THE FOURIER-SERIES METHOD FOR INVERTING TRANSFORMS OF PROBABILITY DISTRIBUTIONS

by

Joseph Abate

900 Hammond Road
Ridgewood, NJ 07450-2908

Ward Whitt

AT&T Bell Laboratories
Room 2C-178
Murray Hill, NJ 07974-0636

February 25, 1991

Revision: September 19, 1991

Short title: Inverting Transforms

THE FOURIER-SERIES METHOD FOR INVERTING TRANSFORMS OF PROBABILITY DISTRIBUTIONS

by

Joseph Abate

900 Hammond Road
Ridgewood, NJ 07450-2908

Ward Whitt

AT&T Bell Laboratories
Room 2C-178
Murray Hill, NJ 07974-0636

February 25, 1991

Revision: September 19, 1991

To appear in *Queueing Systems*

ABSTRACT

This paper reviews the Fourier-series method for calculating cumulative distribution functions (cdf's) and probability mass functions (pmf's) by numerically inverting characteristic functions, Laplace transforms and generating functions. Some variants of the Fourier-series method are remarkably easy to use, requiring programs of less than fifty lines. The Fourier-series method can be interpreted as numerically integrating a standard inversion integral by means of the trapezoidal rule. The same formula is obtained by using the Fourier series of an associated periodic function constructed by aliasing; this explains the name of the method. This Fourier analysis applies to the inversion problem because the Fourier coefficients are just values of the transform. The mathematical centerpiece of the Fourier-series method is the Poisson summation formula, which identifies the discretization error associated with the trapezoidal rule and thus helps bound it. The greatest difficulty is approximately calculating the infinite series obtained from the inversion integral. Within this framework, lattice cdf's can be calculated from generating functions by finite sums without truncation. For other cdf's, an appropriate truncation of the infinite series can be determined from the transform based on estimates or bounds. For Laplace transforms, the numerical integration can be made to produce a nearly alternating series, so that the convergence can be accelerated by techniques such as Euler summation. Alternatively, the cdf can be perturbed slightly by convolution smoothing or windowing to produce a truncation error bound independent of the original cdf. Although error bounds can be determined, an effective approach is to use two different methods without elaborate error analysis. For this purpose, we also describe two methods for inverting Laplace transforms based on the Post-Widder inversion formula. The overall procedure is illustrated by several queueing examples.

Key Words: computational probability; numerical inversion of transforms; characteristic functions; Laplace transforms; generating functions; Fourier transforms; cumulative distribution functions; calculating tail probabilities; numerical integration; Fourier series; Poisson summation formula; the Fourier-series method; the Gaver-Stehfest method.

1. Introduction

Probability distributions of interest in queueing models can often be determined in the form of transforms. It is widely recognized that the transforms are useful to do asymptotic analysis, e.g., to calculate moments and to determine the asymptotic behavior of tail probabilities. However, the transforms are also useful for calculating numerical values of the cumulative distribution functions (cdf's) and probability mass functions (pmf's) by numerical inversion.

Numerical inversion is most important when a transform cannot be analytically inverted by manipulating tabled formulas of special cases (as in Oberhettinger [94]), but numerical inversion may even be convenient when a direct expression for the cdf is available. To illustrate, consider the familiar example of the cdf, say $W(t)$, of the steady-state waiting time before beginning service in the M/G/1 queue. The celebrated Pollaczek-Khintchine formula expresses the Laplace-Stieltjes transform (LST) of this cdf as

$$\hat{W}(s) = \int_0^\infty e^{-st}\, dW(t) = \frac{1-\rho}{1-\rho[1-\hat{G}(s)]/\tau s}\ , \tag{1.1}$$

where ρ is the traffic intensity and $\hat{G}(s)$ is the LST of the service-time cdf $G(t)$ with mean τ; see p. 200 of Kleinrock [75]. Since $[1-\hat{G}(s)]/\tau s$ is the LST of $G_e(t) = \tau^{-1} \int_0^t [1-G(y)]\, dy$, the stationary-excess cdf associated with $G(t)$, the transform $\hat{W}(s)$ is easily inverted analytically (using $(1-y)^{-1} = \sum_{n=0}^{\infty} y^n$), yielding

$$W(t) = (1-\rho) \sum_{n=0}^{\infty} \rho^n\, G_e^{n*}(t)\ ,\quad t \geq 0\ , \tag{1.2}$$

where $G_e^{n*}(t)$ is the n-fold convolution of the cdf $G_e(t)$, with $G_e^{0*}(t) = 1$, $t \geq 0$. However, in general it is not easy to evaluate (1.2), so that numerical inversion of (1.1) remains a viable alternative for obtaining numbers.

Despite the many transform results for probability distributions describing queues and other applied probability models, and despite the recent interest in computational probability, there is surprisingly little discussion of numerical transform inversion in the queueing and applied probability literature. The only applied-probability-textbook discussion we are aware of is pp. 73-74 of Kobayashi [76], which is a brief account of the Fourier-series method to be reviewed here.

Of course, transforms are being inverted numerically, but numerical inversion seems to be considered difficult. Even in the last five years, strong statements have been made about the difficulty of transform inversion. For example, P. G. Harrison [54] states:

> ‘‘Whilst being of value in itself, for example as a source of the moments of the required distribution, the Laplace transform is inadequate for problems which require estimates for related probabilities.’’

Kwok and Barthez [79] write that

> ‘‘The inversion of the Laplace transform is well known to be an ill-conditioned problem. Numerical inversion is an unstable process and the difficulties often show up as being highly sensitive to round-off errors.’’

Platzman, Ammons and Bartholdi [99] write that

> ‘‘The standard inversion formula is a contour integral, not a calculable expression.’’

They also state that

> ‘‘These methods provide convergent sequences rather than formal algorithms; they are difficult to implement (many involve solving large, ill-conditioned systems of linear equations or analytically obtaining high-order derivatives of the transform) and none includes explicit, numerically computable bounds on error and computational effort;’’

see p. 137 of [99].

While there are grains of truth in some of these remarks, they seriously misrepresent the true state of affairs. The remarks seem to be based on extensive experience showing that a poor method of numerical inversion can lead to difficulties. For example, the method of Bellman, Kalaba and Lockett [10], which is based on approximately converting the inversion problem to that of solving a system of linear algebraic equations, is notorious for numerical difficulties (at least by conventional implementations); e.g., see Gautschi [48] and Varah [121].

We contend that numerical transform inversion can be remarkably easy to understand and perform (e.g., on a small computer using a program of less than 50 lines.) To demonstrate how easy the numerical inversion can be, we display a BASIC program that implements a variant of the Fourier-series method called EULER to calculate the M/G/1 waiting-time cdf $W(t)$ by inverting (1.1). Actually, the program is written in the public-domain high-precision UBASIC by Kida [74], which is convenient for doing numerical mathematics on a personal computer with an Intel 86 chip; see Neumann [92].

Insert Algorithm EULER here (or slightly later)

UBASIC permits complex numbers to be specified conveniently and it represents numbers and performs computations with up to 100-decimal-place accuracy. However, ordinary BASIC, FORTRAN or C with double precision would suffice. Indeed, we also have versions of our algorithms in $C++$, which also permits complex numbers to be specified conveniently. For a detailed explanation of the algorithm EULER and further discussion, see §7 and Example 9.2. To quickly see several variants of the Fourier-series method, go directly to §7. Brief accounts of the Fourier-series method for numerically inverting Laplace transforms and generating functions of cdf's are also contained in Abate and Whitt [6], [7].

In this paper we do five things: (1) explain the basic ideas behind the Fourier-series method, (2) present a few specific variants of the Fourier-series method, one of which is the algorithm EULER, (3) review the literature related to the Fourier-series method, (4) present some different alternative numerical inversion methods to serve as checks and (5) illustrate numerical inversion applied to several queueing examples.

One might expect that one or more of these activities would be unnecessary because of the existing literature, but that does not seem to be the case. Unfortunately, the literature on numerical transform inversion is scattered and confusing. There tends to be a separate literature associated with each kind of transform and there is a bewildering multitude of methods; e.g., see the book by Krylov and Skoblya [78] and the bibliography by Piessens [96] and Piessens and Dang [97]. In addition, many different kinds of functions are considered, so that complications not associated with probability distributions arise. *In this paper we primarily restrict attention to the Fourier-series method applied to functions that are cdf's of probability distributions on the real line.* (In probability applications it is typically known that the transform is a transform of a probability distribution, so that there is nothing extra to verify.) The Fourier-series method is not restricted to this class of functions, but this restriction helps in the error analysis. It is also is a convenient simplification, implying that Laplace transforms and Fourier transforms can be regarded as *characteristic functions*, as discussed in Ch. XV of Feller [43], Ch. 6 of Chung [21] and Lukacs [80]. As much as possible, we try to exploit probabilistic structure, so that the method will be easy for probabilists to understand.

The algorithm EULER above for inverting Laplace transforms was developed in 1968-1972 by Dubner and Abate [40] and Simon, Stroot and Weiss [110]. We use the name EULER primarily because the algorithm exploits Euler summation, but also because Laplace himself traced the beginnings of the Laplace transform back to Euler; see Deakin [33]. An essentially equivalent algorithm was developed in Japan in 1989 by Hosono [58], [59], but the derivation was quite different. In Japan Hosono popularized this method, which he calls the Fast Inversion of Laplace Transforms (FILT), by his book [60] in 1984 and subsequent papers [61] and [62]. (We learned about the early references [58] and [59] from Dimitris Bertsimas, who successfully applies the method to queueing problems in Bertsimas and Nakazato [12]. We learned about [60]-[62] from T. Hosono after we sent him a draft of our paper.)

Variants of the Fourier-series method for numerically inverting characteristic functions were developed in 1960–1975 by Bohman [13], [14], [15], Davies [30] and Schorr [108]. The

Fourier-series method was previously applied to Fourier integrals, Laplace transforms and other integrals by Koizumi [77], Fettis [44], De Balbine and Franklin [34]. However, new papers on the Fourier-series method keep appearing, typically without referring to the relevant literature, without indicating that the method is a Fourier-series method, and without identifying the basic ideas behind the method (the trapezoidal rule and the Poisson summation formula); see the literature review in §15. Of course, there also are other good inversion methods, but we primarily restrict attention to the Fourier-series method. Our purpose is to expose the basic ideas of one method, and thereby illustrate that numerical transform inversion, by any of several good methods, is not difficult to understand and perform.

With the Fourier-series method, it is possible to do a careful error analysis, which is what much of this paper is about. For nice problems, the required computation to find an approximation with prescribed accuracy is not great, even for a small computer. For example, we give a convenient algorithm with a simple error bound for numerically inverting generating functions of lattice distributions in §5. However, for hard problems, the required computation to find an approximation with prescribed accuracy can be prohibitive. This difficulty is primarily (but not completely) due to the available error bounds not being tight; i.e., the actual error associated with a given computation is often much less than the error bounds indicate. However, there is a genuine difficulty if the cdf is not lattice but the cdf or its derivative has jumps, because the approximating cdf is always a trigonometric polynomial, and thus is continuously differentiable. One way to cope with this difficulty is to perturb the original function slightly in a controlled manner (to assure that you are satisfied with values of the perturbed function), so that the perturbed function is continuous (and continuously differentiable, if desired) and the required computation to achieve prescribed accuracy becomes manageable; see §6. Convolution smoothing also can improve the quality of the computation for smooth cdf's, as we illustrate in the examples.

It is often convenient (and sometimes necessary) to perform the computation without being absolutely certain that the desired accuracy will be (or has been) achieved. However, it is usually possible to *estimate* the error, and thus the required computation, using asymptotic analysis. It is also standard to perform successive refined computations until only negligible improvement is seen. Of course, with these last two techniques there is *no guarantee* that desired accuracy has been achieved. Doing *both* estimates based on asymptotic analysis and successive computations with refinements is obviously much safer than either one alone. Finally, using two very different methods is almost a guarantee.

Indeed, following Davies and Martin [29], we strongly recommend using two different methods. While much of the paper is devoted to the error analysis that is possible with the Fourier-series method, we propose systematically using two very different procedures, without complete error analysis. (However, two different methods are not needed for generating functions; see algorithm LATTICE-POISSON in §5.) Assuming that the two procedures agree to the prescribed accuracy, we can safely stop. If the two procedures do not agree, then we can try other procedures or apply convolution smoothing. This approach also has the advantage of helping to catch other errors besides shortcomings in the algorithms for difficult functions. Using two different procedures may seem obvious, but we are not aware of any paper where this approach was followed.

To provide specific alternatives for inverting Laplace transforms that are very different from the Fourier-series method, we describe two methods related to the Post [101]–Widder [125] inversion formula ((8.1) below), which is based on differentiation instead of integration. Our first alternative method for inverting Laplace transforms is the Gaver [49]–Stehfest [113] method. Unlike the Fourier-series method, the Gaver-Stehfest method illustrates some of the numerical difficulties referred to by Kwok and Barthez [79] and Platzman, Ammons and Bartholdi [99] above, because it often requires high numerical precision (beyond the standard double precision).

Our second alternative method is an enhancement of an algorithm of Jagerman [67]. In particular, we apply the Stehfest [113] acceleration procedure to obtain greater accuracy. Although it is not advertised as such, the Jagerman [67] algorithm is in fact also a Fourier-series method, because it employs the Poisson summation formula and leads to a trigonometric polynomial. However, since it starts with the Post-Widder formula, the overall procedure is quite different from our other Fourier-series methods. Hence, it serves as a genuine alternative to our other methods. Moreover, with the Stehfest acceleration, it produces good accuracy without requiring high precision. We also mention other candidate procedures in our literature review in §15.

Organization of the Paper

The rest of this paper is organized as follows. In § 2 we define several transforms and review their basic properties. In § 3 we present the basic inversion integrals. In § 4 we specify the proposed procedure for numerical integration, which is the trapezoidal rule. In § 5 we discuss the Poisson summation formula, which identifies the discretization error associated with the trapezoidal rule, justifies using the relatively primitive trapezoidal rule in this context, and explains why the procedure is called a Fourier-series method. The Poisson summation formula typically produces a complete solution to the inversion problem for generating functions of lattice probability distributions or bounded sequences of real numbers, because the associated inversion integral is over a finite interval, so that the Poisson summation formula produces a (manageable) finite sum. However, for non-lattice distributions we are left with an infinite series after applying the Poisson summation formula. We end §5 by briefly discussing the extension to higher dimensions, i.e., numerically inverting transforms of multivariate functions.

In § 6 we discuss the problem of approximately calculating the infinite series obtained from the trapezoidal rule (for nonlattice distributions). In addition to simple truncation, in §6 we discuss convolution smoothing (windowing) and alternative summation methods to accelerate convergence. The algorithm EULER displayed above is based on Euler summation of a nearly alternating series. In § 7 we briefly summarize a few variants of the Fourier-series method, one of which is EULER. In §8 we describe the two methods for inverting Laplace transforms related to the Post-Widder inversion formula.

In Sections 9-14 we discuss queueing examples. We begin in §9 by discussing the M/G/1 waiting-time cdf above. In §10 we discuss the time-dependent mean of reflecting Brownian motion (RBM), which is studied by Abate and Whitt [2] to gain insight into the transient behavior of queues. In §11 we illustrate the lattice function variants of the Fourier-series method by considering the number of customers served in an M/M/1 busy period. In §12 we discuss convolutions of many exponential distributions, which are of interest for calculating sojourn time distributions in open Markovian queueing networks; see Harrison [54]. In §13 we discuss renewal functions to illustrate that the numerical inversion procedures are not restricted to cdf's or probability density functions. In §14, to show the limitations of the numerical inversion methods, we discuss transforms of distributions such as a single point mass that are especially difficult to invert numerically. For most of the examples here, we have alternative exact expressions to use to evaluate the accuracy of the transform inversion.

Finally, in §15 we review the literature, giving special emphasis to the Fourier-series method.

2. The Transforms: Definitions and Basic Properties

Let X be a real-valued random variable with a cdf F, i.e., $F(t) = P(X \le t)$. The *characteristic function* (cf) of X (or F) is the complex-valued function of a real variable

$$\phi(u) \equiv \phi_X(u) \equiv E(e^{iuX}) \equiv \int_{-\infty}^{\infty} e^{iut} dF(t) \equiv \int_{-\infty}^{\infty} \cos ut \, dF(t) + i \int_{-\infty}^{\infty} \sin ut \, dF(t) \ , \quad (2.1)$$

where $i \equiv \sqrt{-1}$ is one of the imaginary roots of $x^2 + 1 = 0$ and $e^{ix} = \cos x + i \sin x$, e.g., see

Chapter 6 of Chung [21], Chapter 15 of Feller [43], Lukacs [80] and Oberhettinger [94].

We will use only elementary properties of the complex numbers. Recall that a complex number z can be expressed as $z = a + ib$ where a and b are real numbers. The complex number $z = a + ib$ has real part $Re\,(z) = a$, imaginary part $Im(z) = b$, conjugate $\overline{z} = a - ib$ and modulus $|z| = (a^2 + b^2)^{1/2}$, so that $Re\,(z) = (z + \overline{z})/2$ and $Im(z) = (z - \overline{z})/2i$. Since the cf is defined in terms of the exponential e^{iut}, we have the simple expressions

$$\overline{\phi}(u) = \phi(-u)\ , \quad Re\,(\phi)(u) = \frac{\phi(u) + \phi(-u)}{2} \quad \text{and} \quad Im(\phi)(u) = \frac{\phi(u) - \phi(-u)}{2i} \qquad (2.2)$$

for all u. We frequently use the fact that $Re\,(\phi)(u)$ and $\cos tu$ are even functions of u, while $Im\,(\phi)(u)$ and $\sin tu$ are odd functions of u; e.g., $Re\,(\phi)(-u) = Re\,(\phi)(u)$ and $Im\,(\phi)(-u) = -Im\,(\phi)(u)$. By the Cauchy-Schwarz inequality, p. 498 of Feller [43],

$$|\phi(u)| \leq 1 = \phi(0) \quad \text{for all } u\ . \qquad (2.3)$$

For applications, including the inversion formulas, a basic property is the expression for the cf of the sum of independent random variables (the convolution of cdf's); if X_1, and X_2 are independent, then

$$\phi_{X_1 + X_2}(u) = \phi_{X_1}(u)\phi_{X_2}(u) \text{ for all } u\ . \qquad (2.4)$$

Another key property is the argument switch between F and ϕ; the *tail behavior* of F (ϕ) is closely related to the *smoothness* of ϕ (F); see pp. 511-514 of [43]. Thus, in § 5 we use the tail of F to bound the error associated with a discretization of ϕ, and in § 6 we use the smoothness of F to bound the tail of ϕ. When F is not sufficiently smooth initially, we can exploit (2.4) and perform a convolution to obtain a smoother perturbed cdf that is sufficiently close to F.

If the cdf F has a density f (i.e., if F is absolutely continuous with respect to Lebesgue measure, so that $F(t) = \int_{-\infty}^{t} f(s)\,ds$ for all t), then of course

$$\phi(u) = \int_{-\infty}^{\infty} e^{iut} f(t)\, dt \,, \tag{2.5}$$

which is a minor modification of the Fourier transform of *f*; i.e., $\phi(-u)$ is a standard form of the *Fourier transform* of *f* and, in general, the *Fourier-Stieltjes transform* of *F*. If the cdf *F* concentrates on the integers, then the cf $\phi(u)$ coincides with the *generating function* or *z-transform*, after making the change of variables $z = e^{iu}$. Similarly, if *F* is concentrated on the positive real line, then $\phi(iu)$ is the *Laplace transform* of *f* and the *Laplace-Stieltjes transform* of *F* (where in general *u* is regarded as a complex variable).

Of course, these transforms are not restricted to probability distributions. For a function *f* on the positive real line (not necessarily a probability density), it is often convenient to work with the Laplace transform

$$\hat{f}(s) = \int_0^{\infty} e^{-st} f(t)\, dt \,, \tag{2.6}$$

where *s* is a complex variable. When we work with Laplace transforms of cdf's, we typically calculate the complementary cdf $F^c(t) \equiv 1 - F(t)$ by numerically integrating the Laplace transform of $F^c(t)$, i.e.,

$$\hat{F}^c(s) = \int_0^{\infty} e^{-st} F^c(t)\, dt \,. \tag{2.7}$$

For a sequence of real numbers $\{p_k : k \geq 0\}$ (not necessarily a probability mass function), it is often convenient to work with the generating function

$$G(z) = \sum_{k=0}^{\infty} p_k z^k \,. \tag{2.8}$$

In this paper we focus on transforms of functions on the real line, but there are corresponding multivariate transforms of functions on $\mathbb{R}^k$. The Fourier-series method can also be applied to invert these multivariate transforms; see §5.

3. Inversion Integrals

In this section we review the basic inversion integrals. We first consider bounded continuous densities, then real-valued functions on the positive real line, cdf's and probability mass functions of lattice distributions. For additional discussion, see Chung [21], Feller [43] and Lukacs [80].

Bounded Continuous Densities

The most familiar inversion formula expresses a density f at t in terms of its cf ϕ via an integral; see p. 509 of Feller [43] or p. 155 of Chung [21]. (This is a variant of the inversion formula for Fourier transforms; or course f need not be a density function.) If $|\phi|$ is integrable, then the cdf F is absolutely continuous with a bounded continuous density f and

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-itu} \phi(u)\, du \quad \text{for each } t . \tag{3.1}$$

Feller's proof of (3.1) using the Parseval relation is particularly revealing. Since $f(t)$ is real, we can write (3.1) directly as the integral of a real-valued function of a real variable, i.e.,

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} Re\, [e^{-itu}\, \phi(u)]\, du = \frac{1}{\pi} \int_{0}^{\infty} [\cos tu\ Re\, (\phi)(u) + \sin tu\ Im(\phi)(u)]\ du\ . \tag{3.2}$$

Real-Valued Functions on the Positive Real Line

The standard inversion integral for the Laplace transform $\hat{f}(s)$ in (2.6) is the Bromwich contour integral, which also 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, see pages 4 and 148 of Doetsch [36], we obtain

$$f(t) = \frac{1}{2\pi i} \int_{a - i\infty}^{a + i\infty} e^{st} \hat{f}(s)\, ds = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{(a+iu)t} \hat{f}(a + iu)\, du$$

$$= \frac{e^{at}}{2\pi} \int_{-\infty}^{\infty} (\cos ut + i \sin ut)\, \hat{f}(a + iu)\, du$$

$$= \frac{e^{at}}{2\pi} \int_{-\infty}^{\infty} [Re(\hat{f}(a + iu)) \cos ut - Im(\hat{f}(a + iu)) \sin ut]\, du$$

$$= \frac{2e^{at}}{\pi} \int_{0}^{\infty} Re(\hat{f}(a + iu)) \cos ut\, du \qquad (3.3)$$

$$= -\frac{2e^{at}}{\pi} \int_{0}^{\infty} Im(\hat{f})(a + iu) \sin ut\, du\ . \qquad (3.3a)$$

If f is a bounded continuous probability density, then it suffices to let $a = 0$, and (3.3) reduces to (3.2), but having $a > 0$ is convenient for controlling the error in the numerical integration; see (5.25)–(5.30) below. Interpreted differently, increasing a has the advantage of moving the contour further away from the singularities in the left halfplane.

To derive (3.3) from (3.2) in the case f is nonnegative (which is not required in general), let

$$g(t) = e^{-at} f(t)/2A \text{ for } t \geq 0 \quad \text{and} \quad g(t) = g(-t) \text{ for } t < 0\ ,$$

where

$$A = \int_{0}^{\infty} e^{-at} f(t)\ dt\ ,$$

so that g is an even function on the whole line. The parameter a is introduced via the damping by $e^{-a|t|}$. Since g is even, $\phi_g = Re(\phi_g)$. Since the conjugate of $\hat{f}(s)$ is $\hat{f}(\overline{s})$,

$$\phi_g(u) = Re(\phi_g)(u) = \frac{Re(\phi_f)(u + ai)}{A} = \frac{Re(\hat{f})(a - iu)}{A} = \frac{Re(\hat{f})(a + iu)}{A}\ .$$

By (3.2), assuming that $|\phi_g|$ is integrable,

$$\frac{e^{-at} f(t)}{2A} = g(t) = \frac{1}{2\pi A} \int_{-\infty}^{\infty} Re[e^{-tu} \phi_g(u)]\, du$$
$$= \frac{1}{2\pi A} \int_{-\infty}^{\infty} Re(\phi_g)(u) \cos tu\, du = \frac{1}{\pi A} \int_0^{\infty} Re(\hat{f})(a + iu) \cos tu\, du\ .$$

Similarly, we can derive (3.3a) from (3.2).

CDFs

As indicated on p. 511 of Feller [43], we can apply (2.4) and (3.1) to obtain corresponding inversion formulas for cdf's. Let f_x be the density of the convolution of F with the cdf of a uniform distribution on $[-x, x]$. Since $\sin (ux)/ux$ is the cf of a uniform distribution on $[-x, x]$,

$$f_x(t) = \frac{F(t + x) - F(t - x)}{2x} = \frac{1}{2\pi} \int_{-\infty}^{\infty} e^{-iut}\, \phi(u)\, \frac{\sin ux}{ux}\, du \tag{3.4}$$

provided that $|\phi(u)|/u$ is integrable.

