2\pi}\sum_{k=-\infty}^{\infty} \hat{f}(b + ikh)e^{ikht}. \end{aligned} \tag{9}$$

Letting $h = \pi/t$ and $b = A/2t$ in (9), we obtain

$$\begin{aligned} f(t) &= \frac{e^{A/2}}{2t}\sum_{k=-\infty}^{\infty} (-1)^k \mathrm{Re}\,\hat{f}\left(\frac{A + 2k\pi i}{2t}\right) \\ &\quad - \sum_{k=1}^{\infty} e^{-kA}f((2k+1)t). \end{aligned} \tag{10}$$

Note that the first term on the right in (10) coincides with the trapezoidal-rule approximation in (5), so that the second term on the right in (10) gives the discretization error associated with the trapezoidal rule, i.e.,

$$e_d = \sum_{k=1}^{\infty} e^{-kA}f((2k+1)t). \tag{11}$$

If, as in probability applications (e.g., when $f(t)$ is a complementary cumulative distribution function) $|f(t)| \leqslant 1$ for all t, then the error is bounded by

$$|e_d| \leqslant \frac{e^{-A}}{1 - e^{-A}}, \tag{12}$$

which is approximately equal to e^{-A} when e^{-A} is small. Hence, to have at most $10^{-\gamma}$ discretization error, we let $A = \gamma \log 10$. (We often use $A = 18.4$ to achieve 10^{-8} discretization error.) Obviously (11) can also be used to obtain discretization error bounds under other assumptions about f.

The remaining problem is to numerically calculate (5), which involves an infinite sum. Since the sum would be an alternating series if $\mathrm{Re}(\hat{f}((A + 2k\pi i)/2t))$ would have constant sign for all k, it is natural to consider acceleration methods for alternating series. Following Simon, Stroot, and Weiss,[32] we suggest using Euler summation. Euler summation is one of the more elementary acceleration techniques; see Johnsonbaugh[25] and Chapters 2 and 12 of Wimp.[34] We use it because of its simplicity. For practical purposes, it seems to provide adequate computational efficiency.

Euler summation can be very simply described as the weighted average of the last m partial sums by a binomial probability distribution with parameters m and $p = 1/2$. (Surprisingly, the binomial averaging representation of Euler summation does not seem well known. This representation is discussed in §4.3.2 of Wimp,[34] however. A variant is used by Hosono.[22]) In particular, let $s_n(t)$ by the approximation $f_h(t)$ in (5) with the infinite series truncated to n terms, i.e.,

$$s_n(t) = \frac{e^{A/2}}{2t}\mathrm{Re}(\hat{f})\left(\frac{A}{2t}\right) + \frac{e^{A/2}}{t}\sum_{k=1}^{n} (-1)^k a_k(t), \tag{13}$$

where

$$a_k(t) = \mathrm{Re}(\hat{f})\left(\frac{A + 2k\pi i}{2t}\right). \tag{14}$$

We apply Euler summation to m terms after an initial n, so that the Euler sum (approximation to (5)) is

$$E(m, n, t) = \sum_{k=0}^{m} \binom{m}{k} 2^{-m} s_{n+k}(t), \tag{15}$$

for $s_n(t)$ in (13). Hence, (15) is the binomial average of the terms $s_n, s_{n+1}, \ldots, s_{n+m}$. (We typically use $m = 11$ and $n = 15$, increasing n as necessary. As in other contexts, the acceleration typically drastically reduces the required computation.) The overall computation is specified by (13)–(15).

In order to *estimate* the error associated with Euler summation, we suggest using the difference of successive terms, i.e., $E(m, n+1, t) - E(m, n, t)$. Our experience indicates that this usually is a good error estimate, but not always so. For example, this is a poor estimate for the (non-smooth) example in Section 11 of [4].