An inversion formula directly for $F(t)$, due to Gil-Palaez [51], which does not seem to be so useful for numerical inversion, is

$$F(t) = \frac{1}{2} - \frac{1}{\pi} \int_0^{\infty} [\cos ut\, Im(\phi)(u) - \sin ut\, Re\, (\phi)(u)]\, \frac{du}{u}\ . \tag{3.5}$$

We obtain a further simplification if F concentrates on $[0, \infty)$ with $F(0) = 0$. By making the change of variables $t = 0$ and $x = t$ in (3.4), we obtain

$$F(t) = \frac{2}{\pi} \int_0^{\infty} Re\, (\phi)(u)\, \frac{\sin tu}{u}\, du\ . \tag{3.6}$$

For (3.4)-(3.6) to be valid, we require that F be continuous. Indeed, just as with (3.1), given that the modulus of the cf appearing in the integral on the right in (3.4) is integrable, the left side of (3.4) is continuous.

More general inversion formulas are given in § 6.2 of Chung [21]; see Exercise 10, p. 159, for the generalization of (3.5). For general cdf's, the integrand in (3.4) need not be absolutely integrable, but the integral always exists in a conditional sense, i.e., the total integral can be represented as the proper limit as $T \to \infty$ of the integrals over $[-T, T]$. In particular, *for any cdf* F and any time points $t_1 < t_2$,

$$F(t_2) - F(t_1) - \frac{[F(t_2) - F(t_{2^-})]}{2} + \frac{[F(t_1) - F(t_{1^-})]}{2} = \lim_{T \to \infty} \frac{1}{2\pi} \int_{-T}^{T} \frac{e^{-iut_1} - e^{-iut_2}}{iu} \phi(u)\, du . \tag{3.7}$$

Given (3.1), (3.4) and (3.7) can be obtained by integrating over *t*.

Lattice Distributions and Sequences of Real Numbers

If the cdf F attaches mass p_k to the point $x + k\lambda$, where $p_k \geq 0$ and $\Sigma\, p_k = 1$, then

$$\phi(u) = \sum_{k=-\infty}^{+\infty} p_k\, e^{i(x+k\lambda)u} \quad \text{and} \quad p_k = \frac{\lambda}{2\pi} \int_0^{2\pi/\lambda} \phi(u)\, e^{-i(x+k\lambda)u}\, du , \tag{3.8}$$

so that the inversion integral concentrates on the *bounded interval* $[0, 3\pi/\lambda]$; see p. 511 of Feller [42]. We can also calculate terms p_k of a sequence of real numbers $\{p_k : k \geq 0\}$ by representing the derivative of the generating function $G(z)$ in (2.8) as the Cauchy contour integral

$$p_k = \frac{1}{2\pi i} \int \frac{G(z)}{z^{k+1}}\, dz . \tag{3.9}$$

Upon making the change of variables $z = re^{iu}$, we obtain

$$\begin{aligned} p_k &= \frac{1}{2\pi r^k} \int_0^{2\pi} G(re^{iu}) e^{-iku}\, du \\ &= \frac{1}{2\pi r^k} \int_0^{2\pi} [Re(G(re^{iu})) \cos ku + Im(G(re^{iu})) \sin ku]\, du \\ &= \frac{2}{\pi r^k} \int_0^{\pi} Re(G(re^{iu})) \cos ku\, du & (3.10) \\ &= \frac{2}{\pi r^k} \int_0^{\pi} Im(G(re^{iu})) \sin ku\, du . & (3.10a) \end{aligned}$$

Just as we obtained (3.3) from (3.2), we can also obtain (3.10) from (3.8), without assuming that $\{p_k\}$ is a probability mass function, by considering the sequence $\{p_k r^k : k \geq 0\}$ and letting $x = 0$, $\lambda = 1$ and $G(z) = \sum_{k=0}^{\infty} p_k z^k$. Just as with the a in (3.3), the r in (3.10) is convenient to control the error in the numerical integration; see (5.35)–(5.42) below.

4. Numerical Integration

The functions given by (3.1)–(3.10) can be calculated by performing a numerical integration. Any available numerical integration routine can be applied (which partly explains why there are so many numerical inversion methods), but of course the numerical integration is complicated by the integrals being defined over the infinite domain $[0, \infty)$, except for (3.8)–(3.10); see Ch. 3 of Davis and Rabinowitz [32]. We propose to approximately calculate the integrals by truncating and applying the trapezoidal rule. The trapezoidal rule is often thought to be a very primitive procedure, but it turns out to be surprisingly effective in this context. In particular, it turns out to be better than familiar alternatives such as Simpson's rule for our inversion integrals.

The Trapezoidal Rule

The trapezoidal rule approximates the integral of a function g over the bounded interval $[a, b]$ by the integral of the piecewise linear function obtained by connecting the $n + 1$ evenly spaced points $g(kh)$, $0 \leq k \leq n$, where $h = (b - a)/n$; i.e.,

$$\int_a^b g(u)\,du \approx T_h(g) \equiv T_h(g;\, a, b) \equiv h\left[\frac{g(a) + g(b)}{2} + \sum_{k=1}^{n-1} g(a + kh)\right]; \qquad (4.1)$$

see p. 51 of Davis and Rabinowitz [32]. Formula (4.1) applies also when $a = -\infty$ or $b = +\infty$ with obvious modifications.

When the integrand g has a bounded continuous second derivative, the standard *exact expression for the error* from (4.1) using step size h is

$$e_d \equiv e_d(g;\, a,\, b,\, h) \equiv \left| \int_a^b g(t)\, dt - T_h(g) \right| = h^2\, \frac{(b-a)}{12}\, |g''(t_0)|\,, \tag{4.2}$$

where t_0 is some point satisfying $a < t_0 < b$, which leads to the associated error bound

$$e_d \le h^2\, \frac{(b-a)}{12}\, \sup\, \{ |g''(t)| : a < t < b \}\,; \tag{4.3}$$

see p. 32 of [27]. When the integrand g is a cf ϕ of a cdf with a finite second moment m_2, we can combine (4.3) with the bound $|\phi''(u)| \le |\phi''(0)| = m_2$; see (2.3).

Since the trapezoidal rule is one of the most primitive numerical integration procedures, its use may seem surprising. However, it tends to work well for periodic and other oscillating integrands, because the errors tend to cancel; indeed, the realized errors can be substantially less than from Simpson's and other less primitive rules; see pp. 134-142 of Davis and Rabinowitz [32], Kendall [72], Fettis [44], De Balbine and Franklin [34] and Rice [106]. The advantage of the trapezoidal rule for periodic functions is demonstrated by improved error bounds. For periodic functions with continuous and bounded k^{th} derivative, the error from using the trapezoidal rule over the interval of one period is bounded by Ah^k as opposed to Bh^2 in (4.3); see p. 137 of [32]. Moreover, we can apply the Poisson summation formula to obtain a convenient representation of the discretization error associated with the trapezoidal rule, enabling us to bound and estimate the error much more easily and precisely than from (4.2); see § 5.

We now apply (4.1) to treat a cdf F concentrating on $[0, \infty)$. From (3.6), we obtain

$$\begin{aligned} F(t) &= \frac{2}{\pi} \int_0^\infty Re\;(\phi)(u)\; \frac{\sin tu}{u}\; du \\ &\approx F_h(t) \equiv \frac{ht}{\pi} + \frac{2}{\pi} \sum_{k=1}^{\infty} \frac{Re\;(\phi)(kh)\, \sin(kht)}{k} \\ &\approx F_{h,N}(t) \equiv \frac{ht}{\pi} + \frac{2}{\pi} \sum_{k=1}^{N} \frac{Re\;(\phi)(kh)\, \sin(kht)}{k}\;, \end{aligned} \tag{4.4}$$

where $e_d \equiv e_d(F, t, h) \equiv |F(t) - F_h(t)|$ is the *discretization error* for the cdf F evaluated at t with step size h and $e_t \equiv e_t(F, t, h, N) \equiv |F_h(t) - F_{h,N}(t)|$ is the *truncation error* resulting from considering only the first N terms of the infinite series. The appropriate choice of h and N can be based on trial and error, but also on error estimates and bounds, as we show in Sections 5 and 6. We call the algorithm based on (4.4) POISSON, because the Poisson summation formula characterizes the discretization error; see §5.

Remarks (4.1) By (3.7), the integral in (4.4) is valid, at least in the limiting sense of (3.7), provided that t is a continuity point of F. Moreover, it is valid at points of discontinuity t if we replace $F(t)$ by $[F(t) + F(t-)]/2$. With this proviso, $F_{h,N}(t) \to F(t)$ as $h \to 0$ and $N \to \infty$. The problem is only to choose suitable h and N.

(4.2) Note that the final approximation $F_{h,N}(t)$ in (4.4) is a *trigonometric polynomial* of order N and period $2\pi/h$ (p. 6 of Tolstov [119]); i.e., the argument t appears only inside the sine function, so that $Re\ (\phi)(u)$ has to be evaluated only N times, no matter how many values of t are considered. Of course, this is primarily important for conceptual clarity. It indicates that numerical integration by the trapezoidal rule in this context reduces to a Fourier-series method. For further explanation, see §5.

(4.3) To apply (4.4), we need to evaluate the real part of the complex-valued function $\phi(u)$, for $u = kh$, $1 \le k \le N$, but with a language such as UBASIC or FORTRAN this will be *done by the computer*, provided that we have an explicit expression for the cf ϕ. For example, we have such an explicit expression for the M/G/1 waiting-time cf given (1.1) and the cf of the service-time distribution; this is illustrated in § 9. However, in many probability applications an explicit expression for the transform is not available. For example, the Kendall functional equation for the M/G/1 busy-period distribution (p. 212 of Kleinrock [75]) provides only an implicit expression for the cf; i.e., for each complex number s we must solve an equation to find

$\hat{f}(s)$. The Gaver-Stehfest procedure in §8 is convenient for this purpose, because it uses the transform values only at real arguments. However, for this example experience indicates that it is possible to iteratively solve for $\hat{f}(s)$ even with complex arguments. (More work needs to be done on transforms defined via functional equations.) ■

We now apply the trapezoidal rule to the Laplace transform inversion formula in (3.3). We obtain

$$f(t) = \frac{2e^{at}}{\pi} \int_0^\infty Re(\hat{f})(a + iu) \cos ut \, du$$

$$\approx f_h(t) \equiv \frac{he^{at} Re(\hat{f})(a)}{\pi} + \frac{2he^{at}}{\pi} \sum_{k=1}^{\infty} Re(\hat{f})(a + ikh) \cos(kht) \ . \tag{4.5}$$

We can now eliminate the cosine terms in (4.5) and produce nearly an alternating series by letting $h = \pi/2t$. Letting $a = A/2t$ at the same time, we obtain

$$f_h(t) = \frac{e^{A/2}}{2t} Re(\hat{f})(\frac{A}{2t}) + \frac{e^{A/2}}{t} \sum_{k=1}^{\infty} (-1)^k Re(\hat{f}) \left[\frac{A + 2k\pi i}{2t} \right] , \tag{4.6}$$

which is (21) of Dubner and Abate [40]. Formula (4.6) is the basis for the program EULER in §1; see §7 below. In (5.27) below we characterize the discretization error associated with (4.6).

If instead of (3.3) we work with the complementary form (3.3a), and instead of the trapezoidal rule apply the midpoint rule, then we obtain

$$f(t) = \frac{e^{A/2}}{t} \sum_{k=1}^{\infty} (-1)^k Im(\hat{f}) \left[\frac{A + (2k-1)\pi i}{2t} \right] , \tag{4.6a}$$

which serves as the basis for Hosono's [58], [59], [60] FILT algorithm. (The use of the midpoint rule was first suggested by Davies [30]; see (7) there.) In particular, (4.6a) is (1) on p. 20 of [60]. For an overview of the FILT, see p. 47 of [60]. In (5.30) below we characterize the discretization error associated with (4.6a).

Remark (4.4) Another mathematical explanation for the good performance of the trapezoidal rule is via the Whittaker cardinal function; see Stenger [114] and p. 215 of [32]. Given a complex-valued function f of a complex variable z and a positive step size h, the cardinal function is

$$C(f,h)(z) = \sum_{k=-\infty}^{\infty} f(kh) \frac{\sin((\pi/h)(z-kh))}{(\pi/h)(z-kh)} . \tag{4.7}$$

For very nice functions (entire functions such that $f \in L^2(R)$ and $|f(z)| \leq C e^{\pi|z|/h}$), $f(z) = C(f,h)(z)$ and

$$\int_{-\infty}^{\infty} f(t)\, dt = h \sum_{k=-\infty}^{\infty} f(kh) , \tag{4.8}$$

i.e., the trapezoidal rule is exact. Consequently, it is possible to get very good error bounds for trapezoidal rule approximations of appropriately nice functions. (For a large class of functions that are analytic in a strip about the real line, the overall error is $O(e^{-\sqrt{N}})$ when there are N terms and $h = O(1/\sqrt{N})$.) ■

Sections 5 and 6 below are devoted to the discretization and truncation errors, respectively. For both errors, the probabilistic structure is helpful. Now we consider ways to calculate the finite sum in (4.4) efficiently. Obviously this issue only becomes relevant when we must perform many calculations.

Computational Efficiency for the Finite Sums

If we want to calculate a cdf $F(t)$ via (4.4) for many values of t, then we can improve the computational efficiency by exploiting the periodic property of the sine function in (4.4). For example, suppose that we wish to calculate F at the $n+1$ evenly spaced values jt/n for $j = 0, 1, \ldots, n$. If we let the step size be $h = \pi/mt$ for some integer m, then $hjkt/n = jk\pi/mn$, so that $\sin(hjkt/n)$ has period at most $2mn$ as a function of k. Hence,

$$F(jt/n) \;=\; \frac{hjt}{\pi n} \;+\; \frac{2}{\pi} \sum_{k=1}^{2mn-1} \alpha_k \sin\left[\frac{\pi\, jk}{mn}\right] \;-\; e_d \tag{4.9}$$

for $j \;=\; 0, 1, \ldots, n$, where

$$\begin{aligned}\alpha_k \;&=\; \sum_{l=0}^{\infty} \frac{Re\,(\phi)((2mnl + k)\,h)}{2mnl + k} \\ &\approx \sum_{l=0}^{N/2mn} \frac{Re\,(\phi)((2mnl + k)\,h)}{2mnl + k}\,, \quad 0 \le k \le 2mn - 1\,.\end{aligned} \tag{4.10}$$

Applying (4.4) directly involves order nN evaluations, multiplications and additions, whereas (4.9) and (4.10) involves order $2mn^2 + N$ evaluations, multiplications and additions.

Moreover, the finite sum in (4.9) is the discrete sine transform (the imaginary part of the discrete Fourier transform) of $\{\alpha_k\colon\ 0 \le k \le 2mn - 1\}$ in (4.10), so that $F(jt/n)$ can be efficiently calculated for all j, $0 \le j \le 2mn - 1$, for large mn (when $n \approx 2mn$ and n is chosen to be a power of 2) by using a Fast Fourier Transform (FFT); see pages 51 and 357-368 of Rabiner and Gold [102], Cooley and Tukey [25], Cooley, Lewis and Welch [22], [23], [24] and Dubner and Abate [40]. (This reduces the computational burden from n^2 to $n \log_2 n$.) In addition to being computationally efficient when n is large, an FFT may be convenient because an FFT program is readily available. FFTs are even in the hardware (chips).

5. The Poisson Summation Formula

We now focus on the discretization error associated with the trapezoidal rule. As Fettis [44] observed in 1955, the trapezoidal rule for approximating certain integrals can be analyzed and justified by applying the Poisson summation formula. With the Poisson summation formula we are able to systematically control the discretization error. The Poisson summation formula itself is a classical result. It is interesting that it has a prominent place in Ramanujan's notebooks; see page 5 and Chapters 13 and 14 of Berndt [11]. The essential idea is to approximate the given function by a periodic function that can be represented by its Fourier-series. This explains the name ‘‘Fourier-series method.’’ In the signal processing and time-series literature, this is called ‘‘aliasing;’’ e.g., pp. 26-28 of Rabiner and Gold [102].

Probability Densities

Starting with a function $f(t)$, which we will regard as a probability density, the idea is to construct the *periodic function*

$$f_p(t) = \sum_{k=-\infty}^{+\infty} f(t + 2\pi k/h) , \tag{5.1}$$

of period $2\pi/h$, which usually can be made suitably close to $f(t)$ by choosing h small. We then represent the periodic function f_p by its *complex Fourier series* (which we assume converges to $f_p(t)$)

$$f_p(t) = \sum_{k=-\infty}^{\infty} c_k e^{ikht} , \tag{5.2}$$

where c_k is the k^{th} *Fourier coefficient* of f_p, which can be expressed in terms of the cf ϕ of f, i.e.,

$$\begin{aligned} c_k &= \frac{h}{2\pi} \int_{-\pi/h}^{\pi/h} f_p(t)\, e^{-ikht}\, dt = \frac{h}{2\pi} \int_{-\pi/h}^{\pi/h} \sum_{k=-\infty}^{\infty} f\left[t + \frac{2k\pi}{h}\right] e^{-ikht}\, dt \\ &= \frac{h}{2\pi} \int_{-\infty}^{\infty} f(t)\, e^{-ikht}\, dt = \frac{h}{2\pi} \phi(-kh) , \end{aligned} \tag{5.3}$$

with the interchange of summation and integration in (5.3) being justified by Fubini or Tonelli. (For background on Fourier-series, see Champeney [20], Tolstov [119] and Natanson [91].)

Combining (5.1)-(5.3), we obtain the *Poisson summation formula*

$$\sum_{k=-\infty}^{\infty} f\left[t + \frac{2\pi k}{h}\right] = \frac{h}{2\pi} \sum_{k=-\infty}^{\infty} \phi(kh)\, e^{-ikht} . \tag{5.4}$$

Then we can write

$$f(t) = \frac{h}{2\pi} \sum_{k=-\infty}^{\infty} \phi(kh)\, e^{-ikht} - e_d$$

$$= \frac{h}{2\pi} + \frac{h}{\pi} \sum_{k=1}^{\infty} [Re\,(\phi)(kh) \cos kht + Im\,(\phi)(kh) \sin kht] - e_d\,, \qquad (5.5)$$

where

$$e_d \equiv e_d(t) = \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} f\left[t + \frac{2\pi k}{h}\right] \qquad (5.6)$$

Note that *the sum in* (5.5) *is just the trapezoidal rule approximation* (4.1) *applied to the inversion integral* (3.1), *so that* e_d *in* (5.6) *is an explicit expression for the discretization error associated with the trapezoidal rule approximation*. Below we will see that we can obtain much better error bounds from (5.6) than from (4.2) and (4.3). Moreover, as noted in Remark 4.2, the approximating finite sums are trigonometric polynomials.

The rest of this subsection through Proposition 5.2 below is primarily devoted to the rigorous justification of (5.2) and (5.4) and so might well be omitted upon first reading. (However, (5.9) below is a useful generalization of (5.4).) A nice account for probabilistic applications involving continuous probability densities is given on pp. 626-631 of Feller [43]; we begin with it. In order to exploit the nonnegativity and integrability properties of densities (see his Theorem 1 on p. 628), Feller starts with the cf ϕ and forms an associated periodic function ϕ_p. Assuming that $|\phi|$ is integrable, so that ϕ is a cf of a bounded continuous density *f*, we form the *periodic function* (of period *h*)

$$\phi_p(u) = \sum_{k=-\infty}^{\infty} \phi(u + kh)\;. \qquad (5.7)$$

Assuming that the series in (5.7) converges to a continuous function, we obtain the Poisson summation formula by representing ϕ_p by its complex Fourier-series using (3.1), as in (5.2)-(5.4) above, i.e.,

$$\phi_p(u) \equiv \sum_{k=-\infty}^{\infty} \phi(u+kh) = \frac{2\pi}{h} \sum_{k=-\infty}^{\infty} f(2k\pi/h) e^{i2k\pi u/h} ; \tag{5.8}$$

see (5.6) on p. 630 of [43] (there $h = 2\lambda$); i.e., $(2\pi/h) f(2k\pi/h)$ is the k^{th} Fourier coefficient of ϕ_p and the Fourier-series converges uniformly to ϕ_p. By applying (5.8) to the density $f(t+y)$, we obtain the alternate form

$$\sum_{k=-\infty}^{\infty} \phi(u+kh) e^{-it(u+kh)} = \frac{2\pi}{h} \sum_{k=-\infty}^{\infty} f\left[\frac{2k\pi}{h} + t\right] e^{i\,2k\pi u/h} ; \tag{5.9}$$

see (5.9) on p. 633 of [43]. The extra variables in (5.9) make it convenient for further derivations. Setting $u = 0$ in (5.9), we obtain (5.4), which is the form we mainly apply.

For continuous densities it is usually possible to directly verify the sufficient conditions on ϕ from [43], but not all distributions have continuous densities. For example, even an exponential density is not continuous at 0, and indeed, for an exponential density with mean m, the cf is $\phi(u) = (1 - m\,iu)^{-1}$, so that $|\phi|(u) = (1 + m^2 u^2)^{-1/2}$ is not integrable. Hence, the validity of (5.5) has not been established in general.

Obviously, in general the series in (5.1) need not even converge; e.g., it is possible to have a bounded continuous density with $f(t + 2\pi k/h) = 1$ for infinitely many k (let f be the indicator function of the union of intervals $[t + (2\pi k/h) - a_k,\ t + (2\pi k/h) + a_k]$, $k \geq 1$, where $a_k \leq \pi/h$ and $\sum_{k=1}^{\infty} 2a_k = 1$). Even when the series in (5.1) converges uniformly to a continuous limit, as will be the case in our applications to cdf's below when the cdf is continuous, the Fourier-series in (5.2) need not converge in the usual sense. However, this is a classical difficulty for Fourier-series which has been studied extensively. There are three ways to resolve this difficulty: (1) require that f satisfy stronger conditions which imply that the Fourier-series converges and (5.2) is valid, (2) perturb the density f by convolution smoothing so that the new density satisfies the conditions or (3) change the form of the summation of the Fourier-series in

(5.2). (These last two approaches turn out to be closely related; see § 6.)

For directly establishing convergence and equality in (5.2), we can apply classical criteria; e.g., see Chapter 3 of Tolstov [119]. (Note that f_p in (5.1) is integrable over the interval of one period as well as periodic.)

Proposition 5.1. Let g be a periodic function such that $|g|$ *is integrable over an interval of one period. If g has only finitely many discontinuities and has left and right derivatives everywhere, then the Fourier series of g converges pointwise to* $[g(t+) + g(t-)]/2$ *at all t.*

Remark (5.1) It is significant that the convergence in Proposition 5.1 is *not uniform* in the neighborhood of a point of discontinuity. Moreover, at a point of discontinuity there is the important *Gibbs phenomenon;* see Ch. 9 of Carslaw [18], p. 88 of Rabiner and Gold [102] and Foster and Richards [46]. In particular, there are systematic undershoots and overshoots in the neighborhood of such a point that approach a constant *c* times the value of the jump. The constant *c* is

$$c = \pi^{-1} \int_{\pi}^{\infty} x^{-1} \sin x \, dx = 0.2811/\pi \approx 0.09 \ .$$

This means that the sequence of graphs of the approximants approach the graph of the given function in the Hausdorff metric if and only if the graph of the given function is completed by adding a vertical line at each point of discontinuity, where this vertical line extends above the upper limit and below the lower limit by about 0.09 times the value of the jump there. ■

Convergence and equality in (5.2) can be established with weaker conditions if we change the form of summation of the Fourier-series; see Chapter 6 of [119]. For example, Proposition 5.1 remains valid without any condition on the derivatives if we use Cesàro convergence (arithmetic means). It is also important to note that convergence is a *local property*; i.e., the conclusions above are valid at *t* if the conditions on *g* hold in some open interval containing *t*.

Finally, we can also apply criteria involving the Fourier coefficients; pp. 91, 100 of [119]. These conditions differ only slightly from Feller's [43].

Proposition 5.2. (a) *If* $\sum_{k=1}^{\infty} (|Re(\phi)(kh)| + |Im(\phi)(kh)|) < \infty$, *then* (5.2) *is valid and the Fourier-series converges uniformly.*

(b) *If* $Re(\phi)(kh)$ *and* $Im(\phi)(kh)$ *are eventually nonnegative and decrease to 0 as* $k \to \infty$, *then* (5.2) *is valid and the Fourier-series converges for any t, except* $t = 2\pi k/h$.

CDFs on the Positive Real Line

In most probability applications we are more interested in cdf's than densities, because probabilities of intervals are usually of primary interest. When there is negligible difference between two cdf's, we are usually content to consider the two distributions equivalent. This is useful because we can make large changes in a density in the neighborhood of a point while producing only a minor change in the corresponding cdf. Focusing on cdf's is also attractive because cdf's have nice properties (monotonicity, boundedness and more smoothness) that make them easier to calculate by numerical inversion.

As Bohman [14] observed, the Poisson summation formula applies very nicely to a cdf F concentrating on the positive half line. When we apply the Poisson summation formula, we obtain both the trapezoidal rule formula (4.4) and the discretization error associated with inversion integral (3.6), which provides the basis for algorithm POISSON. In general (without assuming $F(0-) = 0$), we can apply (5.5) with (3.4) and then set $t = 0$ and $x = t$ to obtain

$$F(t) - F(-t) = \frac{h}{\pi} \sum_{k=-\infty}^{\infty} Re(\phi)(kh) \frac{\sin kht}{kh} - e_d$$
$$= \frac{ht}{\pi} + \frac{2h}{\pi} \sum_{k=1}^{\infty} Re(\phi)(kh) \frac{\sin kht}{kh} - e_d \,, \tag{5.10}$$

where

$$e_d \equiv e_d(t) = \sum_{\substack{k=-\infty \\ k \neq 0}}^{\infty} \left[F\left[\frac{2k\pi}{h} + t \right] - F\left[\frac{2k\pi}{h} - t \right] \right] . \tag{5.11}$$

Alternatively, we can apply the Poisson summation formula directly with the function $F(t) - F(-t)$ to obtain the aliased periodic function $F(t) - F(-t) + e_d(t)$ with period $2\pi/h$ and (5.10). (Of course, to obtain (5.10) this way we need to calculate the Fourier coefficients as in (5.3).)

If, in addition, $F(0-) = 0$ and $0 < t \leq \pi/h$, then $F(t) = F(t) - F(-t)$ and (5.10) becomes (4.4) plus its discretization error. Moreover, the discretization can be conveniently bounded above and below by

$$0 \leq e_d = \sum_{k=1}^{\infty} \left[F\left[\frac{2k\pi}{h} + t \right] - F\left[\frac{2k\pi}{h} - t \right] \right] \leq F^c \left[\frac{2\pi}{h} - t \right] , \tag{5.12}$$

where $F^c(t) = 1 - F(t)$.

Note that the aliased function $F(t) + e_d(t)$ on $[0, \pi/h]$ is related to $F(t)$ in a different way than $f_p(t)$ is related to $f(t)$ in (5.1). To visualize the aliased function (which is actually represented by the series), it is perhaps helpful to look at the complementary cdf $F^c(t)$, which converges to 0 as $t \to \infty$; then the aliased function is

$$F_p^c(t) = F^c(t) - \sum_{k=1}^{\infty} F^c \left[\frac{2k\pi}{h} - t \right] + \sum_{k=1}^{\infty} F^c \left[\frac{2k\pi}{h} + t \right] .$$

The function $F_p^c(t)$ for $t \in [0, \pi/h]$ can thus be thought of as the sum over k, with alternating sign, of the function F_p^c over the subintervals $[k\pi/h, (k+1)\pi/h)$, with the function flipped over ($F_p^c(k\pi/h + t)$ replaced by $F_p^c((k+1)\pi/h - t)$) for odd k.

Since F is a cdf, it is easy to see that the conditions of Proposition 5.1 are satisfied for the periodic function in (5.1) associated with the density $f_x(y)$ in (3.4), which justifies (5.10), when F

is absolutely continuous (has a density), when F is a discrete cdf concentrating on finitely many points, or when F is a mixture of these two kinds of cdf's. Moreover, the discrete component can have countably many jumps provided that all but finitely many occur at the points ky for k integers, and h is chosen so that $2\pi/h$ is an integral multiple of y. Then the periodic function will have only finitely many discontinuities on the interval $[-\pi/h, \pi/h]$. (We rule out the relatively pathological discrete cdf's with infinitely many jumps irregularly spaced and the singular continuous cdf's; see § 1.3 of Chung [21].)

In this case, $F(t)$ is approximated by applying the trapezoidal rule (4.1) to the inversion integral (3.6) with step size h, which yields (4.4). Since $e_d \geq 0$, the infinite series in (4.4) is always an upper bound for $F(t)$. Moreover, *the discretization error* e_d *is bounded above by the simple tail probability* $F^c((2\pi/h) - t)$. Hence, the discretization error is relatively easy to bound and estimate. In particular, the required step size to achieve a prescribed discretization error ε is

$$h = 2\pi/(t+u_\varepsilon) \geq \pi/u_\varepsilon \ , \quad \text{where} \quad F^c(u_\varepsilon) = \varepsilon \quad \text{and} \quad 0 \leq t \leq u_\varepsilon \ . \tag{5.13}$$

If we want to evaluate F for t throughout the interval $[0, \pi/h]$, then we should set $t = \pi/h$ in (5.13), so that $u_\varepsilon = \pi/h$ and $h = \pi/u_\varepsilon$.

Note that the error bound in (5.12) is typically much better than the standard error bound associated with the trapezoidal rule in (4.3). For example, if $F^c(t) \leq e^{-\mu t}$, $t \geq 0$, for some $\mu > 0$, then $e_d\,(F, t, h) \leq e^{\mu t} e^{-2\mu\pi/h}$ from (5.12), so that e_d is of order $e^{-2\mu\pi/h}$, whereas from (4.3) we obtain only that e_d is at most of order h^2. *Moreover, unlike* (4.3)*, the discretization error bound in* (5.12) *is independent of the length of the interval of integration.*

It can happen that the step size h prescribed by (5.13) is very small, so that the resulting computation is daunting. This difficulty occurs when the tail probabilities decay slowly; this difficulty can typically be overcome by considering a damped function, as in the subsection after the next. From the perspective of the contour integral considered there, this corresponds to moving the contour to the right further away from singularities in the left halfplane.

CDFs on the Whole Real Line

Most probability applications seem to involve cdf's on the positive real line, for which §5.2 applies. However, with a simple modification, (5.10) also applies to cdf's on the whole line. It suffices to shift the cdf to the right (adding a constant y to the random variable or replacing $F(t)$ by $F(t-y)$) until $F(0-)$ becomes suitably small, and then shift the cdf back after the inversion has been done. Then $F(-t)$ in (5.10) becomes part of the error and, from (5.11), $e_d \le F(t - (2\pi/h)) + F^c((2\pi/h) - t)$. It typically will be easier to work with the original cf on the whole line multiplied by e^{iuy}, but it is also possible to work with the cf of the shifted distribution restricted to the positive real line (the conditional distribution given that it is nonnegative), provided that this new cf can be determined.

Alternatively, we can apply (5.5) with an analog of (3.4) to obtain

$$
\begin{aligned}
F(t) - F(t-y) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} \phi(u) \, \frac{e^{iuy} - 1}{iu} \, e^{-iut} \, du \\
&= \frac{h}{2\pi} \sum_{k=-\infty}^{\infty} \phi(kh) \frac{e^{ikh(y-t)} - e^{-ikht}}{ikh} - e_d \\
&= \frac{hg(0)}{2\pi} + \frac{h}{\pi} \sum_{k=1}^{\infty} g(kh) - e_d \,, \qquad (5.14)
\end{aligned}
$$

where $g(0) = y$,

$$
g(u) = \frac{Re(\phi)(u)}{u} \, [\sin u\,(y-t) + \sin ut] + \frac{Im(\phi)(u)}{u} \, [\cos u\,(y-t) - \cos ut] \qquad (5.15)
$$

and

$$e_d \equiv e_d(t) = \sum_{k=1}^{\infty} \left[F\left[\frac{2k\pi}{h} + t\right] - F\left[\frac{2k\pi}{h} + t - y\right]\right]$$

$$+ \sum_{k=1}^{\infty} \left[F\left[\frac{-2k\pi}{h} + t\right] - F\left[\frac{-2k\pi}{h} + t - y\right]\right] . \qquad (5.16)$$

If $y \leq 2\pi/h$, then the discretization error e_d is bounded above by the sum of two simple tail probabilities, i.e.,

$$e_d \leq F^c\left[\frac{2\pi}{h} + t - y\right] + F\left[\frac{-2\pi}{h} + t\right] . \qquad (5.17)$$

Formula (5.14) is valid subject to essentially the same qualifications as (5.10). Note that (5.14)-(5.17) reduce to our direct application of (5.10) above when we let $y = 2t$.

To illustrate how we can apply (5.17), suppose that we wish to calculate $F(t)$ within ε for a given t. Let l_ε and u_ε be numbers such that $F(l_\varepsilon) \leq \varepsilon/4$ and $F^c(u_\varepsilon) \leq \varepsilon/4$. If $t \leq l_\varepsilon$, let $F(t) \approx 0$; if $t \geq u_\varepsilon$, let $F(t) \approx 1$. Henceforth, assume that $l_\varepsilon < t < u_\varepsilon$. For t given, let

$$y = t - l_\varepsilon \quad \text{and} \quad h = \frac{2\pi}{u_\varepsilon - l_\varepsilon} , \qquad (5.18)$$

so that

$$y < u_\varepsilon - l_\varepsilon = 2\pi/h, \quad t - y = l_\varepsilon, \quad \frac{2\pi}{h} + t - y = u_\varepsilon, \quad \frac{-2\pi}{h} + t < l_\varepsilon ,$$
$$F(t-y) \leq \varepsilon/4 \quad \text{and} \quad e_d \leq \varepsilon/2 . \qquad (5.19)$$

Hence,

$$\left| F(t) - \left[\frac{hg(0)}{2} + h\sum_{k=1}^{\infty} g(kh)\right]\right| \leq \max\,\{F(t-y),\, e_d\} \leq \varepsilon/2 \qquad (5.20)$$

for g in (5.15). Hence, we obtain $F(t)$ to within ε when we truncate the sum in (5.20) and produce an error of at most $\varepsilon/2$, i.e., when we find N such that

$$\left| h \sum_{k=N+1}^{\infty} g(kh) \right| \leq \varepsilon/2 \ . \tag{5.21}$$

We consider the truncation problem in § 6.

Remarks (5.2) If we want to calculate $F(t)$ for several t with $l_\varepsilon \leq b \leq t \leq c \leq u_\varepsilon$, then instead of (5.18) we let

$$y = c - l_\varepsilon \quad \text{and} \quad h = \frac{2\pi}{(u_\varepsilon - l_\varepsilon) + (c - b)} \ , \tag{5.22}$$

so that

$$y \leq u_\varepsilon - l_\varepsilon \leq \frac{2\pi}{h} \ , \qquad t - y \leq c - y = l_\varepsilon \ ,$$

$$\frac{2\pi}{h} + t - y = u_\varepsilon + t - b \geq u_\varepsilon \quad \text{and} \quad \frac{-2\pi}{h} + t \leq l_\varepsilon \ . \tag{5.23}$$

(5.3) To calculate the cdf $F(t)$, it may seem natural to begin with the direct expression provided by Gil-Palaez [51] in (3.5) instead of (3.4) and (5.14), but this approach seems less satisfactory. We can of course proceed directly as in (5.1)–(5.6), but the resulting expression for the discretization error (the analog of (5.6)) is not as easy to work with as (5.12) or (5.17).

Functions on the Positive Real Line

It is obviously not necessary for f in (5.1) to be a probability density. It suffices for $|f|$ to be integrable, in addition to other conditions given above. Moreover, if f is defined on the positive real line but $|f|$ is not integrable, then we can replace f by the damped function $f_d(t) = e^{-at} f(t)$ with ‘‘cf’’

$$\phi_d(u) = \int_{-\infty}^{\infty} e^{iut} f_d(t) \, dt = \int_0^{\infty} e^{iut} e^{-at} f(t) \, dt = \phi(u + ia) \ .$$

If $|f_d|$ is integrable and satisfies other conditions above, then we can numerically invert $\phi_d(u)$ to get an approximation to $f_d(t)$, say $f_{da}(t)$, and let $f(t) \approx e^{at} f_{da}(t)$. This damping procedure was used by Dubner and Abate [40] to invert general Laplace transforms. For example, we can use this approach to calculate the renewal function from its transform; see §13.

In particular, we can apply the Poisson summation formula (5.4) with the damped function $e^{-at} f(t)$ defined on the positive real line to identify the discretization error associated with the trapezoidal rule applied to inversion integral for Laplace transforms in (3.3), as given in (4.6). From (5.4) with function $e^{-at} f(t)$ and $h = \pi/t$, we obtain

$$\begin{aligned}\sum_{k=0}^{\infty} e^{-a(2k+1)t} f((2k+1)t) &= \frac{1}{2t} \sum_{k=-\infty}^{\infty} (-1)^k \phi(ia + \frac{k\pi}{t}) \\ &= \frac{1}{2t} \sum_{k=-\infty}^{\infty} (-1)^k \hat{f}\left[a - \frac{k\pi i}{t}\right] . \end{aligned} \tag{5.25}$$

Letting $a = A/2t$ in (5.25), we obtain

$$\sum_{k=0}^{\infty} e^{-kA} f((2k+1)t) = \frac{e^{A/2}}{2t} \sum_{k=-\infty}^{\infty} (-1)^k \hat{f}\left[\frac{A - 2k\pi i}{2t}\right] . \tag{5.26}$$

Since $\hat{f}(s) + \hat{f}(\bar{s}) = 2Re(\hat{f})(s)$, we obtain

$$f(t) = \frac{e^{A/2}}{2t} Re(\hat{f})(\frac{A}{2t}) + \frac{e^{A/2}}{t} \sum_{k=1}^{\infty} (-1)^k Re(\hat{f}) \left[\frac{A + 2k\pi i}{2t}\right] - e_d \tag{5.27}$$

as in (4.6), where e_d is the discretization error, i.e.,

$$e_d \equiv e_d(f, t, A) = \sum_{k=1}^{\infty} e^{-kA} f((2k+1)t) . \tag{5.28}$$

Using (5.28), we can easily bound the discretization error in (4.6). First, if $|f(t)| \le c$, then $|e_d| \le ce^{-A}/(1 - e^{-A}) \approx ce^{-A}$. Hence, for cdf's we obtain a convenient bound on the discretization error independent of the distribution. If $|f(t)| \le ct$, then

$$|e_d| \le \sum_{k=1}^{\infty} e^{-kA}(2k+1)ct \le ct\frac{(3e^{-A} - e^{-2A})}{(1 - e^{-A})^2} \approx 3cte^{-A} \ . \qquad (5.29)$$

From (5.28) it is clear that we are motivated to choose A large in order to make the discretization error e_d small. However, increasing A can make the computation (5.27) more difficult, e.g., by making the series harder to sum or by requiring higher precision to avoid roundoff error. Thus we typically do not try to make A too large. From experience, at least $3\gamma/2$-digit machine accuracy is required to achieve an error of order $10^{-\gamma}$; see Remark 5.8 below. We typically aim for 10^{-7} or 10^{-8} accuracy on a machine with 14-digit precision.

As we indicated before, instead of (3.3) and (4.6), we can also use the complementary expressions (3.3a) and (4.6a) with $Im(\hat{f})$ instead of $Re(\hat{f})$. Applying (5.9) to the damped function $e^{-at}f(t)$ and letting $u = -h/2$, $h = \pi/t$ and $a = A/2t$, we obtain

$$f(t) + \sum_{k=1}^{\infty} (-1)^k e^{-kA} f((2k+1)t) = \frac{e^{A/2}}{t} \sum_{k=1}^{\infty} (-1)^k Im(\hat{f}) \left[\frac{A + (2k-1)\pi i}{2t} \right] , \qquad (5.30)$$

which is the variant of (5.27) obtained by Hosono [58], [59], [60].

Remark (5.4). Hosono obtains (5.30) in [58], [59], [60] by a different argument, without mentioning the trapezoidal rule or the Poisson summation formula. In particular, he obtains (5.30) by approximating the integrand in the Bromwich integral (3.3a) using

$$\exp(s) \approx \frac{\exp(a)}{2 \cosh (a-s)} = \frac{\exp(a)}{2} \sum_{n=-\infty}^{\infty} i(-1)^n/[s-a-i(n-0.5)\pi] \ , \qquad (5.31)$$

see (3)–(7) of [59]. In fact, (5.31) also can be derived from the Poisson summation formula; see pages 469-470 of Magnus, Oberhettinger and Soni [85].

Lattice Distributions and Sequences of Real Numbers

To treat pmf's and other sequences of real numbers, we apply the Poisson summation formula to the inversion integral (3.10). Since the inversion integral (3.10) is over a bounded interval, we avoid the truncation problem. Hence, the following analysis of the discretization error leads to a convenient algorithm with a simple error bound; i.e., there is no need to use two methods for generating functions.

Suppose that we have

$$g(u) = \sum_{k=-\infty}^{\infty} a_k e^{iku} \quad \text{and} \quad a_n = \frac{1}{2\pi} \int_0^{2\pi} g(u) e^{-inu}\, du \ . \tag{5.32}$$

Paralleling (5.1), form the periodic sequence

$$a_n^p = \sum_{k=-\infty}^{\infty} a_{n+km} \tag{5.33}$$

for some integer m and construct its discrete Fourier transform; see p. 51 of Rabiner and Gold [102]. In particular,

$$\begin{aligned} b_n^p &= \frac{1}{m} \sum_{j=0}^{m-1} a_j^p \ e^{i2\pi nj/m} \\ &= \frac{1}{m} \sum_{j=0}^{m-1} \sum_{l=-\infty}^{\infty} a_{j+lm} \ e^{i2\pi nj/m} \\ &= \frac{1}{m} \sum_{j=-\infty}^{\infty} a_j e^{i2\pi nj/m} = \frac{1}{m} g(2\pi/m) \ . \end{aligned}$$

From the inversion formula for discrete Fourier transforms,

$$\begin{aligned} a_n^p &= \sum_{k=0}^{m-1} b_k^p \ e^{-i2\pi nk/m} \\ &= \frac{1}{m} \sum_{k=0}^{m-1} g(2\pi k/m) e^{-i2\pi nk/m} \ , \end{aligned}$$

which with (5.33) yields the discrete Poisson summation formula

$$\sum_{k=1}^{m} g(2\pi k/m) e^{-\frac{ik2\pi n}{m}} = m \sum_{k=-\infty}^{\infty} a_{n+km} \ . \tag{5.34}$$

Given the generating function $G(z) = \sum_{k=0}^{\infty} p_k z^k$ in (2.8), we can combine the first line of

(3.10) and (5.34) to obtain, for the case $a_n = p_n r^n$ and $n < m$,

$$p_n = \frac{1}{mr^n} \sum_{k=1}^{m} G(re^{2\pi i\, k/m}) e^{\frac{-2\pi i n k}{m}} - e_d \ , \tag{5.35}$$

where the discretization error is

$$e_d = \sum_{j=1}^{\infty} p_{n+jm} r^{jm} \ . \tag{5.36}$$

We can bound the discretization error in (5.36) just as we did with (5.28). In particular if $\{p_n : n \geq 0\}$ is a probability mass function or a cdf, then

$$e_d \leq \frac{r^m}{1 - r^m} \approx r^m \ . \tag{5.37}$$

Of course, we can always obtain p_0 directly by $p_0 = G(0)$. For $n \geq 1$, it is convenient to let $m = 2n$ in (5.35), so that

$$e^{-2\pi i n k/m} = e^{-ik\pi} = (-1)^k$$

and

$$\begin{aligned} p_n &= \frac{1}{2nr^n} \sum_{k=1}^{2n} (-1)^k Re(G(re^{\pi k i/n})) - e_d \ , \quad n \geq 1 \ , \\ &= \frac{1}{2nr^n} \left\{ G(r) + (-1)^n G(-r) + 2 \sum_{k=1}^{n-1} (-1)^k Re(G(re^{\pi k i/n})) \right\} - e_d \, , \ n \geq 1 \, , \end{aligned} \tag{5.38}$$

for

$$e_d = \sum_{j=1}^{\infty} p_{(2j+1)n} r^{2jn} \ . \tag{5.39}$$

with the second line in (5.38) holding because $Re(G(z)) = Re(G(\overline{z}))$. Since the resulting discretization error e_d in the probability case is estimated by r^{2n}, we achieve accuracy to $10^{-\gamma}$ by setting $r = 10^{-\gamma/2n}$.

As with (5.28), we are motivated to make γ large when we let $r = 10^{-\gamma/2n}$ to produce a small discretization error in (5.38). However, a larger γ means a smaller r and a smaller r can require higher precision in order to avoid roundoff error. As with (5.28), a rough rule of thumb is to have at least $3\gamma/2$-digit precision to achieve $10^{-\gamma}$ accuracy; see Remark 5.8 below.

We call the algorithm based on (5.38) LATTICE-POISSON. We display below a UBASIC program to calculate the complementary cdf of the number of customers served in an M/M/1 busy period. The accuracy has been set at 10^{-8} by having $\gamma \equiv E = 8$ in line 21. The generating function is specified in lines 80-84.

Insert Algorithm LATTICE-POISSON here (or slightly later)

This particular inversion problem is discussed further in §11.

Remarks (5.5) There are many formulas essentially equivalent to (5.38) that can be derived from variants of the Poisson summation formula; e.g., see pages 233 and 235 of Berndt [11]. (Elementary algebra shows that p. 233 of Berndt [11] is equivalent to (5.38) and (5.39). To obtain a discrete analog of (5.30) from (3.10a), we can use the following discrete analog of (5.9)

$$\sum_{k=1}^{m} \phi(u + \frac{k\pi}{m}) e^{il(u + \frac{k\pi}{m})} = 2m \sum_{k=-\infty}^{\infty} a_{2km+l} e^{i2kmu} . \tag{5.40}$$

Paralleling the derivation of (5.30) from (5.9), let $u = -\pi/2m$, $m = n$, $l = n$ and $a_n = p_n r^n$ in (5.40) to obtain

$$\begin{aligned} p_n &= \frac{1}{2nr^n} \sum_{k=1}^{n} \phi((2k-1)\frac{\pi}{2n}) e^{-i((2k-1)\frac{\pi}{2})} - e_d \\ &= \frac{1}{2nr^n} \sum_{k=1}^{n} (-1)^k Im(\phi((2k-1)\frac{\pi}{2n}) - e_d \\ &= \frac{1}{2nr^n} \sum_{k=1}^{n} (-1)^k Im(G(re^{i(2k-1)\pi/2n})) - e_d \end{aligned} \tag{5.41}$$

where

$$e_d = \sum_{k=1}^{\infty} (-1)^k p_{(2k+1)n} r^{2kn} < \sum_{k=1}^{\infty} r^{2kn} \approx r^{2n} . \tag{5.42}$$

Formula (5.41) is the algorithm developed by Hosono [61].

(5.6) Algorithm LATTICE-POISSON should perhaps be considered classical, but it does not appear to be well known. The essential ideas in the algorithm LATTICE-POISSON are expressed in the introduction to Lyness [82], but the focus there is on further analysis using the Möbius function to treat the case in which we need not have $|p_k| \leq 1$ for all k. A variant of algorithm LATTICE-POISSON is used by Jagerman [67] in his approach to Laplace transforms (see §8) and is part of his MATHCALC package [68]. Algorithm LATTICE-POISSON is essentially equivalent to Hosono's algorithm in (5.41). For more on the history, see §15.

(5.7) Abate and Dubner [1] developed a Fourier-series method for lattice functions, which started by constructing the smoothed function obtained by linearly interpolating between successive values of the function. For cdf's, this is equivalent to convolving the original lattice cdf with a random variable uniformly distributed on the interval $[-1, 0]$. Then the Poisson summation formula was applied as for Laplace transforms [40]. The resulting algorithm is effective, as is illustrated by Dubner [39]. However, we have recently discovered that this algorithm is essentially equivalent to a direct application of (3.10); e.g., with a damped function we obtain an algorithm equivalent to LATTICE-POISSON. We intend to discuss this equivalence elsewhere.

(5.8) To understand the roundoff error problem, it is helpful to consider (5.38) in the case $n = 1$. When $n = 1$, (5.38) reduces to the difference of two terms divided by $2r$. In a typical example p_1 will be of order 1. Hence, when r is small, the difference must be of order r. Thus, when r is very small, the subtraction can cause a significant roundoff error. For a concrete example, consider the Bernoulli probability mass function with $p_1 = 1 - p_0 = p$, $0.1 < p < 0.9$. Then $G(z) = 1 - p + pz$ and (5.38) becomes

$$p_1 = [-(1-p-pr) + (1-p+pr)]/2r = p \ .$$

Since $0.1 < p < 0.9$, the precision of p_1 is approximately $m - k$ digits when the machine has m-digit precision and $r = 10^{-k}$. If the discretization error has been set at $10^{-\gamma}$, then by (5.37) $r = 10^{-\gamma/2}$. Thus we obtain the desired $10^{-\gamma}$ accuracy when $m - \gamma/2 \geq \gamma$ or $m \geq 3\gamma/2$. From (5.37) and (5.38), we see that a similar analysis can be done for other values of n. Since $r = 10^{-\gamma/2n}$ to achieve accuracy $10^{-\gamma}$, $r^n = 10^{-\gamma/2}$ for all n in (5.38). Moreover, in the case of a probability mass function $|Re(G(re^{-\pi ki/n}))| \leq 1$, so that essentially the same reasoning applies to any n. If, as in UBASIC, there is a fixed number of decimal places for each number, then we also get the 3/2 rule for the ratio of decimal places to final accuracy.