Remark 1. In order for Euler summation to be effective, we would like $a_k(t)$ in (14) to be of constant sign for all sufficiently large k (even though this condition is neither necessary nor sufficient for Euler summations to be effective). It is significant that this property holds under extra smoothness conditions. (We thank our colleague John Morrison for assistance here.) To see this, note that

$$\mathrm{Re}(\hat{f}(u+iv)) = \int_0^\infty \cos vt g(t)\,dt, \quad (16)$$

where $g(t) = e^{-ut}f(t)$. Assuming that f is twice continuously differentiable, so is g. Assuming in addition that $f(0) = 1$ and $g(\infty) = g'(\infty) = 0$, we can apply integration by parts twice to obtain

$$\mathrm{Re}(\hat{f}(u+iv)) = \frac{u - f'(0)}{v^2} - \frac{1}{v^2}\int_0^\infty \cos vt g''(t)\,dt. \quad (17)$$

Assuming further that $g''(t)$ is integrable, we can apply the Riemann-Lebesgue lemma (p. 514 of Feller[18]) to deduce that

$$\mathrm{Re}(\hat{f}(u+iv)) = \frac{u - f'(0)}{v^2} + o\left(\frac{1}{v^2}\right) \quad \text{as} \quad v \to \infty. \quad (18)$$

Under (18), $\mathrm{Re}(\hat{f}(u+iv))$ is eventually positive for all sufficiently large v provided that $u > f'(0)$, which holds when $u > 0$ and f is a complementary cdf. Moreover, the approximating function $[u - f'(0)]/v^2$ has additional structure which makes Euler summation effective. The real difficulties with Euler summation in this context occur when f does not have enough smoothness; see Remarks 6.8 and 6.9 and Sections 11 and 14 of [4]. As stated in the introduction, when f does not have enough smoothness, we may wish to perform convolution smoothing. This is achieved by multiplying the transform $\hat{f}$ by some other transform $\hat{g}$ before doing the inversion (in a controlled way, so that we have a bound on the error introduced).

Remark 2. From the t in the denominator of the argument of $\mathrm{Re}(\hat{f})$ in (15), we may anticipate that the appropriate value of n increases with t, and this often is the case. According to Hosono,[22] the value of $n(t)$ to achieve prescribed accuracy is often approximately a linear function of t. Thus, on p. 58 of [22], Hosono suggests estimating this linear function by considering two time points, using the error estimate

$$E(m, n+1, t) - E(m, n, t) = \sum_{k=0}^{m} 2^{-m}\binom{m}{k} a_{n+k+1}(t) \quad (19)$$

before performing many runs at other time points (if indeed many time points are to be considered).

2. The Second Method: POST-WIDDER

Our second method, which we call POST-WIDDER, is based on the Post-Widder Theorem, which expresses $f(t)$ as the pointwise limit as $n \to \infty$ of

$$f_n(t) = \frac{(-1)^n}{n!}\left(\frac{n+1}{t}\right)^{n+1} \hat{f}^{(n)}((n+1)/t), \quad (20)$$

where $\hat{f}^{(n)}(s)$ is the nth derivative of the Laplace transform $\hat{f}$ at s; see p. 233 of Feller.[18] Feller shows that the Post-Widder formula is easy to understand probabilistically. By differentiating the transform, it is easy to see that $f_n(t) = E[f(X_{n,t})]$, where $X_{n,t}$ is a random variable with a gamma distribution on $(0,\infty)$ with mean t and variance $t/(n+1)$. Hence, $X_{n,t}$ converges in probability as $n \to \infty$ to the random variable X_t with $P(X_t = t) = 1$, so that $f_n(t) \to f(t)$ as $n \to \infty$ for all bounded real-valued f that are continuous at t (and other f as well).