(5.9) It is not necessary to have a lattice function in order to apply the Poisson summation formula to an integral over a finite interval. For example, the Bessel function satisfies

$$J_0(xt) = \frac{1}{\pi} \int_{-x}^{x} (x^2 - u^2)^{-1/2} e^{-itu} \, du \ , \tag{5.43}$$

from which the Schloemilch series in 3.13.3 on p. 131 of Magnus et al. [85] follows from the Poisson summation formula.

Higher Dimensions

The Poisson summation formula is easily extended to higher dimensions (e.g., see Good [52]), so that we can obtain the discretization error associated with the trapezoidal rule applied to numerically invert transforms of multivariate functions. (However, the computational complexity increases as the dimension increases.) For example, the two-dimensional analogs of (2.5) and (3.1) are

$$\phi(u_1, u_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{i(u_1 t_1 + u_2 t_2)} f(t_1, t_2) \, dt_1 \, dt_2 \tag{5.44}$$

and

$$f(t_1, t_2) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(t_1 u_1 + t_2 u_2)} \phi(u_1, u_2) du_1 du_2 \ . \tag{5.45}$$

Paralleling (5.1), the periodic function $f_p(t_1, t_2)$ constructed by aliasing is

$$f_p(t_1, t_2) = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} f\left[t_1 + \frac{2\pi j}{h_1}, t_2 + \frac{2\pi k}{h_2} \right] . \tag{5.46}$$

Reasoning just as in (5.2)–(5.6), we obtain the bivariate Poisson summation formula

$$f(t_1, t_2) = \frac{h_1 h_2}{4\pi^2} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \phi(-jh_1, -kh_2) e^{-i(jh_1 t_1 + kh_2 t_2)} - e_d \tag{5.47}$$

where

$$e_d = \sum_{\substack{j=-\infty \\ \text{not } j=0,\, k=0}}^{\infty} \sum_{k=-\infty}^{\infty} f\left[t_1 + \frac{2\pi j}{h_1}, t_2 + \frac{2\pi k}{h_2} \right] . \tag{5.48}$$

6. Truncation, Convolution Smoothing and Alternative Methods of Summation

In this section we discuss ways to approximately calculate the infinite series obtained by applying the trapezoidal rule (or, equivalently, the Poisson summation formula) to the inversion integral, i.e., the infinite series in (4.4), (4.6), (5.5), (5.10), (5.14) or (5.27). Of course, no truncation is necessary for lattice distributions using algorithm LATTICE-POISSON in §5.

Direct Truncation Bounds

The problem of truncating the infinite series is nearly equivalent to the problem of truncating the original inversion integral, because if β is a *nonnegative nonincreasing real-valued function,* then

$$h \sum_{k=N+1}^{\infty} \beta(kh) \le \int_{Nh}^{\infty} \beta(u)\, du \le h \sum_{k=N}^{\infty} \beta(kh) \ . \tag{6.1}$$

We can apply (6.1) to an integrand of interest $\gamma(u)$ by bounding $|\gamma(u)|$ above by $\beta(u)$, i.e.,

$$e_t \equiv e_t\,(\gamma, h, N) \equiv \left| h \sum_{k=N+1}^{\infty} \gamma(kh) \right| \leq h \sum_{k=N+1}^{\infty} |\gamma(kh)| \leq h \sum_{k=N+1}^{\infty} \beta(kh)\ . \qquad (6.2)$$

Hence, if $\int_T^{\infty} \beta(u)\, du < \varepsilon$ for some prescribed truncation error ε, then, the required number of terms in the truncated finite series is $N = T/h$.

It is significant that the step size h in the discretization depends only on the prescribed discretization error and the associated error bounds (i.e., is independent of the truncation and T), while the truncation point T for the integral obtained from (6.1) and (6.2) depends only on the prescribed truncation error and the associated truncation error bound for the original integral (i.e., is independent of the discretization and h). Finally, the overall number of terms N in the finite sum is product of T and h^{-1}.

For most of this section, we will focus on the variant POISSON in (3.6), (4.4) and (5.10) applied to a cdf on the positive half line. Hence, consider the case of a cdf F concentrating on the positive real line with the infinite series in (5.10). We want to apply (6.1) and (6.2) with the integrand $\gamma(u) = 2\, Re\, \phi(u)\, \sin\,(ux)/\pi u$, $u \geq 0$, in (4.4). Since $|\sin\, ut| \leq 1$, it suffices to find a nonnegative monotone function $\beta(u)$ such that for prescribed truncation error ε

$$|\gamma(u)| \leq \frac{2|Re\, \phi(u)|}{\pi u} \leq \beta(u) \quad \text{for} \quad u \geq T \qquad \text{and} \qquad \int_T^{\infty} \beta(u)\, du \leq \varepsilon\ . \qquad (6.3)$$

From §2 we know that $|Re\, \phi(u)| \leq |\phi(u)| \leq 1$, but that leaves $\beta(u) = 2/\pi u$, which is not integrable. *Hence, to obtain useful bounds from (6.3), we must exploit additional properties of the cdf F and its cf ϕ.* (This is consistent with the discussion in §5 and will be illustrated by the examples.) The general principle is that $|\phi(u)|$ will decay faster when the cdf F is smoother. If the cdf F has a point mass, then $|\phi(u)|$ is of order 1 as $|u| \to \infty$, and convergence of the integral depends on the oscillations of the sine function to obtain convergence. (If F has only a few atoms, then we can often identify them and remove them before performing the inversion. For example, the M/G/1 waiting time cdf in (1.2) has a single atom at 0, which we remove before inversion in § 9.) When $|\phi(u)|$ does not decay quickly enough and we cannot eliminate the difficulty by removing a few atoms, we suggest using convolution smoothing or an alternative method of summation; see below.

As a consequence of the Riemann-Lebesgue lemma (p. 514 of Feller [43]), if the cdf F has n integrable derivatives, then $|\phi(u)| = o(|u|^{-n+1})$ as $|u| \rightarrow \infty$, so that for $n \geq 2$ it is certainly possible to apply (6.1)-(6.3). For cdf's on the positive real line, we have to consider the behavior at the origin. For example, the exponential cdf with mean m has only one integrable derivative, because the density is discontinuous at zero; then $\phi(u) = (1 - imu)^{-1}$ so that $Re(\phi(u) = (1 + m^2 u^2)^{-1} = O(|u|^{-2})$ as $|u| \rightarrow \infty$. More generally, if F is E_k (Erlang of order k), then $\phi(u) = (1 - imu)^{-k}$ and $Re(\phi)(u) = O(|u|^{-(k+1)})$ for k odd and $O(|u|^{-k})$ for k even. This analysis indicates that convolutions of many exponential distributions should be easy to calculate by numerical inversion; see §12.

Truncation Estimates

Since the sign of the integrand $\gamma(u)$ in (4.4) is oscillating, the bound in (6.3) is certainly a conservative estimate. Often a better *estimate* of the truncation error is simply $h\beta(Nh)$, the bound for the last term in the sum. Moreover, for an estimate we need not insist that β be a bound; we can let β represent the asymptotic behavior, i.e., $2\, Re(\phi)(u)/\pi u \sim \beta(u)$ as $u \rightarrow \infty$ (i.e., the ratio converges to 1 as $u \rightarrow \infty$). A natural refinement is to multiply this estimate by $(2/\pi)(\pi/2h|t|) = 1/h|t|$, provided that $1/h|t| \geq 1$. We choose this refinement, because we anticipate that the partial sums will oscillate around the infinite series because of the sine function. We want to account for consecutive summands with the same sign, so we multiply the first estimate by $\pi/2h|t|$ because that is approximately the number of terms in a quarter sine cycle for sin *tu* with step size *h*. At the beginning of the cycle, when the sine function first becomes positive, the value of the sum should be approximately at a minimum; in the middle of the cycle it should be approximately at a maximum; and at a quarter cycle it should be approximately at the correct value. We also multiply by $2/\pi$ because that is the average value of the sine function over the quarter cycle. Finally, we make this adjustment only when $1/h|t| \geq 1$ because of course one term is the minimum number of terms to consider. Thus, we obtain the *truncation error estimate*

$$e_t \equiv e_t(N, h, t) \approx h\beta(Nh) \max\{1, 1/h|t|\} = \max\{h\beta(Nh), \beta(Nh)/|t|\}, \qquad (6.4)$$

where $2\, Re(\phi)(u)/\pi u \sim \beta(u)$ as $u \to \infty$.

Sometimes it is convenient to have a single estimate independent of *t*, e.g., for computing $F(t)$ for several values of *t*. Assuming that interest is focused on tail probabilities, we suggest letting *t* be the mean of *F* or the mean plus a few standard deviations, both of which are scale invariant. (This usually leads to the error estimate $h\beta(Nh)$ in (6.4).) However, note that the estimated error in (6.4) grows as *t* becomes very small. For very small *t*, the bound in (6.3) may be a better estimate.

Remark (6.1)

The truncation error estimate in (6.4) is supported by related analysis with the inversion integral. If $Re\,(\phi)(u)$ is *nonincreasing*, then it is not difficult to show, using the periodic property of the sine function, that

$$\left| \int_T^{\infty} 2Re\,(\phi)(u)\, \frac{\sin\,(tu)}{\pi u}\, du \right| \leq \frac{2Re\,(\phi)(T)}{\pi t T} \approx \frac{\beta(Nh)}{t} \quad \text{as} \quad T \to \infty\,. \qquad (6.5)$$

The inequality holds because the worst (largest) case occurs when the sine function is beginning a positive half cycle at T and $Re\,(\phi)$ is constant over this positive half cycle, and 0 thereafter. Moreover, (6.4) is supported by the asymptotic behavior of the sine integral (5.2 of Abramowitz and Stegun [8]) if we can assume that

$$\left| \int_T^{\infty} 2\, Re(\phi)(u)\ \frac{\sin\,(tu)}{\pi u}\ du \right| \approx \left| \frac{2\, Re\phi(T)}{\pi} \right| \ \left| \int_T^{\infty} \frac{\sin\, tu}{u}\ du \right| , \qquad (6.6)$$

because

$$\left| \int_T^{\infty} \frac{\sin\, tu}{u}\ du \right| \sim \frac{1}{tT} \quad \text{as} \quad T \rightarrow \infty\ . \qquad (6.7)$$

Convolution Smoothing

Another approach to the truncation problem, which becomes important if the validity of the Poisson summation formula cannot be established, if bounding functions β are hard to find, or if the associated bounds and estimates yield too many terms in the sum, is *convolution smoothing*; i.e., to perturb the original cdf by convolving it with another cdf that has a suitably small variance and is very smooth; see p. 511 of Feller [43], Bohman [14], Silverberg [109] and Platzman et al. [99]. The small variance guarantees that the perturbed cdf differs only negligibly from the original cdf. Since the convolution cdf tends to inherit the smoothness of its smoothest component (p. 146 of Feller [43]), the convolution cdf tends to be much smoother than the original cdf, which is reflected by the final transform having a more rapidly decaying tail than the original transform. As a byproduct, it implies that all the Poisson summation formulas are rigorously justified.

Let F be the original cdf with cf ϕ; let F_{σ^2} be the *smoothing cdf* with mean 0, variance σ^2 and cf ϕ_{σ^2}; and let F_s be the *smoothed cdf*, i.e., the *convolution cdf*

$$F_s(t) \;=\; \int_{-\infty}^{\infty} F(t-y)\; dF_{\sigma^2}(y)\ . \tag{6.6}$$

By (2.3) and (2.4),

$$|\phi_s(u)| \;=\; |\phi(u)\phi_{\sigma^2}(u)| \;\le\; |\phi_{\sigma^2}(u)|\ , \tag{6.7}$$

so that the tail behavior of $|\phi_s(u)|$ is dominated by the tail behavior of $|\phi_{\sigma^2}(u)|$ alone, independent of the original cf ϕ.

From a probabilistic point of view, it is natural to let the smoothing cdf F_{σ^2} be the *normal cdf* with mean 0 and variance σ^2, because we have a good intuitive understanding of the effect of the perturbation and because the normal cf is easy to analyze, i.e., $\phi_{\sigma^2}\ (u) \;=\; e^{-u^2\sigma^2/2}$. In particular, the perturbation tends to be the order of a few standard deviations, i.e., of order $n\sigma$ where $1 \;<\; n \;<\; 8$, because then

$$F^c_{\sigma^2}\ (n\sigma) \;=\; F^c_1(n) \;=\; \begin{cases} \ \ 0.169 & ,\ n \;=\; 1 \\ \le 10^{-15} & ,\ n \;=\; 8. \end{cases} \tag{6.8}$$

In the case of a cdf F on the positive real line, we have $F_s(-t) \;\approx\; 0$ provided t is not in the $n\sigma$-neighborhood of 0. Then, from (5.10),

$$F_s(t) \approx F_s(t) \;-\; F_s(-x) \;=\; \frac{ht}{\pi} \;+\; \frac{2h}{\pi} \sum_{k=1}^{\infty} Re(\phi_s)(kh)\ \frac{\sin\ (kht)}{kh} \;-\; e_d\ .$$

$$\;=\; \frac{ht}{\pi} \;+\; \frac{2h}{\pi} \sum_{k=1}^{\infty} w(k)\ Re(\phi)(kh)\ \frac{\sin\ (kht)}{kh} \;-\; e_d\ , \tag{6.9}$$

where $w(k) \;=\; \phi_{\sigma^2}(kh) \;=\; e^{\frac{-\sigma^2 k^2 h^2}{2}}$ is a decreasing *weighting function*. We can construct a bounding integrand β as in (6.1)-(6.3) by focusing entirely on the smoothing cf (the weighting function), i.e.,

$$|\gamma_s(u)| \equiv \left| 2\, Re(\phi_s)(u)\ \frac{\sin ut}{\pi u} \right| \leq \frac{2\,|\phi_{\sigma^2}(u)|}{\pi u} = \frac{2\, e^{-\frac{u^2\sigma^2}{2}}}{\pi u} \equiv \beta_s(u)\ . \tag{6.10}$$

We obtain an associated truncation *bound* T (assuming $\sigma T \geq 1$ and using standard inequalities for normal tail probabilities) via

$$\int_T^\infty \beta_s(u)\, du \leq \int_T^\infty \frac{2}{\pi T}\, e^{-\frac{u^2\sigma^2}{2}}\, du \leq 2(2/\pi)^{1/2}\, F^c_{\sigma^{-2}}(T) = 2(2/\pi)^{1/2}\, F^c_1(\sigma T)$$

$$\leq \frac{2e^{-\frac{\sigma^2 T^2}{2}}}{\pi\sigma T} \leq e^{-\frac{\sigma^2 T^2}{2}}\ . \tag{6.11}$$

Hence, to achieve a truncation error bound of ε for the smoothed cdf F_s, independent of the original cdf F, it suffices to have

$$\varepsilon = e^{-\frac{\sigma^2 T^2}{2}} \quad \text{or} \quad T = \frac{\sqrt{-2\log\varepsilon}}{\sigma}\ , \tag{6.12}$$

provided that $\varepsilon \leq e^{-1/2} \leq 0.61$, so that $\sigma T \geq 1$, as assumed.

Hence, T is of order σ^{-1}. Indeed, T ranges from $3.0\ \sigma^{-1}$ to $6.1\ \sigma^{-1}$ as ε decreases from 10^{-2} to 10^{-8}. Unfortunately, we cannot choose the standard deviation σ of the perturbation arbitrarily small without suffering a severe penalty in required computation. Convolution smoothing can only be useful if nonnegligible perturbation can be permitted.

In the case of a cdf F on the whole line, we apply (5.14) to obtain

$$F_s(t) \approx F_s(t) - F_s(t-y) = \frac{hy}{2\pi} + \frac{h}{\pi} \sum_{k=1}^{\infty} g_s(kh) - e_d\ , \tag{6.13}$$

where

$$|g_s(u)| = \left| \frac{Re\,(\phi_s)(u)}{\pi u}\,[\sin u(y-t) + \sin ut] + \frac{Im\,(\phi_s)(u)}{\pi u}\,[\cos u(y-t) - cus\ ut] \right|$$

$$\leq \frac{2}{\pi u}\,(|Re\,\phi_s(u)| + |Im\,\phi_s(u)|) \leq \frac{4}{\pi u}\,|\phi_s(u)|$$

$$\leq \frac{4}{\pi u}\,|\phi_{\sigma^2}(u)| = \frac{4e^{-\frac{u^2\sigma^2}{2}}}{\pi u} \equiv \beta_s(u)\ . \tag{6.14}$$

The bounding function β_s in (6.14) is exactly two times the bounding function in (6.10). Hence, instead of (6.12), here $T = \sqrt{-2\log(\varepsilon/2)}/\sigma$.

Remarks (6.2) If convolution smoothing is to be performed, then it is natural to determine the truncation point T first, based on the permissible perturbation σ and truncation error, and then afterwards determine the appropriate step size h to achieve the desired discretization error for the smoothed cdf F_s. The tail probabilities of F_s are not much harder to analyze than the tail probabilities of F because F_s is the cdf of the sum of two independent random variables, one of which is very small. However, there should be no problem if the discretization step size h is determined before convolution smoothing, because the deviation in the discretization error analysis due to convolution smoothing is typically negligible.

(6.3) To get a rough idea about the way convolution smoothing alters the original cdf, we can approximate using Taylor's series. The convolution smoothing yields an approximate cdf something like

$$F_a(t) \approx F(t \pm \sigma) \approx F(t) \pm f(t)\sigma\ , \tag{6.15}$$

where $f(t)$ is the density, so that the error introduced at t by smoothing is of order $f(t)\sigma$. Since the failure rate of a cdf F on the positive real line is $r(t) = f(t)/F^c(t)$, the approximate error at t due to σ in that case can also be expressed as $F^c(t)\ r(t)\sigma$. If $r(t) \approx m_1^{-1}$ where m_1 is the mean of F (equality holds for an exponential cdf), then the error at t is of order $F^c(t)\sigma/m_1$.

(6.4) The algorithm here for a cdf on the whole line, using convolution smoothing with a normal smoothing cdf together with (5.14), is essentially the same as the algorithm proposed by Platzman et al. [99], except they developed a precise bound on the error. The algorithm in [99] specifies a discretization step size h, a standard deviation σ and a number of terms N in order that the approximating cdf F_a be related to the original cdf F by

$$F(t-\varepsilon_1) - \varepsilon_2 \le F_a(t) \le F(t+\varepsilon_1) + \varepsilon_2 \tag{6.16}$$

for given t, ε_1, ε_2, u_ε and l_ε, where $F(l_\varepsilon) \le \varepsilon_2/4$ and $F^c(u_\varepsilon) \le \varepsilon_2/4$. The key idea in (6.16) is to work with the two different errors, ε_1 and ε_2. Note that (6.16) is a modification of the Lévy metric (p. 285 of Feller [43]), i.e., (6.16) is a well known way to measure the distance between cdf's. To achieve (6.16) we let

$$h = \frac{2\pi}{u_\varepsilon - l_\varepsilon + 2\varepsilon_1}, \quad \sigma = \frac{\varepsilon_1}{\sqrt{2\log(2/\varepsilon_2)}}, \tag{6.17}$$

$$T = Nh = \frac{\sqrt{2\log(2/\varepsilon_2)}}{\sigma} = \frac{2\log(2/\varepsilon_2)}{\varepsilon_1}. \tag{6.18}$$

In this case, the weighting function in (6.9) is

$$w(k, N) \equiv w(k) = \phi_{\sigma^2}(kh) = e^{\frac{-\sigma^2 k^2 h^2}{2}} = \left[\frac{\varepsilon_2}{2}\right]^{\left[\frac{k}{N}\right]^2}, \quad 1 \le k \le N, \tag{6.19}$$

with $w(k, N) = 0$ for $k > N$. Note that h, ε_1 and σ appear in the final formula of (6.19) only through ε_2 and N. Note that the expression for h in (6.17) is essentially the same as in (5.18) and the expression for T as a function of ε_2 and σ in (6.18) is identical to what we derived above. Hence, the error bound (6.16) is achieved in the context of our algorithm with convolution smoothing using a normal smoothing cdf essentially by just defining the normal standard deviation σ as in (6.17). Of course, it remains to interpret (6.16), for which Remark 6.3 can be applied with ε_1 replacing σ. Since this variant of the Fourier-series method based on

(6.16)–(6.19) comes from Platzman et al. [99], we call it POISSON-PAB.

(6.5) From the discussion above, it is apparent that convolution smoothing can help establish a better computation from the perspective of the bounds and estimates for the truncation error, but it may not be apparent that convolution smoothing can help improve the computation itself. Indeed, as indicated by Remark 6.3, convolution smoothing for given h and N can seriously reduce the accuracy of the computed $F(t)$ for some t, but it also can improve the accuracy too, as we will show in the examples.

(6.6) There are many variants of the convolution smoothing method. The result is to insert weights in the series as in (6.9), which is closely related to the alternate methods of summing Fourier-series, to be discussed below. In signal processing the use of weights is called windowing and the weighting functions are called windows; see Harris [53] and pp. 88-105 of Rabiner and Gold [102]. Extensive experience indicates that windowing, judiciously applied, can significantly improve the accuracy of the computation. The advantage of convolution smoothing with the normal cdf is that we can have a good intuitive understanding of the effect of the smoothing on the original cdf as well as precise bounds such as (6.16), but other windows are typically preferred.

Alternative Methods of Summation

As remarked in §5, the sequence of partial sums, say $\{s_n(t) : n \geq 1\}$, of the Fourier-series of a continuous periodic function $f_p(t)$ need not converge pointwise to $f_p(t)$; (see p. 153 of Natanson [91]), but it can always be forced to converge uniformly to f_p if we change the method of summation. The classic result in this direction is Fejer's theorem, which concludes that the arithmetic mean (average) of the first n partial sums, say $\sigma_n(t)$, converges uniformly to $f_p(t)$ as $n \to \infty$ see p. 157 of [91]). Thus, it is natural to consider $\sigma_n(t)$ instead of $s_n(t)$ as an approximation to $f_p(t)$. It is easy to see that the approximation $\sigma_n(t)$ is realized in the context of

(6.9) simply by using the weighting function

$$w(k, n) = \begin{cases} (n - k)/n & , 1 \le k \le n \\ 0 & , k > n , \end{cases} \tag{6.20}$$

which is known as the triangular, Fejer or Bartlett window; see p. 59 of Harris [53]. We can also obtain (6.20) if we use convolution smoothing with the smoothing density

$$f_{\sigma^2}(t) = (1 - \cos Nht)/\pi Nht^2 , \quad -\infty < t < \infty ; \tag{6.21}$$

see cf No. 5 in Table 1 on p. 503 of Feller [43]. We obtain (6.21) from (6.20) by noting that (6.20) and (6.21) essentially form a Fourier transform pair (pp. 1, 52, 63, 88 of Champeney [20]) after we consider the symmetric version of (6.20), with $w(k, n) = w(-k, n)$ for $k \le -1$, which is justified because the integral in (3.6) was originally symmetric about 0 over the entire real line.

From the discussion above, it should be clear that convolution smoothing, windowing and alternative methods of summation are essentially equivalent. For example, the standard approximation $s_n(t)$ can be obtained from the full series by applying the rectangular or Dirichlet window, see p. 58 of Harris [53],

$$w(k, n) = \begin{cases} 1 & , 1 \le k \le n \\ 0 & , k > n , \end{cases} \tag{6.22}$$

which corresponds to convolution smoothing with the Dirichlet ‘‘density’’ $\sin (nht)/\pi t$, which unlike (6.21) is *not* a probability density. (It integrates to 1, but it is not nonnegative.)

Remark (6.7) The convolution smoothing representations for $s_n(t)$ and $\sigma_n(t)$ above should be distinguished from the classical integral formulas, which apply to the aliased periodic function f_p, i.e.,

$$s_n(t) = \frac{h}{\pi} \int_{-\pi/h}^{\pi/h} f_p(t+u) \frac{\sin\left[n + \frac{1}{2}\right] uh}{2 \sin(uh/2)} du$$

$$\sigma_n(t) = \frac{h}{\pi n} \int_{-\pi/h}^{\pi/h} f_p(t+u) \frac{\sin^2(nuh/2)}{2\sin^2(uh/2)} du \tag{6.23}$$

see pp. 35, 73, 158 of Tolstov [119] or pp. 151, 158 of Natanson [91]. ■.

From the point of view of summability, $\sigma_n(t)$ or the Fejer window in (6.20) is always superior to $s_n(t)$ or the rectangular window in (6.22). However, from the point of view of approximation, $\sigma_n(t)$ has the drawback of converging slowly. The rate of convergence of $\sigma_n(t)$ tends to be about the same for all functions as for the worst case, namely, of order $1/n$. For example, for the function $f_p(t) = 1 - \cos t$, $s_n(t)$ is exact for $n \geq 1$, whereas $\sigma_n(0) = n^{-1}$; see pp. 161-167 of Natanson [91]. In contrast, the rate of convergence of s_n to f_p tends to improve dramatically as the structure of f_p improves. Indeed, the uniform distance between s_n and f_p is bounded above by $(3 + \log n)$ times the uniform distance between f_p and the best approximating n-term trigonometric polynomial; see pp. 84, 88, 152 of [91].

An attractive alternative to the two methods of summation discussed above is the de la Vallée-Poussin method; with it the n^{th} term, say $\tau_n(t)$, is the arithmetic mean of the last $n/2$ of the first n partial sums. For n even, it is related to $\sigma_n(t)$ by

$$\tau_n(t) = 2\,\sigma_n(t) - \sigma_{n/2}(t)\ . \tag{6.24}$$

The associated de la Vallée-Poussin window is

$$w(k,n) = \begin{cases} 1 & ,\ 1 \leq k \leq n/2 \\ 2(n-k)/n & ,\ n/2 + 1 \leq k \leq n \\ 0 & ,\ k > n\ . \end{cases} \tag{6.25}$$

The uniform distance between f_p and τ_n is bounded above by 4 times the uniform distance between f_p and the best approximating $n/2$-term trigonometric polynomial.

The simple partial sums $s_n(t)$ have the advantage that error bounds and estimates are relatively easy to obtain for them, as described at the beginning of this section. Convolution smoothing with the normal distribution as in (6.9) is appealing because it is relatively easy to interpret and because it leads to the error bounds in (6.16). However, it is clear that many other variants can be considered.

Exploiting Properties of Alternating Series

In the algorithm EULER displayed in §1 we use an inversion formula for Laplace transforms in (4.6) and (5.27) that might have the form of an alternating series, because the trigonometric terms were forced to assume the form $(-1)^k$. Thus, for this series it is natural to apply an alternative method of summation that is especially well suited to alternating series. This approach was originally proposed by Simon, Stroot and Weiss [110]. They obtained the nearly alternating series in (4.6) from (21) of Dubner and Abate [40]. The specific alternative form of summation proposed was Euler summation. This same approach was also later proposed by Hosono [58], [59], [60] in the context of (3.3a), (4.6a) and (5.30).

Euler summation is usually defined in terms of finite differences, see pp. 158, 230 of Davis and Rabinowitz [32] and Johnsonbaugh [70], but for an alternating series it is 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, this does not seem well known.) In particular, if we apply the Euler sum to *m* terms after an initial *n* terms, then we understand the Euler sum, as a function of *n* and *m*, to be

$$E(t, m, n) = \sum_{k=0}^{m} \binom{m}{k} 2^{-m} s_{n+k}(t) \;, \tag{6.26}$$

where $s_n(t) \equiv \sum_{k=0}^{n} (-1)^k a_k(t)$ is the n^{th} partial sum as before. Paralleling (6.9), (6.19), (6.20), (6.22) and (6.25), Euler summation can be represented via a weighting function, in particular,

$(-1)^k a_k(t)$ should be weighted by

$$w(k, m, n) = \begin{cases} 1 & 1 \le k \le n \,, \\ \sum_{j=0}^{l} \begin{bmatrix} m \\ j \end{bmatrix} 2^{-m} \,, & k = n + m - l \,, \quad 0 \le l \le m - 1 \,, \end{cases} \tag{6.27}$$

This form is used by Hosono [58], [59], [60].

The Euler sum (6.26) can also be described as repeated pairwise averaging. That is, we start with the m successive partial sums $s_n(t), s_{n+1}(t), \ldots, s_{n+m-1}(t)$ and construct the $(m-1)$ averages of adjacent pairs. Then we repeat the process, constructing the $(m-2)$ averages of adjacent pairs of these averages. We continue this way until we have a single average. The result is (6.26).

More conventionally, the Euler sum is defined in terms of the difference operator Δ. With $\{a_n : n \ge 1\}$ being a sequence, let $\Delta a_n = a_{n+1} - a_n$ and let Δ^k be the k-fold iterate; see Johnsonbaugh [70] (who uses the negative of the standard definition above). If we have an alternating series (the k^{th} term is $(-1)^k a_k$ for $a_k \ge 0$), then (6.26) is equivalent to the conventional definition

$$E(t, m, n) = s_n(t) + (-1)^{n+1} \sum_{k=0}^{m-1} (-1)^k 2^{-(k+1)} \Delta^k a_{n+1}(t) \,. \tag{6.28}$$

The idealized situation involves partial sums $s_n = \sum_{k=0}^{n} (-1)^k a_k$, where $a_k \ge a_{k+1} \ge 0$ for all k (which is what we mean by an alternating series); see Johnsonbaugh [70]. Then $s_n \to s_\infty$ as $n \to \infty$,

$$s_{2n+1} \le s_\infty \le s_{2n} \quad \text{for all } n \,. \tag{6.29}$$

Moreover, if $\Delta a_n = a_{n+1} - a_n$ is also increasing in n, then

$$|s_n - s_\infty| \le \frac{a_n}{2} \ ; \tag{6.30}$$

see Theorem 1, p. 638, of [70].

Let $E(m, n)$ be the Euler sum associated with $\{a_n\}$. Note that

$$\begin{aligned}(-1)^m \Delta^m a_{n+k} &= \sum_{j=0}^{m} (-1)^j \binom{m}{j} a_{n+k+j} \\ &= 2^m \left| E(m, n+k-1) - E(m-1, n+k-1) \right| \ . \end{aligned} \tag{6.31}$$

It turns out that if $(-1)^m \Delta^m a_{n+k}$ is decreasing in k for $k \ge 1$, then

$$|E(m, n) - s_\infty| \le \frac{\Delta^m a_{n+1}}{2^m} = |E(m, n) - E(m-1, n)| \ ; \tag{6.32}$$

see p. 643 of [70]. However, the condition for (6.32) typically is not verifiable, so that we do not regard (6.32) as a useful bound. Table 2 of [70] shows that Euler summation often works remarkably well for alternating series. Indeed, the accuracy is typically orders of magnitude greater than the bounds in (6.29) and (6.30).

In order for Euler summation to be effective, we would like to have $a_k \equiv Re(\hat{f})((A + 2k\pi i)/2t)$ in (4.6) have three properties for sufficiently large k: (1) to be of constant sign, (2) to be monotone, and (3) to have the higher-order differences $(-1)^m \Delta^m a_{n+k}$ be monotone, as required for (6.32). In applications we will usually not check these properties, and even if they hold, we will not know when k is large enough, so that our use of Euler summation in EULER is a heuristic without guarantees.

However, we can cite some supporting theory to show that the desired properties can be expected, at least approximately, when the underlying function has appropriate smoothness. Note that

$$Re(\hat{f})(u + iv) = \int_0^\infty \cos vt \, e^{-ut} f(t) \, dt \; . \tag{6.33}$$

Assuming that $f(t)$ is twice continuous differentiable, so is the damped function $g(t) \equiv e^{-ut} f(t)$. Assuming in addition that $g(\infty) = g'(\infty) = 0$ (which occurs when $f(\infty) = f'(\infty) = 0$), we can apply integration by parts twice to obtain

$$Re(\hat{f})(u + iv) = \frac{u - f'(0)}{v^2} - \frac{1}{v^2} \int_0^\infty \cos vt \; g''(t) \; dt \; . \tag{6.34}$$

Assuming further that $g''(t)$ is integrable, we can apply the Riemann-Lebesgue lemma (p. 514 of Feller [43]) to deduce that

$$Re(\hat{f})(u + iv) = \frac{u - f'(0)}{v^2} + o\left[\frac{1}{v^2}\right] \quad \text{as} \quad v \to \infty \; . \tag{6.35}$$

Moreover, with additional smoothness and integrability, we can apply the argument above one more time to replace the $o(v^{-2})$ term in (6.35) by an $o(v^{-3})$ term, or we can apply it twice more to obtain an $O(v^{-4})$ term.

As a consequence of (6.35), $Re(\hat{f})(u + iv)$ is indeed eventually positive provided that $u > f'(0)$ and the other conditions hold. Moreover, then the asymptotic approximating function $[u - f'(0)]/v^2$ has all three properties desired for Euler summation. (In our application to cdf's $u > 0$ and $f(t) = F^c(t)$, so that indeed $u > f'(0)$.)

Remark (6.8) To see that the desired properties supporting Euler summation need not hold in general, consider the cdf of a point mass at *x*,

$$\hat{F}^c(s) = \frac{1 - e^{-sx}}{s} \; , \tag{6.36}$$

so that

$$Re(\hat{F}^c)(u + iv) = \frac{u(1 - e^{-ux} \cos vx) + v \sin vx}{u^2 + v^2} \tag{6.37}$$

and

$$Im(\hat{F}^c)(u+iv) = \frac{u \sin vx - v(1 - e^{-ux} \cos vx)}{u^2 + v^2} \ . \qquad (6.38)$$

From (6.37), we see that, for every u, there is no v_0 such that $Re(\hat{F}^c)(u+iv)$ is of constant sign for all $v > v_0$. From (6.38), we see that the imaginary part is better behaved, because for any nonnegative n, $Im(\hat{F}^c)(u+iv)$ is eventually negative for all sufficiently large v. However, $Im(\hat{F}^c)(u+iv)$ is not nearly monotone in v for large v because of the cos vx term. In §14 we show that the inversion formula (5.30) using the imaginary part (6.38) fares no better on (6.36) than (5.27) using the real part (6.37).

(6.9) To see that Euler summation does not perform well on *all* alternating series, it is instructive to consider the series

$$\sum_{k=1}^{\infty} (-1)^k (\sin kx)^2 / k = \frac{\log(1/\cos x)}{2}$$

for $x < \pi/2$. The performance of Euler summation degrades seriously as x increases. For example, the error in $E(x, 11, 30)$ is of order 10^{-9}, 10^{-3} and 10^{-1} for $x = 0.1$, 1.0 and 1.5, respectively. For $x = 1.5$, s_{41} had an error of order 10^{-2}, which is better. ■

We suggest $|E(t, m, n) - E(t, m-1, n)|$ and

$$|E(t, m, n+1) - E(t, m, n)| = \left| \sum_{k=0}^{m} 2^{-m} \binom{m}{k} a_{n+k+1}(t) \right| \qquad (6.39)$$

as *estimates* of the truncation error associated with Euler summation. Formula (6.39) is also used by Hosono [58], [59], [60]. Experience indicates that these error estimates are often good, but not always so; e.g., see §11.

Remark (6.10). From the t in the denominator of $Re(\hat{f})$ in (4.6) or (5.27), we may anticipate that the appropriate value of n in the algorithm EULER might increase with t and this is sometimes the case. According to Hosono [60], the value of $n(t)$ producing prescribed accuracy is often approximately a linear function of t. Thus, on p. 58 of [60] Hosono suggests estimating this linear function by considering two time points, using the error estimate (6.39), whose explicit form is due to Hosono, before performing many runs at other time points.

7. Summaries of Variants of the Fourier-Series Method

We now indicate how the previous sections can be combined to obtain full algorithms. These algorithms are all variants of the Fourier-series method. We first describe EULER, the alternating-series approach for inverting Laplace transforms exploiting Euler summation, which is displayed in §1. It is an automatic algorithm but without guaranteed accuracy. Then we describe the algorithm POISSON with associated error bounds and estimates. We do not further discuss the algorithm LATTICE-POISSON for the special case of lattice distributions, which was already adequately treated in §5.

The Algorithm EULER for Laplace Transforms

If we have a function on the positive half line, then we can use Laplace transforms and apply the Bromwich inversion integral in (3.3). In this variant we use numerical integration with the trapezoidal rule so that the cosine terms become $(-1)^k$ and obtain the (approximately) alternating series in (4.6) or (5.27). Next we use (5.28) to bound the discretization error. When the function is a cdf, the bound is approximately e^{-A}, so we choose the parameter A to make this suitably small. For example, $A = 19.1$ ensures a discretization error less than 10^{-7}. As discussed in Remark 5.8, accuracy to $10^{-\gamma}$ typically requires at least $3\gamma/2$-digit precision. To meet the needs of queueing applications without requiring more than 14-digit precision, we usually aim for an accuracy of 10^{-7} or 10^{-8}. (Note that this typically means double precision.) A high-precision language like UBASIC is useful to treat difficult cases, but it is typically not necessary.

Finally, we use Euler summation as in (6.26) or (6.27) to approximately calculate the resulting infinite series. We have used $m = 11$ for the binomial parameter. For a cdf F, we actually work with the Laplace transform $\hat{F}^c(s) \equiv \int_0^\infty e^{-st} F^c(t)\,dt$ of the complementary cdf $F^c(t) = 1 - F(t)$. (We also remove all known atoms in the cdf.) We make this choice, because intuitively the terms in the series then should be closer to n^{-1} for which Euler summation is known to be good. (Little difference was seen in the examples, however.) We then estimate the error from using $n + m$ terms by the difference of successive terms in n, i.e., by $|E(t, m, n) - E(t, m, n+1)|$ and stop when the estimated error is suitably small. Unfortunately, however, this error estimate is *not a bound,* so that accuracy is not guaranteed. We thus use a second method such as one of the methods in §8 as a check. The performance of EULER as n is changed to $2n$ and $4n$ is also a good practical check.

The algorithm EULER is displayed in §1 for the M/G/1 waiting-time complementary cdf in Example 9.2 (with the known atom at zero removed). The program EULER starts in lines 10-13 by defining two vectors SU and C. Since the parameter m has been set at 11, there are 12 binomial coefficients. The terms $SU(k)$ are the last 13 partial sums $s_n(t)$ of the infinite series in (5.27). The terms $C(k)$ are the 12 binomial coefficients needed to compute the Euler sum (6.26).

In lines 20-22 the desired time value t is input and the parameters A in (5.27) and n in (6.26) are set at 19.1 and 15, respectively. The parameter A is adjusted to control the discretization error using (5.28). As indicated above, the value $A = 19.1$ ensures a discretization error of at most 10^{-7} for a cdf. The number n of terms in (6.26) is increased until the estimated error $|E(t, m, n) - E(t, m, n+1)|$ is suitably small. Our experience indicates that $n = 15 - 20$ is typically sufficient to obtain accuracy to 10^{-7}, so that $n = 100$ often produces a convenient conservative initial setting.

Lines 23-25 contain other convenient parameters for calculating the partial sums $s_n(t)$ from

(5.27). The transform to be inverted and the summands in (5.27) are defined in lines 90-97, using (1.1) and the Laplace transform of the service time distribution: $Gs \equiv \hat{f}(s) = (1 + 2s)^{1/2}$; see Example 9.2 below. We first eliminate the known atom of $1 - \rho$ at the origin; i.e., since $W(t) = 1 - \rho + \rho F(t)$, we focus on $F(t)$. We numerically invert the Laplace transform of $F^c(t) \equiv 1 - F(t)$, i.e.,

$$Fs \equiv \hat{F}^c(s) \equiv \int_0^\infty e^{-st} F^c(t)\, dt = \frac{1 - \hat{G}_e(s)}{s(1 - \rho \hat{G}_e(s))} \ , \tag{7.1}$$

where $\hat{G}_e(s) = [1 - \hat{G}(s)]/\tau s$; see lines 94 and 95 in the displayed program.

The partial sums $s_n(t)$ are computed in lines 30-33; the vector of the last 13 partial sums are formed in lines 40-45; and the Euler sums $E(t, m, n)$ and $E(t, m, n + 1)$ in (6.26) for $m = 11$ and $n = 15$ are calculated in lines 50-54. The error estimate $|E(t, m, n) - E(t, m, n + 1)|$ is computed in line 60. Finally, the output is printed in lines 70-75. The output contains the value of t, the estimated function value $F^c(t)$ and the estimated error.

The Algorithm POISSON with Asymptotic Estimates

We now suppose that we are trying to evaluate a cdf F on the positive half line at a point t with an allowable error ε. (If F is defined on the whole line, then we shift F to the right and apply the same method, as described in §5. We consider successive shifts to check that an appropriate shift has been made.)

For a cdf on the positive half line, the inversion integral is (3.6) and the trapezoidal rule approximation is (4.4). We think of (4.4) as the algorithm POISSON. It remains to specify the discretization step size h and the number of terms N in (4.4). Appropriate values of h and N can easily be found by trial and error, but these can also be determined by bounds or estimates, for which we apply §5 and §6.

What we regard as the standard approach is to choose h and N based on *estimates* of the errors using asymptotic analysis. From the point of view of a pure numerical procedure, this asymptotic analysis is a severe requirement, but we regard the asymptotic analysis as desirable in its own right in order to understand the behavior of the model. Given that we would do the asymptotic analysis anyway, it is natural to exploit it to control and understand the computation.

The first step is to use the transform to determine the asymptotic behavior of $F^c(t)$ as $t \to \infty$; i.e., we identify a convenient decreasing function $\alpha(t)$ such that $F^c(t) \sim \alpha(t)$ as $t \to \infty$ $(F^c(t)/\alpha(t) \to 1)$. The asymptotic behavior of $F^c(t)$ as $t \to \infty$ can be determined from the behavior of the Laplace transform of $F^c(t)$ near its singularity with largest real part. The asymptotic behavior depends on the nature of the singularity, whether the singularity is a simple pole or a branch point; see Sections 35 and 37 of Doetsch [36]. These two cases are each illustrated by two examples in § 9-11. Alternatively, bounds and estimates for the tail probabilities may be obtained by probabilistic arguments.

Given the function α, we let $v_{\varepsilon/2}$ be such that $\alpha(v_{\varepsilon/2}) = \varepsilon/2$ and apply the discretization error bound in (5.12) to obtain the estimate

$$e_d \le F^c \left[\frac{2\pi}{h} - t \right] \approx \alpha \left[\frac{2\pi}{h} - t \right] . \tag{7.2}$$

For an estimated discretization error of $\varepsilon/2$, we let $(2\pi/h) - t = v_{\varepsilon/2}$, so that the estimated step size is $h \approx 2\pi/(v_{\varepsilon/2} + t)$. To calculate $F(t)$ for t throughout the interval $[0, \pi/h]$, we let $t = \pi/h$ in (7.2) and obtain $h \approx \pi/v_{\varepsilon/2}$.

To estimate N given h, we determine the asymptotic behavior of the integrand $\gamma(u)$ in (4.4); i.e., we identify convenient functions β and ξ such that $|\gamma(u)| \le \beta(u)$ and $\beta(u) \sim \xi(u)$ as $u \to \infty$. Then we apply (6.4) and let $T_{\varepsilon/2}$ be such that $\max\{h\xi(T_{\varepsilon/2}),\ \xi(T_{\varepsilon/2})/|t|\} = \varepsilon/2$ and obtain the estimate $N \approx T_{\varepsilon/2} h^{-1}$.

Having determined h and N, we calculate the finite sum in (4.4), performing successive refinements to provide an *accuracy check*. In particular, we calculate the sum with $N/4$, $N/2$, N, $2N$ and $4N$ terms; then we repeat these calculations with h replaced by $h/2$ and $h/4$. If the changes going first through the pairs (h, N), $(h, 2N)$ and $(h, 4N)$ and then through the pairs (h, N), $(h/2, 2N)$, $(h/4, 4N)$ are negligible (of order ε or less), then we have a satisfactory accuracy check. If the changes are not suitably small, then we can consider further refinements or convolution smoothing.

As a further refinement, we can calculate the average of the last half of the partial sums in order to obtain the de la Vallée-Poussin approximation described in (6.24) and (6.25). We call this variant of the Fourier-series method POISSON-POUSSIN.

In calculating the finite sum, we use double precision to ensure that the roundoff error is negligible. Assuming that the accuracy of each number is of order 10^{-14} and $N \leq 10^6$, the roundoff error will be less than 10^{-8}. (Unlike EULER and LATTICE-POISSON, with algorithm POISSON there is no damping and no multiplication of the finite sum in (4.4) by a large number. Hence, the roundoff error analysis is different from Remark 5.8. Since $|Re(\phi)(kh)\sin(kht)| \leq 1$ in (4.4), all terms in the finite sum are less than 1 in absolute value, so that m-decimal-place (or m-digit) accuracy on the machine will lead to approximately $(m - \log_{10} N)$-decimal-place final accuracy.)

The Algorithm POISSON With Bounds

Of course, the procedures in the algorithm POISSON with estimates that we have just described do not yield bounds. When we set h by (7.1), the discretization error is *bounded* by $\varepsilon/2$ if, instead of the asymptotic relation between F^c and α, we establish that $F^c(t) \leq \alpha(t)$ for $t \geq t_0$ for some suitable t_0. Bounds on tail probabilities can often be obtained, but finding good bounds is much less routine than doing the asymptotic analysis. Relatively crude bounds, such as can readily be obtained from the moments and Chebychev's inequality, tend to be much too conservative, e.g., see (9.8)-(9.10) below.

Similarly, the truncation point can yield a bound if we let $T_{\varepsilon/2}$ be such that $\int_{T_{\varepsilon/2}}^{\infty} \beta(u)\, du = \varepsilon/2$, as in (6.3). However, our experience indicates that this bound is also very conservative.

Finally, if the truncation point bound $T_{\varepsilon/2}$ is too large, then we consider convolution smoothing as in §6. The variant POISSON-PAB due to Platzman et al. [99] discussed in Remark 6.4 yields a bound, but it often requires a big computation, unless ε_1 need not be small. Since convolution smoothing actually tends to improve the accuracy of the computation (see Sections 10 and 12 below), it is reasonable to use it. Moreover, in many practical applications, ε_1 in (6.16) need not be too small.

8. Methods Based on the Post-Widder Formula

We now present alternative inversion procedures for Laplace transforms to use in addition to one of the previous methods in order to provide a check; other candidates are mentioned in §15. For diversity, we choose procedures based on the Post [101]–Widder [125] inversion formula, which involves differentiation instead of integration. In particular, if $\hat{f}(s) = \int_0^\infty e^{-st} f(t)\, dt$ with f satisfying appropriate regularity conditions, then $f_n(t) \to f(t)$ as $n \to \infty$, where

$$f_n(t) = \frac{(-1)^n}{n!} \left[\frac{n+1}{t} \right]^{n+1} \hat{f}^{(n)}((n+1)/t) \;, \tag{8.1}$$

and $\hat{f}^{(n)}$ is the n^{th} derivative of $\hat{f}$.

The convergence of $f_n(t)$ in (8.1) to $f(t)$ is easy to understand from the probabilistic proof given on p. 333 of Feller [43]. 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 a point mass at t, so that $E[f(X_{n,t})] \to f(t)$ as $n \to \infty$ for all bounded real-valued f that are continuous at t (and other f as well).

The Gaver-Stehfest Method

Since the Gaver [49] – Stehfest [113] method has a probabilistic derivation, it should be relatively easy to understand. Just like EULER in §1, the Gaver-Stehfest method is also easy to program in less than 50 lines, as demonstrated by the display below. However, the Gaver-Stehfest method is much less robust than the Fourier-series method. For many problems, it works very well, but for others it does not. Also there is not yet any error analysis. Thus, the Gaver-Stehfest method illustrates the difficulties that are encountered by some numerical inversion methods.

As a basis for performing numerical inversion, Gaver [49] established a discrete analog of (8.1) involving finite differences. Let $\Delta \hat{f}(n\alpha) = \hat{f}((n+1)\alpha) - \hat{f}(n\alpha)$ and let $\Delta^k = \Delta(\Delta^{k-1})$, so that

$$(-1)^n \Delta^n \hat{f}(n\alpha) = \sum_{k=0}^{n} (-1)^k \binom{n}{k} \hat{f}((n+k)\alpha) \ . \tag{8.2}$$

Here is Gaver's result in [49].

Proposition 8.1. If f is a bounded real-valued function that is continuous at t, then

$$f(t) = \lim_{n \to \infty} \tilde{f}_n(t) \ ,$$

where

$$\tilde{f}_n(t) = (-1)^n \frac{ln 2}{t} \frac{(2n)!}{n!(n-1)!} \Delta^n \hat{f}(n \ln 2/t) \ . \tag{8.3}$$

Proof. As in the proof of the Post-Widder formula on p. 233 of Feller [43], we can apply the weak law of large numbers. Let X_n have a beta density with parameters n and $n+1$, i.e., let X_n

have probability density

$$g_{X_n}(y) \;=\; \frac{(2n)!}{(n-1)!n!}\, y^{n-1}(1\,-\,y)^n \;,\; 0 \le y \le 1 \;;$$

see p. 50 of [43]. Since the mean of X_n is $n/(2n+1)$ and the variance of X_n is $n(n+1)/(2n+1)^2(2n+2)$, $X_n \xrightarrow{p} 1/2$ as $n \to \infty$, where $\xrightarrow{p}$ denotes convergence in probability, by the weak law of large numbers. By the continuous mapping theorem, $-\alpha^{-1}\ln X_n \xrightarrow{p} \ln 2/\alpha$ as $n \to \infty$, where $-\alpha^{-1} \ln X_n$ has probability mass function

$$g_n(y) \;=\; \alpha\frac{(2n)!}{(n-1)!n!}(e^{-\alpha y})^n(1\,-\,e^{-\alpha y})^n \;,\; y \ge 0 \;.$$

Then $E[h(-\alpha \ln X_n)] \to h(\ln 2/\alpha)$ for any bounded real-valued h that is continuous at t. Since

$$E[\,f(-\alpha \ln X_n)] \;=\; \frac{\alpha(2n)!}{(n\,-\,1)!n!}(-1)^n\Delta^n\hat{f}(n\alpha) \;\;,$$

we have the desired result if we let $\alpha \;=\; \ln 2/t$. ■

Since the arguments of $\hat{f}$ are $k \ln 2/t$, they are real, so that $\tilde{f}_n(t)$ is a real-valued function of a real variable. In view of the factorials in (8.2) and (8.3), we can not let n be very large before we get significant roundoff problems. Hence, even more than in §6, it is desirable to consider alternative methods of summation to accelerate convergence.