Following Jagerman,[23, 24] we numerically calculate $f_n(t)$ via a generating function

$$G(z) \equiv \sum_{n=0}^{\infty} a_n(t) z^n = \frac{n+1}{t}\hat{f}\left(\frac{(n+1)}{t}(1-z)\right), \quad (21)$$

whose nth coefficient is $f_n(t)$, i.e., $a_n(t) = f_n(t)$. Using the Cauchy contour integral, we obtain

$$f_n(t) = \frac{1}{2\pi i}\int_{C_r} \frac{G(z)}{z^{n+1}}\,dz, \quad (22)$$

where C_r is a circle of radius r. After making the change of variables $z = re^{iu}$, we obtain the inversion integral

$$\begin{aligned} f_n(t) &= \frac{1}{2\pi r^n}\int_0^{2\pi} G(re^{iu})e^{-inu}\,du \\ &= \frac{n+1}{t}\frac{1}{2\pi r^n}\int_0^{2\pi}\hat{f}\left(\frac{(n+1)}{t}(1-re^{iu})\right)e^{-inu}\,du. \end{aligned} \quad (23)$$

Next, paralleling our treatment of (2), we apply the Fourier-series method (Poisson summation formula) to obtain the trapezoidal-rule approximation to (23) with an explicit error bound. Unlike (2), the integral in (23) is over a finite interval, so that the resulting sum is finite (and manageable), so that no truncation is necessary.

Starting from (23), for any n we can apply the discrete Poisson summation formula to obtain the trapezoidal rule approximation with step size π/n and the associated error bound, i.e.,

$$\begin{aligned} f_n(t) &= \frac{n+1}{2tnr^n}\sum_{k=1}^{2n}(-1)^k \mathrm{Re}(\hat{f})\left(\frac{n+1}{t}(1-re^{\pi ik/n})\right) - e_d \\ &= \frac{n+1}{2tnr^n}\Bigg\{\hat{f}((n+1)(1-r)/t) \\ &\quad + (-1)^n\hat{f}((n+1)(1+r)/t) \\ &\quad + 2\sum_{k=1}^{n-1}(-1)^k \mathrm{Re}(\hat{f})\left(\frac{n+1}{t}(1-re^{\pi ik/n})\right)\Bigg\} - e_d, \end{aligned} \quad (24)$$

where

$$e_d = \sum_{j=1}^{\infty} f_{n+jm}\left(t + \frac{tj2m}{n+1}\right) r^{2jn}. \quad (25)$$

Assuming that $|f(t)| \leqslant 1$ for all t, we have $|f_n(t)| \leqslant 1$ for all n and t, so that

$$|e_d| \leqslant \frac{r^{2n}}{1-r^{2n}} \approx r^{2n}. \quad (26)$$

In particular, given that

$$\phi(u) = \sum_{k=-\infty}^{\infty} a_k e^{iku} \quad \text{and} \quad a_n = \frac{1}{2\pi}\int_0^{2\pi}\phi(u)e^{-inu}\,du, \tag{27}$$

as occurs here with $a_n = f_n(t)r^n$ and $\phi(u) = G(re^{iu})$, we can form the periodic sequence

$$a_k^p = \sum_{j=-\infty}^{\infty} a_{k+jm} \tag{28}$$

with period m, paralleling (6). If $|f(t)| \leq 1$ for all t, then $|f_n(t)| \leq 1$ for all t and n, so that $\sum_{k=-\infty}^{\infty}|a_k| < \infty$. Next, paralleling (7)–(9), construct the discrete Fourier transform of $\{a_k^p\}$, see p. 51 of Rabiner and Gold,[31] to obtain

$$\begin{aligned}\hat{a}_k^p &= \frac{1}{m}\sum_{j=0}^{m-1} a_j^p e^{i2\pi kj/m} \\ &= \frac{1}{m}\sum_{j=0}^{m-1}\sum_{l=-\infty}^{\infty} a_{j+lm}e^{i2\pi jk/m} \\ &= \frac{1}{m}\sum_{j=-\infty}^{\infty} a_j e^{i2\pi jk/m} = \frac{1}{m}\phi(2\pi k/m)\end{aligned} \tag{29}$$

and applying the inversion formula for discrete Fourier transforms to obtain

$$\begin{aligned}a_k^p &= \sum_{j=0}^{m-1}\hat{a}_j^p e^{-i2\pi jk/m} \\ &= \frac{1}{m}\sum_{j=0}^{m-1}\phi(2\pi j/m)e^{-i2\pi jk/m}.\end{aligned} \tag{30}$$