As in §6, we accelerate convergence by using a linear combination of the terms, i.e., $\sum_{k=1}^{n} w(k,\, n)\tilde{f}_k(t)$, instead of $\tilde{f}_n(t)$ as the approximant of $f(t)$. As a basis for this refinement, Gaver developed the asymptotic expansion

$$\tilde{f}_k(t) \,-\, f(t) \sim \sum_{j=1}^{\infty} c_j(t)k^{-j} \;. \tag{8.4}$$

For example, from (8.4) it follows that the first error term is eliminated by considering

$2\tilde{f}_{2n}(t) - \tilde{f}_{n(t)}$ instead of $\tilde{f}_{2n}(t)$; i.e.,

$$2\tilde{f}_{2n}(t) - \tilde{f}_n(t) \sim \sum_{k=2}^{\infty} c_k(t) n^{-k} . \tag{8.5}$$

Gaver went further to develop an improved extrapolation formula, but later Stehfest [113] found what is in some sense the optimal linear combination. Here is Stehfest's result in [113].

Proposition 8.2. Let $\tilde{f}_n(t)$ be given as in (8.3) and let

$$f_n^*(t) = \sum_{k=1}^{n} w(k, n) \tilde{f}_k(t) \tag{8.6}$$

for

$$w(k, n) = (-1)^{n-k} \frac{k^n}{k!(n-k)!} . \tag{8.7}$$

Then

$$f_n^*(t) - f(t) = o(n^{-k}) \text{ as } n \to \infty \text{ for all } k .$$

Proof. By (8.4), it suffices to apply the combinatorial identity

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

see (12.7) and (12.17) on pp. 64-65 of Feller [42]. If (8.4) were an equality, then we would have

$$f_n^*(t) - f(t) = (-1)^{n+1} \frac{c_n(t)}{n!} + \sum_{j=1}^{\infty} \frac{c_{n+j}(t)}{n!} \sum_{k=1}^{n} (-1)^{n-k} \binom{n}{k} k^{-j} . \quad \blacksquare \tag{8.8}$$

Thus, the Gaver-Stehfest algorithm is (8.6) using the weights $w(k, n)$ in (8.7) and $\tilde{f}_n(t)$ in (8.3) and (8.2). However, to facilitate computation, we also use a recursive procedure from (27) and (29) of Gaver [49] to calculate $\tilde{f}_n(t)$. In particular, we let

$$\tilde{f}_m(t, j) = (1 + \frac{m}{j}) \tilde{f}_m(t, j-1) - \left[\frac{m}{j} \right] \tilde{f}_{m+1}(t, j-1) \tag{8.9}$$

with $\tilde{f}_m(t, 0) = m\alpha \hat{f}(m\alpha)$, $1 \le m \le 2n$. Then $\tilde{f}_n(t) = \tilde{f}_n(t, n)$.

Even with (8.9), the computation is not easy because of the factorials in (8.2), (8.3) and (8.7). Thus, having high precision is very important. We typically use $n = 16$ in (8.6). To do so, we need nearly thirty digit precision. For a PC, this is easily achieved using Kida's [74] UBASIC. The number of digits precision increases with n. As a function of n, the number decreases from about $2n$ to about $1.5n$ as n increases from 8 to 60.

The Gaver-Stehfest algorithm, here called GAVER, is displayed below for the M/G/1 waiting time transform (1.1). In this M/G/1 example the service-time distribution is gamma with mean 1 and shape parameter $1/2$, just as in the displayed version of EULER in §1 and Example 9.2 below. As before, we calculate the complentary cdf after removing the known atom at zero.

Insert Algorithm GAVER here (or slightly later)

The algorithm GAVER starts in lines 10 and 11 by specifying the number n of terms in (8.6) and defining three vectors whose length depends on n. In this case we have set $n = 16$. Running this program required 28 decimal places in UBASIC. The Laplace transform is defined in Lines 90-95. Lines 30-42 calculate $\tilde{f}_m(t)$ in (8.3) for $1 \le m \le n = 16$ using the recursion in (8.9), while lines 50–55 calculate the weighted sum in (8.6).

The Jagerman-Stehfest Method

In [66] Jagerman studied the Post-Widder approximants $f_n(t)$ in (8.1) and pointed out that the first few can often serve as useful rough approximations for the function $f(t)$ because they inherit structural properties of $f(t)$ such as monotonicity and convexity and they often can be calculated analytically. (Others have also suggested using these approximants, e.g., see ter Haar [117] and p. 149 of van der Pol and Bremmer [120], but Jagerman conducted an extensive study. These approximants in a particular M/M/1 example have also been studied in Abate and Whitt [4].)

In order to obtain greater numerical accuracy, Jagerman [67] proposed an algorithm for calculating the approximants $f_n(t)$ in (8.1); further discussion appears in Jagerman [68], [69] and Obi [95]. The essential idea behind the algorithm is to determine the generating function with coefficients $f_n(t)$, which is

$$G(z) \equiv \sum_{n=0}^{\infty} f_n(t) z^n = \frac{(n+1)}{t} \hat{f}\left[\frac{n+1}{t}(1-z)\right] , \tag{8.10}$$

and then apply the Fourier-series method for generating functions, exactly as described in §3 and §5 above.

The key to Jagerman's method is expressing the generating function in terms of the transform $\hat{f}$, as in (8.10), which is based on Theorem 12 of [66]. This is easily discovered by observing that

$$\hat{f}\left[\frac{n+1}{t} - z\right] = \sum_{n=0}^{\infty} \frac{(-1)^n \hat{f}^{(n)}((n+1)/t) z^n}{n!} . \tag{8.11}$$

From (8.11) it just remains to insert $(n+1)/t$ before z inside $\hat{f}$.

Given (8.10), Jagerman [67] uses the Cauchy contour integral, as in (3.9) and (3.10), to obtain

$$f_n(t) = \frac{n+1}{t} \frac{1}{2\pi r^n} \int_0^{2\pi} \hat{f}\left[\frac{n+1}{t}(1 - re^{iu})\right] e^{-inu}\, du . \tag{8.12}$$

Finally, he applies (without discussion) the Poisson summation formula, as in (5.35), to obtain

$$f_n(t) = \frac{n+1}{tmr^n} \sum_{k=1}^{m} \hat{f}\left[\frac{n+1}{t}\left[1 - re^{ikh}\right]\right] e^{-inkh} - e_d , \tag{8.13}$$

where $h = 2\pi/m$ and

$$e_d = \sum_{j=1}^{\infty} f_{n+jm}\left[t + \frac{tjm}{n+1}\right] r^{jm} ; \tag{8.14}$$

see (21) and (22) of [67]. We call this algorithm JAGERMAN. The last step makes it a Fourier-series method.

As with the treatment of generating functions in §5, a complete error analysis is available for computing $f_n(t)$ using (8.13) and (8.14), but it is important to note that the error between $f(t)$ and $f_n(t)$ is not controlled. Indeed, as Jagerman shows in [66] and (4)–(7) of [67], $f_n(t)$ approaches $f(t)$ quite slowly (of order n^{-1}).

In order to obtain improved numerical accuracy, we propose an enhancement to the algorithm JAGERMAN. In particular, since Jagerman shows in (4)–(7) of [67] that the errors here also satisfy (8.4), it is natural to consider applying the Stehfest [113] summation to this algorithm too, and it turns out to be effective. (This enhancement is more effective than Jagerman's σ-enhancement, which is equivalent to (8.5).)

First, for simplicity, we let $m = 2n$, so that, paralleling (5.38), (8.13) becomes

$$\begin{aligned} f_n(t) &= \frac{n+1}{2tnr^n} \sum_{k=1}^{2n} (-1)^k Re(\hat{f}) \left[\frac{n+1}{t} \left[1 - re^{\pi ik/n} \right] \right] - e_d \\ &= \frac{n+1}{2tnr^n} \Big\{ \hat{f}((n+1)(1-r)/t) + (-1)^n \hat{f}((n+1)(1+r)/t) \\ &\quad + 2 \sum_{k=1}^{n-1} (-1)^k Re(\hat{f}) \left[\frac{n+1}{t} (1 - re^{\pi ik/n}) \right] \Big\} - e_d \ . \end{aligned} \tag{8.15}$$

Assuming that $|f_n(t)| \leq 1$, we have $e_d \approx r^{2n}$, so that we let $r = 10^{-\gamma/2n}$ when we want to achieve accuracy to $10^{-\gamma}$. We then approximate $f(t)$ as in (8.6) by

$$f(t) \approx \sum_{k=1}^{m} w(k, m) f_{jk}(t) \ , \tag{8.16}$$

where $w(k, m)$ are the Stehfest coefficients in (8.7). We use $n = jk$ in $f_n(t)$ in (8.16) to achieve relatively large n and still benefit from (8.4). (Experience indicates that substantially higher n is needed for $f_n(t)$ in (8.1) to be close to $f(t)$ than for $\tilde{f}_n(t)$ in (8.3). This is intuitively reasonable because $\tilde{f}_n(t)$ is based on the beta distribution, which has finite support, whereas $f_n(t)$ is based on the gamma distribution.)

We call this modification of algorithm JAGERMAN algorithm POST-WIDDER. A UBASIC program implementing POST-WIDDER on the same M/G/1/ waiting-time example is displayed below. In this program we let $m = 6$ and $j = 10$ in (8.16) via lines 10 and 31. We have specified 10^{-8} accuracy via line 32.

Insert Algorithm POST-WIDER here (or slightly later)

Remark (8.1) in [6] we establish a direct connection between algorithms POST-WIDDER and EULER. In particular, we show that (8.15) converges to (5.27) term by term as $n \rightarrow \infty$ if we let $r = 1 - (A/2n)$. However, since the algorithms are based on fixed n, they serve as useful checks on each other.

9. Examples: The M/G/1 Waiting Time CDF

We now consider examples to illustrate the numerical inversion procedures described above. We start with the M/G/1 waiting-time cdf (1.2) with LST (1.1). We first eliminate the known atom of mass $(1-\rho)$ at the origin by writing $W(t) = (1-\rho) + \rho F(t)$. The cf ϕ of the cdf F is then $\phi(u) = (1 - \rho)\psi_e(u)/(1 - \rho\, \psi_e(u))$, where $\psi_e(u) = (\psi(u) - 1)/\tau iu$ with $\psi(u)$ the cf of the service-time cdf, say $S(t)$, with mean τ; i.e., ψ_e is the cf of the service-time stationary-excess distribution. With Laplace transforms, we consider the Laplace transform $\hat{F}^c(s) \equiv \int_0^\infty e^{-st} F^c(t)\, dt$ of $F^c(t) \equiv 1 - F(t)$, which is $[1 - \phi(is)]/s \equiv [1 - \hat{f}(s)]/s$ where $\hat{f}$ is the LST of $F(t)$ and Laplace transform of the density $f(t)$ when $F(t)$ is absolutely continuous. The difficulty in performing the numerical inversion depends on the service-time distribution and its cf ψ.

A Rational Transform: AN H_2 Service-Time Distribution

To begin with an easy case, suppose that the service-time cdf is H_2 (hyperexponential, a mixture of two exponentials), i.e., $S^c(t) = pe^{-\mu_1 t} + (1-p)e^{-\mu_2 t}$, $t \geq 0$, with parameters $p = 2/3$, $\mu_1 = 2$ and $\mu_2 = 1/2$, so that the first three moments of S are $s_1 = 1$, $s_2 = 3$, $s_3 = 33/2$, and the cf is

$$\psi(u) = \frac{2}{3}(1 - 0.5\, iu)^{-1} + \frac{1}{3}(1 - 2iu)^{-1} = \frac{2 - 3iu}{(2-iu)(1-2iu)} \ . \qquad (9.1)$$

The associated cf's for the stationary-excess cdf and F are

$$\psi_e(u) = \frac{2\,(1 - iu)}{(2 - iu)(1 - 2iu)} \quad \text{and} \quad \phi(u) = \frac{4\theta\,(iu - 1)}{2u^2 + (3 + 4\theta)iu - 4\theta} \ , \qquad (9.2)$$

where $\theta = (1-\rho)/2$.

In this case we can invert the associated Laplace transform of the complementary cdf analytically, using partial fractions. Indeed, *whenever the Laplace transform of F^c is rational* (*a ratio of polynomials*), *we would usually prefer a partial-fraction algorithm, because it provides an analytical inversion as well as numbers.* Here we obtain

$$F^c(t) = qe^{-\eta_1 x} + (1-q)e^{-\eta_2 x}, \quad x \geq 0, \qquad (9.3)$$

where

$$\begin{gathered} q = \frac{1}{2}\left[1 - \frac{1 - 4\theta/3}{\sqrt{1 - 8\theta\,\rho/9}}\right] \\ \eta_2^{-1} = \frac{1}{2} + \frac{3}{8\theta}\left[1 + \sqrt{1 - 8\theta\,\rho/9}\right] \quad \text{and} \quad \eta_1^{-1} = 1 + \frac{3}{4\theta} - \eta_2^{-1} \ . \end{gathered} \qquad (9.4)$$

Remark (9.1) It is well known that F is H_2 when S is H_2; more generally, F is a mixture of exponentials whenever S is a mixture of exponentials; see Smith [111] and Keilson [71]. ■

Now we numerically invert the cf ϕ in (9.2). For a specific computation, let $\rho = 0.75$, so that

$q = 0.0648$, $\eta_1^{-1} = 0.6277$, $\eta_2^{-1} = 6.372$, $m_1 = 6$ and $m_2 = 76$. Suppose that we want to calculate $F^c(t)$ for $0 \leq t \leq 30$. We first apply EULER, the alternating series variant of the Fourier-series method, which is summarized at the beginning of §7, and then GAVER and POST-WIDDER, the two alternative methods in §8. For EULER with Euler summation parameter $m = 11$, we found that $A = 19.1$ and $n = 15$ (as displayed in §1) produced accuracy to 10^{-7}, while $A = 20.7$ and $n = 20$ produced accuracy to 10^{-8}. (This requires double precision; see Remark 5.8.) Similar results were also obtained for the variant of EULER in (5.30), which uses the imaginary part. The numbers obtained from GAVER with $n = 16$ terms agreed with EULER to 10^{-8}, thus providing an independent check. In order for GAVER to work with $n = 16$, we needed about 28-digit precision. The accuracy increases with n, but so does the required precision. The algorithm POST-WIDDER with $r = 10^{-7/2n}$, $m = 6$ and $j = 10$ also produced accuracy to 10^{-8}. Similar results were obtained for $m = 4$ and $j = 20$, but $m = 4$ and $j = 10$ was not enough.

Next we apply the variant of the Fourier-series method POISSON. Since F concentrates on the positive real line, we apply inversion formula (3.6) and the trapezoidal rule (4.4). It thus remains to determine an appropriate step size h and truncation point T, for which we apply Sections 5 and 6.

Using algorithm POISSON, we *estimate* the step size h required to yield a discretization error of 10^{-k} by doing an asymptotic analysis of $F^c(y)$ as $y \to \infty$. From the partial fraction expansion used to obtain (9.3) from (9.2), or from p. 238 of Doetsch [36], we obtain

$$F^c(y) \sim (1-q)\, e^{-\eta_2 y} = 0.935 e^{-0.157y} \quad \text{as} \quad y \to \infty\,, \tag{9.5}$$

so that, from (5.13),

$$h^{-1} \approx \frac{1}{2\pi} \left[30 + \frac{\log\,(10^{-k}/0.935)}{-0.157} \right] = 16.4 \quad \text{for} \quad k = 5\,. \tag{9.6}$$

Alternatively, we can determine the required step size h to achieve a discretization error *bound* of 10^{-k}. One approach is to bound the tail probabilities using the moments. From the cf in (9.2), as well as from (9.3), we can calculate the moments of F (see Kleinrock [75] for general formulas), the first two being

$$m_1 \equiv \int_0^\infty x dF(t) = \frac{3}{2(1-\rho)} \quad \text{and} \quad m_2 \equiv \int_0^\infty x^2 dF(t) = \frac{11}{2(1-\rho)} + \frac{9\rho}{2(1-\rho)^2} \ . \qquad (9.7)$$

We can apply Chebychev's inequality to bound the tail probabilities by

$$F^c(t) \le \frac{m_1}{t} \quad \text{and} \quad F^c(m_1 + t) \le \frac{m_2 - m_1^2}{t^2} , \qquad (9.8)$$

but these bounds are not very useful. Applying (5.13), with a discretization error bound of 10^{-k}, we obtain $F^c\ ((2\pi/h) - t) = 10^{-k}$ for $t \le 30$, so that

$$\frac{1}{h} \le \left[\frac{1}{2\pi} \right] \left[30 + \frac{m_1}{10^{-k}} \right] = 95{,}498 \quad \text{for} \quad k = 5 \qquad (9.9)$$

and

$$\frac{1}{h} \le \left[\frac{1}{2\pi} \right] \left[m_1 + 30 + \left[\frac{m_2 - m_1^2}{10^{-k}} \right]^{1/2} \right] = \frac{2036}{2\pi} = 324 \quad \text{for} \quad k = 5 \ , \qquad (9.10)$$

respectively.

In this case, since F is H_2, we actually have the *much better exponential bound*

$$F^c(t) \le e^{-0.157t} , \quad t \ge 0 \ . \qquad (9.11)$$

The bound in (9.11) holds because an H_2 cdf has decreasing failure rate, which implies that it is stochastically dominated in the failure rate ordering (which is stronger than the usual stochastic ordering) by an exponential distribution with the limiting rate; see Chapter 8 of Ross [107]. Alternatively, we can obtain the same bound by applying Prop. 5.3.2 of Stoyan [115]. From

(9.11), we obtain

$$\frac{1}{h} \leq \frac{1}{2\pi} \left[30 + \frac{\log (10^{-k})}{-0.157} \right] = 16.5 \quad \text{for} \quad k = 5 \tag{9.12}$$

which is essentially equivalent to the estimate (9.6).

An *estimate* for *h* could be obtained by assuming that *F* has an exponential tail (which turns out to be correct by (9.5)), and using the known mean to approximate the rate. If we do this, then we obtain

$$\frac{1}{h} \approx \frac{1}{2\pi} \left[30 + \frac{\log (10^{-k})}{-0.167} \right] = 15.8 \quad \text{for} \quad k = 5 \, , \tag{9.13}$$

which is a good estimate in this case.

Turning to the truncation, we easily obtain $Re(\phi)$ from (9.2), i.e.,

$$Re\,(\phi)(u) = \frac{4\theta\,[(1 + 4\theta)u^2 + 4\theta]}{(2u^2 - 4\theta)^2 + (3 + 4\theta)^2\, u^2} \; . \tag{9.14}$$

Note that $Re(\phi)(u)$ in (9.14) is a decreasing positive function of *u*, so that we can construct the bounding function $\beta(u)$ in (6.3) for $\gamma(u)$ in (4.4) by

$$|\gamma(u)| \leq \frac{2\,Re(\phi)(u)}{\pi u} = \frac{8\theta\,[(1 + 4\theta)u^2 + 4\theta]}{\pi u\,[(2u^2 - 4\theta)^2 + (3 + 4\theta)^2\, u^2]} = \frac{6u^2 + 2}{\pi\,(16u^5 + 41u^3 + u)}$$

$$\leq \frac{3}{\pi\, 8u^3} \equiv \beta(u) \quad \text{for} \quad u \geq 1 \, . \tag{9.15}$$

Remark (9.2) For an exponential cdf with mean *m*, $Re(\phi)(u) = 1/(1 + [mu]^2)$. Since the cf of a mixture is the mixture of the cf's (pp. 504, 509 of Feller [43]), $Re(\phi)(u)$ is monotone whenever ϕ is the cf of a mixture of exponentials. Moreover, whenever *F* is a finite mixture of exponentials, as in this example, $Re(\phi)(u) = O(|u|^{-2})$. When we apply EULER using (4.6), it is significant that the real part of the transform of $F^c(t)$ is appropriately monotone. In particular, since

$$Re(\hat{F}^c)(u + iv) = m(1 + mu)/((1 + mu)^2 + m^2 v^2)$$

when F is an exponential cdf with mean m, $Re(\hat{F}^c)(u + iv)$ is positive and decreasing in v when $u \geq 0$ and F is a mixture of exponentials. Thus, (4.6) is indeed an alternating series. ■

We can *estimate* the truncation point $T = Nh$ yielding a truncation error of 10^{-k} by applying (6.4). If we let t be the mean of F (i.e., $t = 6$), then $1/|t| = 1/6 > h$ for $k > 2$ (see Table 1), so that (6.4) becomes

$$\max \{h\beta(T), \beta(T)/|t|\} = \beta(T)/|t| = \frac{1}{16\pi T^3} = 10^{-k} \tag{9.16}$$

and $T \approx (10^k/16\pi)^{1/3} = 12.6$ for $k = 5$. Finally, we can combine (9.6) and (9.16) to obtain an *estimate* for the total number of terms of $N = Th^{-1} = (12.6)(16.4) = 207$ for $k = 5$.

If instead we want to achieve a truncation error *bound*, then we can set

$$\int_T^\infty \beta(u)\, du = \frac{3}{16\pi T^2} = 10^{-k} \quad \text{and} \quad T = \left[\frac{3}{16\pi\, 10^{-k}} \right]^{1/2} = 77.3 \quad \text{for} \quad k = 5\ . \tag{9.17}$$

Hence, a *bound* of 2×10^{-k} on the total error is achieved by combining (9.12) and (9.17) to obtain $h^{-1} = 16.5$, $T = 77.3$ and $N = T/h = 1276$ for $k = 5$. Even with the relatively crude Chebychev bound in (9.10), we obtain the feasible computation based on $N = 25{,}050$ for $k = 5$. Table 1 displays the discretization parameter h^{-1}, truncation parameter $T = Nh$ and total number of terms N required to achieve prescribed errors of 10^{-k} as a function of k for this example, using estimates and bounds.

Insert Table 1 here (or slightly later)

Table 2 contains a comparison of numerical inversion approximations based on POISSON with exact values of $F^c(t)$ based on (9.3). Also included is the asymptotic expression in (9.5), which is known to be an excellent approximation for this example when t is not too small; see pp. 54-62 of Tijms [118]. (This illustrates the power of asymptotic analysis for computing small tail probabilities.) Indeed, the asymptotic approximation (9.5) is exact to 10^{-8} for $t \geq 12$, but does not perform well for $t < 1$.

For the inversion, three levels of accuracy are considered: 2×10^{-k} for $k = 3, 5$ and 7. The inversion parameters are determined by the standard estimates described above, with $t = 1$ and $t = 6$ both being used in (9.16) to determine the truncation point T and thus N. Changing from $t = 6$ (as in Table 1) to $t = 1$ increases the required number of terms N by a factor of $6^{1/3} = 1.82$. The accuracy of the approximation is described in Table 2 by giving the error as a factor times 10^{-k} when the desired error is 2×10^{-k}.

Insert Table 2 here (or slightly later).

From Table 2 it is apparent that the estimates are quite accurate. Indeed, the error at the intended t is 1×10^{-k} when the desired error is 2×10^{-k} in all six cases. Consistent with (6.4) and (9.16), the error increases as t gets small. This lack of accuracy is anticipated because $F^c(t)$, when reflected about $t = 0$, has a discontinuous derivative at 0.

Finally, we consider convolution smoothing to obtain a truncation bound independent of the cdf F. As in §6, we use a normal smoothing cdf with mean 0 and variance σ^2. From Remark 6.3, we know that the error resulting from convolution smoothing with a normal cdf having variance σ^2 is of order $f(t)\sigma$, where $f(t)$ is the density of F at t. From (9.3), here the density $f(t)$ is 0.146 at $t = 1$ and 0.057 at $t = 6$. Thus, for the desired error to be of order 2×10^{-k}, σ should be of order 6.9×10^{-k} for $t = 1$ and 1.75×10^{-k} for $t = 6$. Thus, the appropriate value of T as a function of k when $t = 1$ ($t = 6$) is 411 (162) for $k = 3$, 6960 (2714) for $k = 5$ and 82,600 (32,200) for $k = 7$. The corresponding values of N, assuming the estimated values of h in (100), are 4809 (1895) for $k = 3$, 114,144 (44,510) for $k = 5$ and 1,743,000 (680,000) for $k = 7$. *These values are obviously substantially greater than actually required without convolution smoothing* (see the last column of Table 1), so that convolution smoothing provides no assistance in the required computation in this example.