Combining (28) and (30) yields the discrete Poisson summation formula

$$\sum_{k=1}^{m} g(2\pi k/m)e^{-ik2\pi n/m} = m\sum_{k=-\infty}^{\infty} a_{n+km}, \tag{31}$$

which with (23) implies that

$$f_n(t) = \frac{n+1}{tmr^n}\sum_{k=1}^{m} f\left(\frac{n+1}{t}(1 - re^{ikh})\right)e^{-inkh} - e_d, \tag{32}$$

with $h = 2\pi/m$ and

$$e_d = \sum_{j=1}^{\infty} f_{n+jm}\left(t + \frac{tjm}{n+1}\right)r^{jm}. \tag{33}$$

Letting $m = 2n$ in (32) yields (24).

We can make the error in the calculation of $f_n(t)$ by (24) suitably small by a proper choice of r; i.e., to obtain $10^{-\gamma}$ accuracy, we let $r = 10^{-\gamma/2}$. However, roundoff problems increase as r decreases. Roughly speaking $(3\gamma/2)$-digit accuracy is required to produce accuracy to $10^{-\gamma}$; see Remark 5.8 of [4].

The gap in POST-WIDDER is that we have indicated how to calculate the approximating function $f_n(t)$ in (20) instead of $f(t)$ itself. Moreover, $f_n(t)$ is known to converge to $f(t)$ quite slowly as $n \to \infty$ (of order n^{-1}) and the computation gets more difficult as n increases. In order to enhance the accuracy, we use a linear combination of the terms, i.e.,

$$\tilde{f}_{j,m}(t) = \sum_{k=1}^{m} w(k,m)f_{jk}(t), \tag{34}$$

e.g., with $j = 10$ and $m = 6$. In particular, we suggest using the linear combination developed by Stehfest.[33]

From Jagerman [23, 24], it follows that the error in (20) has the asymptotic form

$$f_n(t) - f(t) \sim \sum_{j=1}^{\infty} c_j(t)n^{-j}. \tag{35}$$

Hence it is natural to use (34) where the weights $w(k,m)$ are chosen to cancel the leading coefficients $c_j(t)$ in (35). In fact, Jagerman does this in [23, 24] for the case $m = 2$. General weights for knocking out all these coefficients were found by Stehfest[33]; they are

$$w(k,m) = (-1)^{m-k}\frac{k^m}{k!(m-k)!}, \tag{36}$$

as can be seen from the combinatorial identity

$$\sum_{k=1}^{m}(-1)^{m-k}\binom{m}{k}\frac{k^j}{m!} = \begin{cases}0, & j = 1,2,\ldots,m-1\\ 1, & j = 0 \text{ and } m;\end{cases} \tag{37}$$

see (12.7) and (12.17) on pp. 64–65 of Feller.[17] (Also see p. 35 of Wimp.[34]) Hence, our final approximation is (34). We start with $j = 10$ and $m = 6$ and increase them if necessary. We found that the Stehfest enhancement in (34) provides substantially greater accuracy than Jagerman's[24] σ enhancement (for examples such as those in [4]).

3. Implementation

The two methods presented here have the virtue that they are easy to understand and easy to perform. Programs implementing the algorithms can be written in less than fifty lines, as illustrated in Exhibits I and II.

The sample programs are written in UBASIC, which is a public-domain high-precision version of BASIC created by Kida[26] to do mathematics on a personal computer; see Neumann.[29] UBASIC permits complex numbers to be specified conveniently and it represents numbers and performs computations with up to 100-decimal-place accuracy. (Diskettes containing UBASIC and the algorithms are available from the authors.) However, other languages such as FORTRAN and C are also fine, but they should be used with double precision.