A similar analysis applies with the variant POISSON-PAB using (6.16). By (6.18), $T = 2 \log (2/\varepsilon_2)/\varepsilon_1$, but the analog of Remark 6.3 implies that we must have $f(t)\varepsilon_1 \approx \varepsilon_2$ if the perturbation error ε_1 does not seriously inflate ε_2. Hence, we need $T \approx 2 f(t) \log (2/\varepsilon_2)/\varepsilon_2$, which leads to even larger values, e.g., for $k = 5$ and $x = 1$, $T \approx 356{,}000$ and $N \approx 5{,}840{,}000$. From these calculations, it is apparent that we cannot be too demanding about the errors ε_1 and ε_2 in (6.16), because the resulting computation will soon get out of hand. Of course, we have little difficulty if we can choose ε_1 relatively large with ε_2 small, but the interpretation of such a choice is questionable. In an application, we may well know that ε_1 not need be too small, but in that case it does not seem to make sense to require that ε_2 be very small if $f(t)\varepsilon_1$ is much larger than ε_2.

An Irrational Transform: A $\Gamma_{1/2}$ Service-Time Distribution

We now consider the M/G/1 queue with a slightly more difficult service-time distribution, in particular, a gamma distribution with shape parameter 1/2, which has density

$$s(t) = \frac{e^{-t/2}}{\sqrt{2\pi t^3}}, \quad t \geq 0, \tag{9.18}$$

first three moments $s_1 = 1$, $s_2 = 3$ and $s_3 = 15$ and the non-rational cf $\psi(u) = (1 - 2iu)^{-1/2}$. The programs for applying the algorithms EULER, GAVER and POST-WIDDER to this example are displayed in Sections 1 and 8.

The cf ϕ of the cdf F (again $W(t) = 1 - \rho + \rho F(t)$) is thus

$$\phi(u) \; = \; \frac{(1-\rho)([1 \; - \; 2iu]^{1/2} \; - \; 1)}{-(\rho \; + \; iu)[1 \; - \; 2iu]^{1/2} \; + \; \rho} \; , \tag{9.19}$$

where

$$\begin{aligned} &(1 \; - \; 2iu)^{1/2} \; = \; X(C \; + \; iS) \, , \quad X \; = \; (1 \; + \; 4u^2)^{1/4} \; , \\ &C \; = \; \cos \, (\theta/2) \, , \quad S \; = \; \sin \, (\theta/2) \quad \text{and} \quad \theta \; = \; \arctan \, (-2u) \; . \end{aligned} \tag{9.20}$$

It turns out that the cf ϕ in (9.19) can be inverted analytically (which we derive elsewhere) to obtain the following exact expression for the cdf

$$\begin{aligned} F^c(t) \; &= \; Ae^{-\delta t} \; + \; \rho^{-1} \, \Phi^c(\sqrt{t}\,) \; - \; Ae^{-\delta t} \, \Phi^c \, (([1 \; + \; 8\rho]^{1/2} \; - \; 1)\sqrt{t}\,/2) \\ &- \; Be^{\eta t} \, \Phi^c \, (([1 \; + \; 8\rho]^{1/2} \; + \; 1)\sqrt{t}\,/2) \; , \end{aligned} \tag{9.21}$$

where $\Phi^c(t)$ is the complementary standard normal cdf with mean 0 and variance 1,

$$\begin{aligned} &A \; = \; 1 \; - \; \frac{2(1-\rho)}{\sqrt{1 \; + \; 8\rho} \; (1 \; + \; 2\rho \; + \; \sqrt{1 \; + \; 8\rho}\,)} \; , \quad B \; = \; A \; + \; \frac{1 \; - \; 2\rho}{\rho} \\ &\delta \; = \; \frac{\sqrt{1 \; + \; 8\rho} \; - \; (4\rho \; - \; 1)}{4} \quad \text{and} \quad \eta \; = \; 2\rho \; + \; \delta \; - \; \frac{1}{2} \; . \end{aligned} \tag{9.22}$$

Now we calculate $F^c(t)$ by numerical inversion using $\phi(u)$ in (9.19). EULER, GAVER and POST-WIDDER all work very well, just as for Example 9.1. EULER produces accuracy of 10^{-7} with $n \; = \; 15$ and $A \; = \; 19.1$, and accuracy of 10^{-8} with $n \; = \; 20$ and $A \; = \; 20.7$. (As before, this requires double precision.) GAVER produces an independent check to 10^{-8} with $n \; = \; 16$ (and even higher precision). POST-WIDDER produces an independent check to 10^{-8} with $m \; = \; 6$ and $j \; = \; 10$. For POST-WIDDER, double precision suffices.

Turning to POISSON, we consider estimates and bounds of the tail probabilities. Not using (9.21), we can *estimate* the tail probability by doing asymptotic analysis using the Laplace transform of the complementary cdf $F^c(t)$, which can be expressed as $\hat{F}^c(s) \; = \; N(s)/D(s)$, where

$$N(s) = s^{-1}[1 + (s-1)(1 + 2s)^{1/2}]$$
$$D(s) = \rho + (s-\rho)(1 + 2s)^{1/2} . \tag{9.23}$$

In this case, $\hat{F}^c(s)$ is not rational, but the right-most singularity is a simple pole at $s = -\delta$ for δ in (9.22), so that

$$F^c(t) \sim Ae^{-\delta t} \quad \text{as} \quad t \to \infty \tag{9.24}$$

where A is in (9.22) and

$$A = \lim_{s \to -\delta} (s+\delta)\hat{F}^c(s) = \frac{N(-\delta)}{D'(-\delta)} \tag{9.25}$$

and $D(-\delta) = 0$; see p. 238 of Doetsch [36]. For example, when $\rho = 0.75$,

$$F^c(t) \sim 0.963e^{-0.161t} \quad \text{as} \quad t \to \infty . \tag{9.26}$$

Assuming that we want to calculate $F(t)$ for $t < 30$, our estimate for the discretization parameter based on (5.13), (9.26) and a discretization error of 10^{-k} is

$$\frac{1}{h} \approx \frac{1}{2\pi} \left[30 + \frac{\log(10^{-k}/0.963)}{-0.161} \right] , \tag{9.27}$$

which equals 11.5, 16.1 and 20.6 for $k = 3, 5$ and 7.

It turns out that a good *bound* can also be found in this case, because as noted above this service-time distribution is also a mixture of exponentials. Hence, F is a mixture of exponentials by Keilson [71] and $F^c(t) \leq e^{-0.161t}$.

Turning to the truncation, we first apply (9.19) to determine that

$$Re(\phi)(u) = \frac{2\,(1 - \rho)\,A(u)}{A(u)^2 + B(u)^2} , \tag{9.28}$$

where

$$A(u) = 1 - 2\rho + XC \quad \text{and} \quad B(u) = 2u + XS \tag{9.29}$$

with X, C and S in (9.20). (By Remark 9.1, $Re(\phi)(u)$ is monotone.) Since $\arctan(-2u) \rightarrow -\pi/2$ as $u \rightarrow \infty$, $S \rightarrow \sin(-\pi/4) = -1/\sqrt{2}$ and $C \rightarrow \cos(-\pi/4) = 1/\sqrt{2}$ as $u \rightarrow \infty$, so that $Re(\phi)(u) \sim (1 - \rho)/2u^{3/2}$, which equals $1/8u^{3/2}$ when $\rho = 0.75$. For a point $t \leq h^{-1}$ and a truncation error of 10^{-k}, we apply (6.4) to obtain an estimated truncation point of

$$T = Nh \approx \left[\frac{10^k}{4\pi} \right]^{2/5} , \tag{9.30}$$

which is 2.8, 17.7 and 112 for $k = 1$, 2 and 3, respectively.

Table 3 contains a comparison of numerical inversion approximations based on POISSON with exact values of $F^c(t)$ based on (9.21).

Insert Table 3 here (or slightly later)

(The ‘‘exact’’ solution in (9.21) can be computed using the rational approximation for the complementary cdf $\Phi^c(t)$ in 26.2.17 of Abramowitz and Stegun [8].) The results are similar to those for the M/H$_2$/1 case in Example 9.1.

Other Service-Time Distributions

The two service-time distributions considered above were both mixtures of exponential distributions, implying that $Re(\hat{F}^c)(a + iu)$ is a decreasing positive function of u, thus favoring EULER; see Remarks 9.1 and 9.2. Hence, we also consider two service-time distributions for which $Re(\hat{F}^c)(a + iu)$ is *not* a positive decreasing function of u. (We keeping the traffic intensity at $\rho = 0.75$.)

In particular, as service-time distributions we consider the uniform distribution and the E_2 (Erlang of order 2) distribution. Since the cf of the uniform distribution on $[0, a]$ is

$$\psi(u) \; = \; \frac{e^{iua} \; - \; 1}{iau} \; = \; \frac{\cos ua \; + \; i \; \sin ua \; - \; 1}{iau} \; , \tag{9.31}$$

we see that the real and imaginary parts of $\psi_e(a + iu)$ and thus also $Re(\hat{F}^c)(a + iu)$ involve trigonometric functions, so that $Re(\hat{F}^c)(a + iu)$ is not monotone in u.

Similarly, by (12.1)–(12.6) below, for an E_k service-time distribution the real and imaginary parts of the cf's $\psi(a + iu)$ and $\psi_e(a + iu)$ involve trigonometric functions, so that $Re(\hat{F}^c)(a + iu)$ is again not monotone. However, since $arctan(u) \rightarrow \pi/2$ as $u \rightarrow \infty$ (see (12.6)), $Re(\hat{F}^c)(a + iu)$ turns out to be eventually monotone as u gets large. Moreover, (12.4) shows that the transform of the E_k distribution decays rapidly. Thus, the E_k service-time distribution is substantially less difficult than the uniform service-time distribution.

It turns out that EULER performs well on both these examples, obtaining 10^{-7} accuracy with $A = 19.1$ and n about 20, but GAVER performs well only for the E_2. For the uniform service-time distribution, GAVER was only able to obtain accuracy to about 10^{-4} for $N = 30$. These examples are also more difficult for POST-WIDDER. For $m = 6$, $j = 20$ and $E \equiv \gamma = 8$, the error is about $10^{-7} - 10^{-8}$ for an E_2 service time and $10^{-5} - 10^{-6}$ for a uniform service time.

10. The Time-Dependent Mean of RBM

This example has three main purposes: first, to illustrate how to obtain the asymptotic behavior of $F^c(t)$ when the right-most singularity of the LST is not a simple pole; second, to consider a slowly decaying cf for which the standard truncation bound (6.3) for POISSON can be inadequate; and, third, to illustrate the use of convolution smoothing and other windowing.

For these purposes, we consider the time-dependent mean of regulated or reflecting Brownian motion (RBM), drawing on Abate and Whitt [2], to which we refer in this section by AW. Let $R(t)$ be canonical RBM, i.e., Brownian motion with negative unit drift and unit diffusion coefficient, modified by an impenetrable reflecting barrier at the origin. In this section, let

$$F(t) \; = \; \frac{E(R(t) \mid R(0) \; = \; 0)}{ER(\infty)} \;\; , \quad x \geq 0 \; , \tag{10.1}$$

i.e., the time-dependent mean starting at 0, normalized by the steady-state limit, which turns out to be a bonafide cdf on the positive real line with cf

$$\phi(u) \; = \; \int_0^\infty e^{iut} \; dF(t) \; = \; \frac{2}{1 \; + \; (1 \; - \; 2iu)^{1/2}} \; = \; \frac{1 \; - \; (1 \; - \; 2iu)^{1/2}}{iu} \;\; ; \tag{10.2}$$

see (4.5) of AW. Explicit expressions for the cdf F, its density f, all its moments, and the asymptotic behavior as $t \to 0$ and as $t \to \infty$ are given there in Theorems 1.1(a), 1.3 and 4.1, Corollaries 1.3.1, 1.3.4 and 1.3.5, and the concluding remark in § 4.3. (The first three moments are $1/2$, 1 and $15/4$.) Numerical values of the complementary cdf F^c and the density f are given in Table 1 there.

Suppose now that we wish to calculate $F(t)$ by numerically inverting (10.2). First, the algorithms EULER (with $n \; = \; 15$ and $A \; = \; 19.1$), GAVER (with $n \; = \; 16$) and POST-WIDDER (with $m \; = \; 6$ and $j \; = \; 10$) give 10^{-7} accuracy, just as in §9. As in §9, this cdf is a mixture of exponentials (Theorems 1.3 and 1.7 of AW), so that (4.6) is again a true alternating series.

Next we turn to POISSON. Since F concentrates on the positive real line, we apply the inversion integral (3.6) using the trapezoidal rule (4.1), which yields (4.4). We can estimate and bound the discretization error by applying (5.13). As noted in Remark 4.4 of AW, the right-most singularity of the Laplace transform of $F^c(t)$ at $s \; = \; -1/2$ is not a simple pole, but a branch point, so that we apply Heaviside's theorem, p. 254 of Doetsch [36], to deduce that

$$F^c(t) \sim \frac{2e^{-t/2}}{\sqrt{\pi \, t^3}} \quad \text{as} \quad t \to \infty \; . \tag{10.3}$$

(This is obtained in Corollary 1.1.2 of AW by a different argument.) Numerical values of (10.3) are given in Tables 3 and 4 of AW. Alternatively, we could apply the exponential approximation for F^c in (1.3) of AW, which is

$$F^c(t) \approx 0.276\, e^{-0.764t}\ , \quad t \geq 0\ . \tag{10.4}$$

As in §9, the cdf F is a mixture of exponentials, so that it has decreasing failure rate (Theorems 1.3 and 1.7 of AW) and we obtain the *exponential bound* $F^c(t) \leq e^{-t/2}\ , \quad t \geq 0.$

Since $F^c(10) \approx 0.00024$ by (10.4), we decide to calculate $F^c(t)$ for $0 < t < 10$. Since $F^c(8) \approx 9.1 \times 10^{-4} \approx 10^{-3}$ and $F^c(16) \approx 8.4 \times 10^{-6} \approx 10^{-5}$ by (10.4), we apply (5.13) to obtain *estimates* for the discretization parameter h^{-1} of $26/2\pi \approx 4.14$ and $18/2\pi \approx 2.87$ when the discretization error is 10^{-5} and 10^{-3}.

Turning to the truncation error, we find from (9.20) and (10.2) that

$$Re(\phi)(u) = \frac{-Im\,(\sqrt{1 - 2iu}\,)}{u} = \frac{-(1 + 4u^2)^{1/4}\sin(\theta/2)}{u} \sim \frac{1}{\sqrt{u}} \quad \text{as} \quad u \to \infty \tag{10.5}$$

where $\theta = \arctan(-2u)$. Hence, from (6.2), (6.4) and (10.5), our *estimated* truncation point is

$$T = Nh \approx \left[\frac{2(10^k)}{\pi\, t} \right]^{2/3} \tag{10.6}$$

which is 1594 (483) for $t = 1$ $(t = 6)$ and $k = 5$; the associated values of N using $h^{-1} = 4.14$ are 6600 (2000) for $t = 1$ $(t = 6)$, which obviously are feasible. Table 4 displays the computational results for $h^{-1} = 4.41$ and $N = 500$ and 2000. As predicted by the estimate, the error for $N = 2000$ at $t = 6$ is 10^{-5}. By (10.6), the estimated error for $N = 500$ at $t = 1$ $(t = 6)$ is 48×10^{-5} (8×10^{-5}), which is also consistent with Table 4.

Insert Table 4 here (or slightly later).

Unlike § 9, we have some difficulty when we apply the truncation *bound* in (6.3). A bounding function for $u \geq 1$ is $\beta(u) = 3/\pi u^{3/2}$, which for an error bound of 10^{-k} leads to the truncation point of $T = Nh \leq [6(10^k)/\pi]^2$, which tends to be an *infeasible computation* for large k, e.g., for $k = 5$.

For this example, it is thus reasonable to consider convolution smoothing. Given that the truncation error is 10^{-5}, $h^{-1} = 4.14$ as above and $N = 2000$ (500), we find from (6.12) that the normal standard deviation must be $\sigma = 0.01$ ($\sigma = 0.04$). Moreover, from (6.17) we see that the associated allowable perturbation in (6.16) must be $\varepsilon_1 = \sigma \sqrt{2 \log 2/\varepsilon_2} = 5.47\sigma = 0.055$ ($\varepsilon_1 = 0.22$), which tends to be inconsistent with an overall accuracy of 10^{-5}. (See Remark 6.3 and Table 1 of AW for numerical values of the density.) To summarize, we can obtain the error *bounds* in (6.16) with $\varepsilon_2 = 10^{-5}$ and $\varepsilon_1 = 100/N$ by using $h^{-1} = 4.14$, $\sigma = 182/N$ and N. Even though this is not consistent with an overall error of 10^{-5}, this should usually be adequate *guaranteed accuracy* for most applications. (Given $\varepsilon_2 = 10^{-5}$, we have $\varepsilon_1 = 0.05$ and $\sigma = 0.01$ for $N = 2000$, or $\varepsilon_1 = 0.005$ and $\sigma = 0.001$ for $N = 20{,}000$.)

To show how the convolution smoothing actually affects the computation, the results with convolution smoothing are also displayed in Table 4 for the same values of h^{-1} and N. As indicated above, the normal standard deviation is 0.01 and 0.04 for $N = 2000$ and 500. From (6.9), (6.12) and (6.19), the weight function here is $w(kh) = \varepsilon_2^{(k/N)^2}$ for $\varepsilon_2 = 10^{-5}$ and $N = 2000$. From Table 4, we see that *the convolution smoothing actually tends to improve the accuracy, except at very low values of t.*

Finally, we also consider convolution smoothing with the smoothing cdf that attaches probability $1/2$ to 0 and probability $1/2$ to π/Nh, which yields (6.9) with the new weight function

$$w(kh) = [1 + \cos(k\pi/N)]/2 . \tag{10.7}$$

This weight function is a popular window in signal processing, corresponding to a special case of both the Hanning and Hamming windows; see (27a) and (30a) of Harris [53]. Table 4 shows that this weight function performs even better than the Gaussian window (convolution smoothing with the normal cdf). Notice that for this smoothing cdf, the standard deviation is $\sigma = \pi/2Nh$, which is 0.0033 (0.013) for $N = 3000$ ($N = 500$), i.e., smaller than for the normal smoothing cdf.

11. The Number Served in an M/M/1 Busy Period

This example illustrates how lattice probability distributions can be treated by the Fourier-series method using the algorithm LATTICE-POISSON in §5. Let $F(n)$ be the cdf of the number of customers served in the busy period of an M/M/1 queue with traffic intensity ρ. From p. 65 of Riordan [103], the cf is

$$\phi(u) = \frac{1 - \sqrt{1 - 4\beta e^{iu}}}{\sqrt{4\beta\rho}} \quad \text{for} \quad \beta = \rho/(1+\rho)^2 . \tag{11.1}$$

The known probability mass function is

$$p_n \equiv F(n) - F(n-1) = \frac{1}{n}\begin{bmatrix} 2n-2 \\ n-1 \end{bmatrix} \rho^{n-1}(1+\rho)^{-2n+1} , \quad n \geq 1 . \tag{11.2}$$

The mean is $(1-\rho)^{-1}$ and the variance is $(\rho + \rho^2)(1-\rho)^{-3}$. As before, we will consider $\rho = 0.75$, which yields a mean of 4 and a variance of 84.

The algorithm LATTICE-POISSON easily produces the required accuracy (e.g., to 10^{-8}) with no special analysis; see Table 1 of [7]. In contrast, when we applied EULER, GAVER and POST-WIDDER directly to the transform in (11.1), we got no useful results, which is not surprising, because the cdf has jumps; see §14. We also applied EULER, GAVER and POST-WIDDER to the smoothed or interpolated continuous cdf with cf.

$$\tilde{\phi}(u) = \frac{1 - e^{-iu}}{iu}\,\phi(u) , \tag{11.3}$$

corresponding to the convolution with a uniform distribution on the interval $[-1, 0]$, which leaves the values at integer points unchanged. (This was motivated by Abate and Dubner [1].) Unfortunately, we then only obtained accuracy to about 10^{-4}, even after increasing the number of terms. Moreover, the estimated truncation errors with EULER seriously underestimated the true errors. Hence, the sharp corners of the continuous smoothed cdf still cause serious problems for EULER, GAVER and POST-WIDDER. We do much better applying EULER, GAVER or POISSON after smoothing with a nicer smoothing cdf, e.g., the normal with standard deviation 0.05, so that more than 10 standard deviations are required to have the point masses at adjacent points cross over. Indeed, the algorithm POISSON-PAB in Remark 6.4 with $\varepsilon_1 = 0.49$ in (6.16) works quite well for lattice distributions. However, we have not seen anything outperform LATTICE-POISSON. For further discussion about this example, see [7].

12. Convolutions of Exponential Distributions

As indicated by Harrison [54], convolutions of exponential distributions arise in many contexts, such as first-passage-time distributions in continuous-time Markov chains, Cox distributions and sojourn-time distributions in queueing networks. However, contrary to the claim on p. 74 of [54], these distributions are not difficult to evaluate by numerically inverting Laplace transforms. First, the transform of the convolution is easily expressed as the product of the transforms. Second, convolution is a smoothing operation, so that the final distribution tends to be very smooth, which translates into $Re(\phi)(u)$ tending to have a very rapidly decaying tail (by virtue of the Riemann-Lebesgue lemma discussed after (6.3)).

To support our second claim above, we calculate the real part of the transform of a convolution of exponentials, bound its tail and describe the asymptotic behavior in the Erlang case.

Proposition 12.1. If ϕ is the cf of a convolution of exponentials, i.e., if

$$\phi(u) = \prod_{j=1}^{n} (1 - ium_j)^{-k_j} \tag{12.1}$$

for positive integers and positive reals m_j, then

$$Re(\phi)(u) \;=\; \frac{\cos\left[\sum_{j=1}^{n} k_j \xi_j\right]}{\prod_{j=1}^{n} \left(1 + (m_j u)^2\right)^{k_j/2}} \tag{12.2}$$

for $\xi_j \;=\; arctan(m_j u)$, so that

$$|Re(\phi)(u)| \le \left[\prod_{j=1}^{n} m_j u\right]^{-1} . \tag{12.3}$$

Moreover, for $n \;=\; 1$, as $u \;\to\; \infty$

$$|Re(\phi)(u)| \sim \begin{cases} \dfrac{k_1}{(m_1 u)^{k_1+1}} & \text{for } k_1 \text{ odd} \\[2ex] \dfrac{1}{(m_1 u)^{k_1}} & \text{for } k_1 \text{ even .} \end{cases} \tag{12.4}$$

Proof. Using the polar form, write

$$(1 \;-\; ium_k) \;=\; r_k e^{i\xi_k} \;=\; r_k(\cos\xi_k \;+\; i\sin\xi_k) , \tag{12.5}$$

where

$$r_k \;=\; \sqrt{1 \;+\; (m_k u)^2} \quad \text{and} \quad \xi_k \;=\; arctan(-m_k u) , \tag{12.6}$$

to obtain

$$Re[(1 \;-\; ium_k)^{-1}] \;=\; r_k^{-1}\cos\xi_k \tag{12.7}$$

for r_k and ξ_k in (12.6), from which (12.2) follows. It is easy to see that $(1 + x^2)^{1/2} \ge x$, from which (12.3) follows. Finally, since $arctan(u) \;\to\; \pi/2$ as $u \;\to\; \infty$, $|\cos k_1 \xi_1| \;\to\; 1$ for k_1 even and $|\cos k_1 \xi_1| \to 0$ for k_1 odd as $u \to \infty$. However, further analysis shows that $|\cos k_1 \xi_1| \sim k_1/m_1 u \sim k_1/r_1$ as $u \;\to\; \infty$, which implies (12.4). ■

Note that the inequality (12.3) makes it relatively easy to bound the truncation error associated with POISSON using (6.3). Consistent with this analysis, the numerical results for this example are excellent. When there is a large number of components with nearly equal means, great accuracy is obtained by all the methods with little effort. For example, this was the case for an E_{100} (Erlang of order 100) distribution. Accuracy to 10^{-12} was obtained with only 200 terms using POISSON. For this case, the exact result can be computed directly from a sum of 101 terms; see (3.5) on p. 11 of Feller [43].

A more interesting example is the convolution of 5 exponentials with mean 10 plus 50 exponentials with mean 1. In fact, this example presented difficulties for EULER, GAVER and POST-WIDDER with the parameter settings in §9. However, when the number of terms was increased from 15 to 30 in EULER and from 16 to 32 in GAVER, they agreed to at least 10^{-7}. For POST-WIDDER, the error was $10^{-7} - 10^{-8}$ with $m = 8$, $j = 20$ and $E = 8$. The additional terms in EULER and POST-WIDDER present no problem, but as indicated before, an increase in n in GAVER must be associated with higher precision. For $n = 32$, we required 48 decimal places, which tends to slow down the algorithm somewhat. Neither POISSON nor EULER were seriously stressed by this example, but GAVER was. It is not difficult to apply POISSON with detailed error analysis.

To quickly understand the nature of these distributions, note that the mean of the distribution in (12.1) is $\sum_{j=1}^{n} k_j m_j$ and the variance is $\sum_{j=1}^{n} k_j m_j^2$. As the number of components increases, without the mean of any one being large compared to the total mean, the distribution approaches a normal distribution by virtue of the Lindeberg version of the central limit theorem, p. 262 of Feller [43].