The displayed sample programs compute a function $F^c(t)$ (depending on two parameters Rho and Mean) whose Laplace transform is

$$\hat{F}^c(s) \equiv \int_0^{\infty} e^{-st}F^c(t)\,dt = \frac{1 - \hat{g}_e(s)}{s(1 - \rho\hat{g}_e(s))}, \tag{38}$$

where

$$\hat{g}_e(s) = \frac{1 - g(s)}{(\text{Mean} * s)} \tag{39}$$

and

$$\hat{g}(s) = \int_0^\infty e^{-st} g(t)\,dt = \frac{1}{\sqrt{1+2s}}. \tag{40}$$

The function $\hat{F}^c(s)$ is specified in lines 92–95 of the two programs. Different problems can be solved by inserting different transforms here.

The function $F^c(t)$ whose transform is given in (38) is the complementary cdf of the waiting time in an M/G/1 queue, after eliminating the known atom at the origin, when the service-time distribution is gamma with mean 1

Exhibit I. The UBASIC Program for EULER

```
1   'The Algorithm EULER
2   '
3   'A variant of the Fourier-series method
4   'using Euler summation
5   'applied to the M/G/1 transform (39)
6   '
10  dim SU(13), C(12)
11  C(1) = 1:C(2) = 11:C(3) = 55:C(4) = 165:C(5) = 330:
      C(6) = 462
12  C(12) = 1:C(11) = 11:C(10) = 55:C(9) = 165:
      C(8) = 330:C(7) = 462
13  '
20  input "TIME = "; T
21  A = 18.4
22  Ntr = 15
23  U = exp(A/2)/T
24  X = A/(2*T)
25  H = #pi/T
26  '
30  Sum = fnRf(X, 0)/2
31    for N = 1 to Ntr: Y = N*H
32    Sum += (-1)^N*fnRf(X, Y):next
33  '
40  SU(1) = Sum
41    for K = 1 to 12:N = Ntr + K:Y = N*H
42    SU(K + 1) = SU(K) + (-1)^N*fnRf(X, Y):next
43  '
50  Avgsu = 0:Avgsu1 = 0
51    for J = 1 to 12
52    Avgsu += C(J)*SU(J)
53    Avgsu1 += C(J)*SU(J + 1):next
54  Fun = U*Avgsu/2048:Fun1 = U*Avgsu1/2048
55  '
60  Errt = abs(Fun - Fun1)/2
61  '
70  print
71  print "TIME = ";T,"FUNCTION = ";using(2,7),Fun1
72  print
73  print "Truncation Error Estimate = ";using(1,7),Errt
74  end
75  '
90  fnRf(X, Y)
91  S = X + #i*Y
92  Rho = 0.75:Mean = 1
93  Gs = 1/sqrt(1 + 2*S)
94  Gse = (1 - Gs)/(Mean*S)
95  Fs = (1 - Gse)/(S*(1 - Rho*Gse))
96  Rfs = re(Fs)
97  return(Rfs)
```

Exhibit II. The UBASIC Program for POST-WIDDER

```
1   'The Algorithm POST-WIDDER
2   '
3   'The Jagerman-Stehfest method
4   'based on the Post-Widder inversion formula
5   'for inverting Laplace transforms
6   'applied to the M/G/1 transform (39)
7   '
10  NN = 6
11  dimG(NN)
12  '
20  input "TIME = "; T
21  '
30  for I = 1 to NN
31  N = 10*I
32  E = 8
33  R = 1/10^(E/(2*N))
34  U = (N + 1)/(T*2*N*R^N)
35  H = #pi/N
36  '
40  Sum = 0
41    for J = 1 to N - 1
42    S = (N + 1)*(1 - R*exp(#i*H*J))/T
43    Sum += (-1)^J*fnF(S):next J
44  Sum = fnF((N + 1)*(1 - R)/T) + 2*Sum
      +(-1)^N*fnF((N + 1)*(1 + R)/T)
45  '
46  G(I) = U*Sum:next I
47  '
50  Fun = 0
51    for K = 1 to NN
52    Wt = (-1)^(NN - K)*K^NN/(!(K)*!(NN - K))
53    Fun += Wt*G(K):next
54  '
60  print
61  print "TIME = ";T,"FUNCTION = ";using(2,7),Fun
62  end
63  '
91  fnF(S)
92  Rho = 0.75:Mean = 1
93  Gs = 1/sqrt(1 + 2*S)
94  Gse = (1 - Gs)/(Mean*S)
95  Fs = (1 - Gse)/(S*(1 - Rho*Gse))
96  return(re(Fs))
```

and shape parameter 0.50 and the traffic intensity (which equals the arrival rate) is $\rho = 0.75$, as in the second part of Section 9 in [4]. In particular, the service-time distribution has probability density

$$g(t) = \frac{e^{-t/2}}{\sqrt{2\pi t}}, \quad t \geq 0, \tag{41}$$

and Laplace transform (40). The programs compute the complementary cdf $F^c(t) = 1 - F(t)$, where $W(t)$ is the steady-state waiting-time cdf and $W(t) = 1 - \rho + \rho F(t)$. Thus, $F(t)$ is the cdf of the conditional steady-state waiting time given that a customer is delayed. We remove the known atom of $(1 - \rho)$ at the origin in order to obtain a smoother function. (This is more important when Euler summation is not used, i.e., when the series in (5) is simply truncated.) By virtue of the Pollaczek-Khintchine formula, the Laplace transform of $F^c(t)$ is (38).

For the program EULER, the computation is (15). There are three parameters: the discretization-error parameter A in (11)–(14) and the parameters m and n in the Euler sum (15).

The parameter A has been set at 18.4 in line 21 of the program. As is indicated after (12), this produces a discretization error of about 10^{-8}. This is appropriate to get accuracy to 10^{-7} (having the seventh decimal correct). In order to have a discretization error of $10^{-\gamma}$, we let $A = \gamma \log 10$, but much higher values of A (much smaller discretization errors) can cause computational difficulties (e.g., roundoff error).

We have specified the binomial coefficients associated with Euler summation in lines 10–12. We have chosen $m = 11$, so that there are 12 binomial coefficients. It should not be necessary to change m. In line 22 we have set $n = 15$. This makes the computation a sum of 27 terms. It may be necessary to increase n, and it is easy to do so. (Considering different parameter triples (A, m, n) provides an additional check.)

For the program POST-WIDDER, the computation is in (34) and (24). There are three parameters: the discretization-error parameter as specified below (33), and the Stehfest weight parameters j and m in (34). We have made the discretization error about 10^{-8} by setting $\gamma = E = 8$ in line 32 of the program. The associated parameter r in (33) is then $10^{-\gamma/2}$. As with EULER, computational problems can occur if we try to make γ too big (r too small). We have specified the Stehfest weights as $m = NN = 6$ in line 10 and $j = 10$ by setting $N = 10 * I$ in line 31. (Computational problems also can occur if we try to make m too large.)

With the parameter settings described here, there should be no serious roundoff errors with double precision, but double precision is recommended.

4. A Connection Between the Two Methods

From the Post-Widder theorem, we know that $f_n(t)$ in (20) approaches $f(t)$ as $n \to \infty$. Hence, it is natural to consider how the sum in (24) is related to the sum in (5) as $n \to \infty$. If we let $r_n = 1 - (A/2n)$, so that $r^n \to e^{-A/2}$ as $n \to \infty$, then the sum in (24) approaches the sum in (5) term by term as $n \to \infty$. (The two procedures still serve as valid checks on each other, because we do not consider large n.) In particular, the sum in (24) becomes

$$\frac{n+1}{2nr^n t} \sum_{j=-(n-1)}^{n} (-1)^j \mathrm{Re}\left(\hat{f}\left(\frac{n+1}{t}(1 - re^{i\pi j/n})\right)\right), \tag{42}$$

after replacing j by $j - 2n$ for $n + 1 \leq j \leq 2n$. First, note that

$$\frac{n+1}{2nr^n t} \to \frac{e^{A/2}}{2t} \quad \text{as} \quad n \to \infty \tag{43}$$

and

$$\begin{aligned} \frac{n+1}{t}(1 - re^{i\pi j/n}) &= \frac{n+1}{t}\left(1 - r\cos\frac{\pi j}{n} - ri\sin\frac{\pi j}{n}\right) \\ &\to \frac{A}{2t} + \frac{i\pi j}{t} \quad \text{as} \quad n \to \infty, \end{aligned} \tag{44}$$

because $r_n = 1 - (A/2n)$, $\cos(\pi j/n) \to 1$ as $n \to \infty$ and $\sin(\pi j/n) \sim \pi j/n$ as $n \to \infty$. The limits in (43) and (44) imply that

$$\mathrm{Re}\left(\hat{f}\left(\frac{n+1}{t}(1 - re^{i\pi j/n})\right)\right) \to \mathrm{Re}\left(\hat{f}\left(\frac{A}{2t} + \frac{i\pi j}{t}\right)\right) \quad \text{as} \quad n \to \infty. \tag{45}$$

Hence (42) approaches

$$\frac{e^{A/2}}{2t} \sum_{j=-\infty}^{\infty} (-1)^j \mathrm{Re}\left(\hat{f}\left(\frac{A}{2t} + \frac{i\pi j}{t}\right)\right), \tag{46}$$

which coincides with (5).

5. Conclusions

We have presented two methods for numerically inverting Laplace transforms of cumulative distribution functions or complementary cumulative distribution functions. The probabilistic structure (i.e., the fact that $|f(t)| \leq 1$) enables us to obtain simple effective bounds on the discretization error associated with the trapezoidal-rule approximation of the inversion integrals, but in each case there is another step for which we have no bounds. In both cases we can estimate the final errors by considering the improvement obtained upon successive refinement. However, we rely on the agreement of the two methods to confirm accuracy. The resulting programs and run times are very short, so the procedure is easily carried out. For further discussion, see [4].

Acknowledgments

We thank Dimitris Bertsimas, Bharat Doshi, Kerry Fendick, Daniel Heyman, Toshio Hosono, David Jagerman, John Morrison, and Henk Tijms for assistance. In particular, we thank John Morrison for his contribution to Remark 1 and Kerry Fendick for writing C + + versions of the programs.

References

1. J. Abate, G.L. Choudhury, and W. Whitt, 1993. Calculation of the GI/G/1 Waiting-Time Distribution and its Cumulants from Pollaczek's Formulas, *Archiv für Elektronik und Übertragungstechnik 47*, 311–321.

2. J. ABATE, G.L. CHOUDHURY, and W. WHITT, 1994. Asymptotics for Steady-State Tail Probabilities in Structured Markov Queueing Models, *Stochastic Models, 10*, 99–143.
3. J. ABATE, G.L. CHOUDHURY, and W. WHITT, 1995. Exponential Approximations for Tail Probabilities in Queues, I: Waiting Times, *Operations Research*, to appear.
4. J. ABATE and W. WHITT, 1992. The Fourier-Series Method for Inverting Transforms of Probability Distributions, *Queueing Systems 10*, 5–88.
5. J. ABATE and W. WHITT, 1992. Numerical Inversion of Probability Generating Functions, *Operations Research Letters 12*, 245–251.
6. J. ABATE and W. WHITT, 1992. Solving Probability Transform Functional Equations for Numerical Inversion, *Operations Research Letters 12*, 275–281.
7. D. BERTSIMAS and D. NAKAZATA, 1992. Transient and Busy Period Analysis for the GI/G/1 Queue; the Method of Stages, *Queueing Systems 10*, 153–184.
8. G.L. CHOUDHURY and D.M. LUCANTONI, 1995. Numerical Computation of the Moments of a Probability Distribution from its Transforms, *Operations Research*, to appear.
9. G.L. CHOUDHURY, D.M. LUCANTONI and W. WHITT, 1994. Multi-Dimensional Transform Inversion with Applications to the Transient M/G/1 Queue, *Annals of Applied Probability 4*, 719–740.
10. G.L. CHOUDHURY, D.M. LUCANTONI, and W. WHITT, 1995. Numerical Solution of $M_t/G_t/1$ Queues, *Operations Research*, to appear.
11. G.L. CHOUDHURY, D.M. LUCANTONI, and W. WHITT, 1995. Squeezing the Most Out of ATM. *IEEE Transactions on Communications*, to appear.
12. G. DAHLQUIST, 1993. A "Multigrid" Extension of the FFT for the Numerical Inversion of Fourier and Laplace Transforms, *BIT 33*, 85–112.
13. B. DAVIES and B.L. MARTIN, 1970. Numerical Inversion of Laplace Transforms: A Critical Evaluation and Review of Methods, *Journal of Computational Physics 33*, 1–32.
14. G. DOETSCH, 1974. *Introduction to the Theory and Application of the Laplace Transformation*, Springer-Verlag, New York.
15. H. DUBNER and J. ABATE, 1968. Numerical Inversion of Laplace Transforms by Relating Them to the Finite Fourier Cosine Transform, *JACM 15*, 115–123.
16. P. EMBRECHTS, R. GRÜBEL, and S.M. PITTS, 1993. Some Applications of the Fast Fourier Transform Algorithm in Insurance Mathematics, *Statistica Neerlandica 47*, 59–75.
17. W. FELLER, 1968. *An Introduction to Probability Theory and its Applications, Vol. I*, 3rd ed., Wiley, New York.
18. W. FELLER, 1971. *An Introduction to Probability Theory and its Applications, Vol. II*, 2nd ed., Wiley, New York.
19. R. GRÜBEL, 1991. Algorithm AS265: G/G/1 Via Fast Fourier Transform, *Applied Statistics 40*, 355–365.
20. T. HOSONO, 1981. Numerical Inversion of Laplace Transform, *J. Inst. Elec. Eng. Japan* A54–A64, 494 (in Japanese).
21. T. HOSONO, 1981. Numerical Inversion of Laplace Transform and Some Applications to Wave Optics, *Radio Science 16*, 1015–1019.
22. T. HOSONO, 1984. *Fast Inversion of Laplace Transform by BASIC*, Kyoritsu Publishers, Japan (in Japanese).
23. D.L. JAGERMAN, 1978. An Inversion Technique for the Laplace Transform With Applications, *Bell System Tech. J. 57*, 669–710.
24. D.L. JAGERMAN, 1982. An Inversion Technique for the Laplace Transform, *Bell System Tech. J. 61*, 1995–2002.
25. R. JOHNSONBAUGH, 1979. Summing an Alternating Series, *Amer. Math. Monthly 86*, 637–648.
26. Y. KIDA, 1990. UBASIC Version 8.12, Faculty of Science, Kanazawa University, 1-1 Marunouchi, Kanazawa 920, Japan.
27. D.M. LUCANTONI, 1991. New Results on the Single Server Queue with a Batch Markovian Arrival Process, *Stochastic Models 7*, 1–46.
28. D.M. LUCANTONI, G.L. CHOUDHURY, and W. WHITT, 1994. The Transient BMAP/G/1 Queue, *Stochastic Models, 10*, 145–182.
29. W.D. NEUMANN, 1989. UBASIC: A Public-Domain BASIC for Mathematics, *Notices Amer. Math. Soc. 36*, 557–559.
30. L.K. PLATZMAN, J.C. AMMONS, and J.J. BARTHOLDI, 1988. A Simple and Efficient Algorithm to Compute Tail Probabilities from Transforms, *Operations Research 36*, 137–144.
31. L.R. RABINER and B. GOLD, 1975. *Theory and Application of Digital Signal Processing*, Prentice-Hall, Englewood Cliffs, NJ.
32. R.M. SIMON, M.T. STROOT, and G.H. WEISS, 1972. Numerical Inversion of Laplace Transforms with Application to Percentage labeled Experiments, *Comput. Biomed. Res. 6*, 596–607.
33. H. STEHFEST, 1970. Algorithm 368. Numerical Inversion of Laplace Transforms, *Comm. ACM 13*, 479–490 (erratum 13, 624).
34. J. WIMP, 1981. *Sequence Transformations and Their Applications*, Academic Press, New York.