We close this section by mentioning that a special variant of the Fourier-series method (using the mid-point rule instead of the trapezoidal rule) was developed by Davies [31] to compute the distribution of a linear combination of independent χ^2 random variables, which includes exponentials as a special case.

13. Renewal Functions

The purpose of this section is to illustrate the numerical inversion of Laplace transforms of functions on the positive real line that are not cdf's or probability density functions. An example that is often encountered in queueing and applied probability is the renewal function; e.g., see Heffes and Lucantoni [55] and Heyman [56].

Let $H(t)$ be the renewal function, recording the expected number of renewals in the interval $(0, t)$, associated with a cdf $F(t)$ with $F(0) = 0$, defined here as

$$H(t) = \sum_{n=1}^{\infty} F^{n*}(t) \ , \ t \geq 0 \ , \tag{13.1}$$

where $F^{n*}(t)$ is the n-fold convolution of $F(t)$, as in Cox [26]. (Some authors have an extra term $F^{0*}(t) \equiv 1$ in (13.1); see (1.2) on p. 358 of Feller [43].) Let $\hat{f}(s)$ denote the LST of $F(t)$, i.e., $\hat{f}(s) = \int_0^{\infty} e^{-st} dF(t)$, which is the Laplace transform of the density $f(t)$ of $F(t)$ when $F(t)$ is absolutely continuous. Then the Laplace transform of $H(t)$ is

$$\hat{H}(s) = \int_0^{\infty} e^{-st} H(t)\, dt = \frac{\hat{f}(s)}{s(1 - \hat{f}(s))} \ , \tag{13.2}$$

as in (4) on p. 46 of Cox [26].

Thus, given the LST $\hat{f}(s)$, we can calculate $H(t)$ by numerically inverting $\hat{H}(s)$. EULER, GAVER and POST-WIDDER all can be applied. For EULER, we can use the known asymptotic behavior of $H(t)$ together with (5.28) to estimate the discretization error; in particular, we can apply

$$\lim_{t \to \infty} H(t) - \frac{t}{m} = \frac{(c^2 - 1)}{2} \ , \tag{13.3}$$

where m is the mean of $F(t)$ and c^2 is the squared coefficient of variation (variance divided by the square of the mean), see p. 58 of Cox [26]. We can also apply known bounds for special classes of distributions.

Assuming that $M(t) \approx t/m + (c^2 - 1)/2$, we obtain from (5.28) and (5.29) the estimate

$$|e_d| \approx \left[\frac{(c^2 - 1)}{2} + \frac{3t}{m} \right] e^{-A} \ , \tag{13.4}$$

from which we obtain an initial estimate of A. (If the method does not seem to be working, then we can try larger values of A.)

As a concrete example, we considered the case of an E_3 (Erlang of order 3) distribution, with mean m which has been analyzed analytically in (9) on p. 50 of Cox [26], yielding

$$H(t) = \frac{t}{m} - \frac{1}{3} + \frac{1}{3\sqrt{3}} e^{-9t/2m} (\sin(3\sqrt{3}\, t/2m) + \sqrt{3} \cos(3\sqrt{3}\, t/2m)) \ . \tag{13.5}$$

For this example, the three inversion methods EULER, GAVER and POST-WIDDER worked well, producing agreement to 10^{-8} with the same number of terms as in §9. (The parameter A was set so that $10^{-8} = (3t/m) e^{-A}$.)

14. Examples to Reveal the Limitations of the Fourier Series Method

The Fourier-series method is satisfactory for numerical inversion unless finite approximations of the infinite series in (4.4) or (4.6) are not sufficiently accurate. First, note that the step size h actually presents no difficulty, because given any cdf $F(t)$ yielding a step size h, we can scale (change the measuring units) by considering $F(t/y)$ to obtain an equivalent step size of h/y. Hence, it suffices to assume that h is of order 1. *The critical issue then is the truncation point* $T = Nh$, *which depends on the smoothness of the cdf F* (*in a scale for which h is approximately of order* 1).

The worst case occurs when F is discontinuous, i.e., when the probability distribution has atoms (point masses). *Since the approximate cdf is always continuous, even with an alternate* *method of summation, there is no N such that the error from the approximation is uniformly small in a neighborhood of a jump.* (We avoid this difficulty for lattice distribution in §5 by recognizing that F is lattice.)

Whenever F has an atom, $|Re\ \phi(u)|$ does not converge to 0 as $|u| \to \infty$, so that the integral for the truncation bound in (6.3) diverges. (This behavior of the cf holds for any choice of the measuring units.) Moreover, this difficulty will occur with continuous cdf's having a concentration of probability like a point mass, i.e, with a density having very high peaks (in the scale for which h is approximately of order 1). However, this difficulty is typically not so serious for practical applications, because we rarely need to know the cdf precisely in the neighborhood of a jump.

A Mixture of Two Out-of-Scale Distributions

A simple example to illustrate the possible difficulties is a mixture of two out-of-scale distributions. Thus, suppose that F is the cdf of the mixture of two exponential distributions, one with mean 10^{-6} and the other with mean 10^6, with each component having probability $1/2$. From (5.12), we see that we achieve a discretization error of 0.0025 for $h^{-1} = 10^6$ and $t = 1$. On the other hand, since

$$Re(\phi)(u) = \frac{1}{2(1 + 10^{12} u^2)} + \frac{1}{2(1 + 10^{-12} u^2)} \approx \frac{1}{2(1 + 10^{-12} u^2)} , \qquad (12.1)$$

we can apply (6.4) to obtain the estimated truncation error of 0.0032 when $h^{-1} = 10^6$, $t = 1$ and $N = 10^8$. Hence, we get overall accuracy of only about 0.006 at $t = 1$ with the large computation based on $N = 10^8$.

This example is easier to understand if we change the scale, e.g., so that the two exponential components have means 10^{-12} and 1 instead of 10^{-6} and 10^6. Then we recognize that in the new scale of order 1 the distribution is essentially the same as a mixture of a point mass at 0 and an exponential with mean 1. This distribution is essentially just as difficult to invert numerically as a simple point mass at 0, which we consider below. The difficulty is not great provided we do not consider t too close to 0. Certainly we have no difficulty if t is of order 1.

Translating this back to the original problem, we see that we have no serious difficulty there either, provided that we only seek moderate accuracy and consider t of order 10^6. Indeed, when we consider $x > h^{-1}$, the truncation error estimate in (6.4) becomes $h\beta(Nh)$ instead of $\beta(Nh)/t$. Since $e^{-8} = 3.3 \times 10^{-4}$, we achieve a discretization error of at most 3.3×10^{-4} in the original problem if we set $\pi/h - x = 8 \times 10^6$, by (5.13). For $t \leq 8 \times 10^6$, we have $h^{-1} = 2.5 \times 10^6$. Hence, for $2.5 \times 10^6 < t < 8 \times 10^6$, the estimated truncation error from (6.4) is $h\beta(Nh) \approx 1/\pi N$. For $N = 1000$, then, the estimated truncation error is 3.2×10^{-4} and the estimated overall error is 6.5×10^{-4}. As noted above, we run into serious difficulty only when we try to consider t much smaller than 10^6.

A Unit Point Mass

To continue exploring difficult cases for numerical inversion, we now consider the numerical inversion of the cdf of a unit point mass (a simple step function). To relate to §9, we put the unit point mass at $t = 6$.

If we apply POISSON in the framework of the whole line, then *we can choose h^{-1} arbitrarily small and obtain no discretization error.* To relate to the previous examples, we work instead on the positive real line, where we apply (5.13) to obtain no discretization error for $h^{-1} > (6 + t)/2\pi$. We consider $t \leq 8$, so we set $h^{-1} = 7/\pi$.

The difficulty, of course, is with the truncation. As noted before, the truncation error bound in (6.3) is not applicable. The estimated error from (6.4) for $t > 7/\pi$ is $h\beta(Nh) = 2/\pi N$. Of course, we know that this estimate is useless in the immediate neighborhood of $t = 6$, but we expect it to give a rough indication elsewhere.

Table 5 displays the errors for POISSON with $h^{-1} = 7/\pi$ and two cases of N: $N = 5000$ and $N = 100$. The estimated error in these two cases is 0.00013 and 0.0064, which is roughly correct for t not too near 6. The errors multiplied by 10^5 are displayed in Table 5.

Insert Table 5 here (or slightly later).

For this example we also consider convolution smoothing with the Gaussian window in §6 and the Hanning window in (10.7). For the Gaussian window, we apply (6.16)-(6.19) with $\varepsilon_2 = 2 \times 10^{-5}$, $h^{-1} = 7/\pi$ and N as above (100 and 5000). Thus, from (6.17) and (6.18), we obtain an overall 2×10^{-5} error bound for $|t - 6| > \varepsilon_1$ by setting $\varepsilon_1 = 2 \log (2/\varepsilon_2)/Nh \approx 67/N$ and $\sigma = \varepsilon_1/\sqrt{2 \log (2/\varepsilon_2)} \approx \varepsilon_1/5.5 \approx 12.2/N$. The final weighting function is $w(k, N) = (10^{-5})^{(k/N)^2}$. Table 5 shows that the actual performance of the convolution smoothing away from the point $t = 6$ is much better than these bounds indicate. Similar performance is seen with the Hanning window.

In addition, for this example we consider EULER, GAVER and POST-WIDDER. These methods produce results similar to those for POISSON without convolution smoothing, as shown for EULER and POST-WIDDER in Table 5. For the parameters used in §9-10, the accuracy is much less here, being about 10^{-4} at $t = 5$ and 7 instead of 10^{-7}. However, it is significant that the estimated truncation error in EULER is quite accurate. We also consider the variant of EULER based on (4.6a) and (5.30) instead of (4.6) and (5.27). Just as for other examples, its performance on this example is essentially the same as EULER.

Finally, we also apply algorithm LATTICE-POISSON for lattice distributions. Of course, it works very well. We would also have small error with convolution smoothing algorithms.

15. Literature Review

The Fourier-Series Method

The 1968 paper by Dubner and Abate [40] is widely recognized as the first to present the Fourier-series method for numerically inverting Laplace transforms, but in 1935 Koizumi [77] proposed a variant of the same method, as was only recently discovered by Squire [112]. Independently, in 1960-1972, Bohman [13], [14], [15] proposed variants of the same method for numerically inverting cf's, applying the Poisson summation formula to characterize the discretization error and convolution smoothing to reduce the truncation error. Between 1935 and 1970 others recognized the value of the trapezoidal rule and the Fourier-series method for calculating Fourier and related integrals. The Poisson summation formula is evidently due to Poisson [100] in 1823. We have remarked it plays a prominent role in Ramanujan's notebooks; see Berndt [11]. Clear presentation of its use were given by Fettis [44] in 1955 and De Balbine and Franklin [34] in 1966, who were apparently unaware of Fettis [44]. De Balbine and Franklin cite a related 1955 paper by Luke [81]. Fettis [44] was discovered by Rice [105], [106], who was apparently unaware of De Balbine and Franklin [34].

The Fourier-series method is probably best known through its use with the Fast Fourier Transform (FFT), which we discussed in §4. Key early references are Cooley and Tukey [25] and Cooley, Lewis and Welch [22], [23], [24]. References [22] and [23] relate specifically to the inversion problem. They understood the advantage of constructing the periodic function as in (5.1), but they did not use the Poisson summation formula and they did not exhibit the discretization error. See Rabiner and Gold [102] for further discussion about the FFT. We have not discussed the FFT much because there seems to be little need for it itself in standard numerical inversion applications. It may play a greater role in more computationally intensive inversions involving many functions and/or many time points, perhaps associated with elaborate graphics.

Dubner and Abate [40] rederived the Poisson summation formula for a damped function $e^{-at} f(t)$ on the positive real line in (5.25), thus obtaining the Fourier-series method for inverting Laplace transforms with explicit expression for the discretization error. Related literature on the Fourier-series method (in chronological order since 1969) include Nuttall [93], Brigham and Conley [16], Requicha [103], Silverberg [109], Ichikawa and Kishima [64], [65], Simon, Stroot and Weiss [110], Rice [105], [106], Veillon [122], [123], Davies [30], Durbin [41], Schorr [108], Crump [27], Mullineux and Reed [89], Hosono [58], [59], Kiefer and Weiss [73], De Hoog, Knight and Stokes [35], Hosono [60], Honig and Hirdes [57], Piessens and Huysmans [98], Squire [112], Carasso [17], Hsu and Dranoff [63], Platzman et al. [99], Kwok and Barthez [79] and Beaulieu [9]. Surprisingly, several papers, including recent ones such as Hosono [58], [59], [60], Platzman et al. [99], Kwok and Barthez [79] and Beaulieu [9], develop variants of the Fourier-series method without recognizing that they are in fact variants of the Fourier-series method, and thus without recognizing that there is a substantial body of related literature.

The Euler summation we have used to accelerate convergence of the nearly alternating series in (4.6) and (5.27) was first proposed in 1972 by Simon, Stroot and Weiss [110]. The nearly alternating series itself (with cosine terms replaced by ± 1) comes from (21) of Dubner and Abate [40]. Essentially this same procedure was proposed by Hosono [57], [58], [59]; see (3.3a), (4.6a) and (5.30). Hosono's [59] method was discovered by Bertsimas and Nakazato [12] and found to be very effective for queueing problems. Following Hosono [59], Kwok and Barthez [79] rederive (5.27).

Brigham and Conley [16], Crump [27], Kiefer and Weiss [73], De Hoog et al. [35], Honig and Hirdes [57] and Piessens and Huysmans [98] consider other methods for accelerating convergence. The method described here using Euler summation is appealing for its simplicity, especially when expressed as binomial averaging as in (6.26). The algorithms by Honig and Hirdes [57] and Piessens and Huysmans [98] require 550 and 740 lines of code, respectively, while the algorithm here in Figure 1 requires less than 50 lines of code.

In §6 we pointed out that convolution smoothing or windowing is intimately related to alternative methods of summation to accelerate convergence. There is a long history, as can be seen from Harris [53]. In the early inversion literature, Bohman [13], [14], [15] and Silverberg [109] applied convolution smoothing. Platzman et al. [99] present the convolution smoothing using the normal distribution with a nice computational complexity analysis, which we discussed in Remark 6.4.

An early Fourier-series method for inverting generating functions is due to Abate and Dubner [1], which was mentioned in Remark 5.7. It is applied in Dubner [39]. The direct Fourier-series method for inverting generating functions in §5 is proposed without error analysis by Cavers [19]. The full procedure was employed by Jagerman [67] in his algorithm for inverting Laplace transforms discussed in §8. Hosono [61] also develops the essentially equivalent algorithm involving the complementary form in (5.41). A variant of the Fourier-series method without the error-controlling damping constant r was recently suggested by Daigle [28]. Other earlier work related to inverting generating functions is contained in Lyness and Moler [84], Lyness [82] and Fornberg [45]. Another application is Mills [87].

Other Methods

Of course, the Fourier-series method is *only one of many methods for performing numerical transform inversion.* An extensive bibliography on the numerical inversion of Laplace transforms was completed by Piessens [96] and Piessens and Dang [97]. Davies and Martin [29] systematically compare fourteen different methods for inverting Laplace transforms, including three versions of the Fourier-series method (Dubner and Abate [40], Silverberg [109] and

Crump [27]). The comparison of fourteen methods on sixteen test functions seems very laudable, but we doubt that all methods were equally well implemented. From the numerical results in [29], it appears that the last test function, $t^{-1} \sin t$, can perhaps be considered most difficult. The stated results for the three Fourier-series methods on this example were not good, but we easily obtained accuracy to 10^{-7} using the algorithms EULER, GAVER and POST-WIDDER. (As for the examples in §9-13, the Gaver-Stehfest method required high precision to work.)

At AT&T Bell Laboratories, a method for inverting Laplace transforms developed by M. Eisenberg (related to Weeks [124], which he helped develop) and the algorithm JAGERMAN in §8, [66] and [67] have been used quite frequently for probability applications; e.g., see Doshi [37], Doshi and Kaufman [38], Heffes and Lucantoni [55], and Heyman [56]. (The last two applications involve the renewal function, as in §13.) The Weeks method is also an excellent candidate to use as one of two methods. Indeed, Davies and Martin [29] found the Weeks [124] method to be the most accurate of the methods they considered. A variant of the Weeks [124] method is now available from the ACM library of software algorithms in Algorithm 662 by Garbow et al. [47]. The algorithm JAGERMAN in §8 is only one of many routines in Jagerman's [68] MATHCALC library. Even though the direct Fourier-series method (e.g., the algorithms POISSON and EULER) is not used in this library for inverting transforms, extensive use is made of the trapezoidal rule, the Poisson summation formula and the Whittaker cardinal function in (4.7). Hence, the present paper is in the same spirit as [68] and the algorithms here can easily be implemented using [68].

The Gaver [49]-Stehfest [113] method for inverting Laplace transforms in §8 has also been used with success for probability applications by Gaver [50], Nance, Bhat and Claybrook [90], Middleton [86] and Abate and Whitt [2], [3], [4]. The Gaver-Stehfest algorithm is especially convenient for *implicitly* defined Laplace transforms, as occurs with the M/G/1 busy-period cdf (p. 212 of Kleinrock [75]), because it works with the transform evaluated only at *real arguments*.

(However, iterative methods also seem to work with complex arguments.) However, as noted in §8, the Gaver-Stehfest procedure requires high precision.

There are five methods for inverting Laplace transforms available from the ACM library of software algorithms: (1) Algorithm 368 by Stehfest [113], (2) Algorithm 486 by Veillon [123] (3) Algorithm 619 by Piessens and Huysmans [98], (4) Algorithm 662 by Garbow et al. [47] and (5) Algorithm 872 by Murli and Rizzardi [69]. Of course, Algorithm 368 is the Gaver-Stehfest method. Algorithms 486 and 619 are variants of the Fourier-series method. Algorithm 662 is a variant of the Weeks [124] method; and Algorithm 682 is Talbot's [116] method. (Talbot's method is the trapezoidal rule on a very special contour applied to the Bromwich inversion integral (3.3).) The program line counts for Algorithms 619, 662 and 682 are 740, 4540 and 3090, respectively. These program line counts might suggest that inversion is necessarily difficult. The versions of the Fourier-series method EULER, POISSON, LATTICE-POISSON and POST-WIDDER presented here are elementary to program directly, as illustrated by the displays. While there are many possibilities for sophisticated refinements, we believe that methods here, which are easy to understand and implement, are adequate for most queueing applications.

Acknowledgments. We thank Dimitris Bertsimas, Bharat Doshi, Kerry Fendick, Daniel Heyman, Toshio Hosono, David Jagerman, John Morrison, Henk Tijms, Bill Turin and the referees for assistance. In particular, we thank Kerry Fendick for writing C++ programs implementing the algorithms EULER, POST-WIDDER and LATTICE-POISSON.

References

[1] J. Abate and H. Dubner, A new method for generating power series expansions of functions, *SIAM J. Numer. Anal.* 5 (1968) 102-112.

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

[3] J. Abate and W. Whitt, Transient behavior of the M/M/1 queue via Laplace transforms, *Adv. Appl. Prob.* 20 (1988) 145-178.

[4] J. Abate and W. Whitt, Approximations for the M/M/1 busy-period distribution, In *Queueing Theory and its Applications, Liber Amicorum for J. W. Cohen*, O. J. Boxma and R. Syski (eds), (North-Holland, Amsterdam, 1988), 149-191.

[5] J. Abate and W. Whitt, Simple spectral representations for the M/M/1 queue, *Queueing Systems* 3 (1988) 321-346.

[6] J. Abate and W. Whitt, Numerical inversion of Laplace transforms of probability distributions, AT&T Bell Laboratories, Murray Hill, NJ, 1991.

[7] J. Abate and W. Whitt, Numerical inversion of probability generating functions, AT&T Bell Laboratories, Murray Hill, NJ, 1991.

[8] M. Abramowitz and I. A. Stegun, *Handbook of Mathematical Functions,* (National Bureau of Standards, Washington, D.C., 1972).

[9] N. C. Beaulieu, An infinite series for the computation of the complementary probability distribution function of a sum of independent random variables and its application to the sum of Rayleigh random variables, *IEEE Transactions on Comm.* 38 (1990) 1463-1474.

[10] R. Bellman, R. E. Kalaba and J. Lockett, *Numerical Inversion of the Laplace Transform. Application to Biology, Economics, Engineering and Physics,* (American Elsevier, New

York, 1966).

[11] B. C. Berndt, *Ramanujan's Notebooks*, Part II (Springer, New York, 1989).

[12] D. Bertsimas and D. Nakazato, Transient and busy period analysis for the GI/G/1 queue; the method of stages, *Queueing Systems,* to appear.

[13] H. Bohman, A method to calculate the distribution function when the characteristic function is known, *Ark. Mat.* 4 (1960) 99-157.

[14] H. Bohman, A method to calculate the distribution function when the characteristic function is known, *BIT* 10 (1970) 237-242.

[15] H. Bohman, From characteristic function to distribution function via Fourier analysis. *BIT* 12 (1972) 279-283.

[16] E. O. Brigham and R. E. Conley, Evaluation of cumulative probability distribution functions: improved numerical methods, *IEEE Proc.* 58 (1970) 1367-1368.

[17] A. S. Carasso, Infinitely divisible pulses, continuous deconvolution, and the characterization of linear time invariant systems, *SIAM J. Appl. Math.* 47 (1987) 892-927.

[18] H. S. Carslaw, *Introduction to the Theory of Fourier's Series and Integrals,* third edition (Dover, New York, 1930).

[19] J. K. Cavers, On the fast Fourier transform inversion of probability generating functions, *J. Inst. Math. and Appl.* 22 (1978) 275-282.

[20] D. C. Champeney, *A Handbook of Fourier Theorems* (Cambridge University Press, New York, 1987).

[21] K. L. Chung, *A Course in Probability Theory,* Second edition (Academic Press, New York, 1974).

[22] J. W. Cooley, P. A. W. Lewis and P. D. Welch, Application of the fast Fourier transform to the computation of Fourier integrals, Fourier-series, and convolution integrals, *IEEE Transactions* AU-15 (1967) 79-84.

[23] J. W. Cooley, P. A. W. Lewis and P. D. Welch, Historical notes on the fast Fourier transform, *Proc. IEEE* 55 (1967) 1675-1677.

[24] J. W. Cooley, P. A. W. Lewis and P. D. Welch, The fast Fourier Transform algorithm: programming considerations of sine, cosine and Laplace transforms, *J. Sound Vib.* 12 (1970) 315-337.

[25] J. W. Cooley, and J. W. Tukey, An Algorithm for the machine computation of complex Fourier-series, *Math. Comp.* 19 (1965) 297-301.

[26] D. R. Cox, *Renewal Theory* (Methuen, London, 1962).

[27] K. S. Crump, Numerical inversion of Laplace transforms using a Fourier-series approximation, *J. ACM* 23 (1976) 89-96.

[28] J. N. Daigle, Queue length distributions from probability generating functions via discrete Fourier transforms, *Oper. Res. Letters* 8 (1989) 229-236.

[29] B. Davies and B. L. Martin, Numerical inversion of Laplace transforms: a critical evaluation and review of methods, *J. Comp. Phys.* 33 (1970) 1-32.

[30] R. B. Davies, Numerical inversion of a characteristic function, *Biometrika* 60 (1973) 415-417.

[31] R. B. Davies, The distribution of a linear combination of χ^2 random variables, *Appl. Stat.* **29** (1980) 323-333.

[32] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration,* Second edition, (Academic Press, New York, 1984).

[33] M. A. B. Deakin, Euler's version of the Laplace transform, *Amer. Math. Monthly* 87 (1980) 264-269.

[34] G. de Balbine and J. Franklin, The calculation of Fourier integrals, *Math. Comp.* 20 (1966) 570-589.

[35] F. R. de Hoog, J. H. Knight and A. N. Stokes, An improved method for numerical inversion of Laplace transforms, *SIAM J. Sci. Stat. Comput.* 3 (1982) 357-366.

[36] G. Doetsch, *Introduction to the Theory and Application of the Laplace Transformation,* (Springer-Verlag, New York, 1974).

[37] B. T. Doshi, Analysis of clocked schedules – high priority tasks, *AT&T Tech. J.* 64 (1985) 633-659.

[38] B. T. Doshi and J. Kaufman, Sojourn Times in an M/G/1 Queue with Bernoulli Feedback. In *Queueing Theory and Its Applications, Liber Amicorum for J. W. Cohen*, O. J. Boxma and R. Syski (eds.) (North-Holland, Amsterdam, 1988).

[39] H. Dubner, Partitions approximated by finite cosine series, *Math. Computation,* to appear.

[40] H. Dubner and J. Abate, Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform, *J. ACM* 15 (1968) 115-123.

[41] F. Durbin, Numerical inversion of Laplace transforms: An efficient improvement to Dubner and Abate's Method, *Comput. J.* 17 (1974) 371-376.

[42] W. Feller, *An Introduction to Probability Theory and its Applications,* Vol. I, Third edition (Wiley, New York, 1968).

[43] W. Feller, *An Introduction to Probability Theory and its Applications*, Vol. II, Second edition (Wiley, New York, 1971).

[44] H. E. Fettis, Numerical calculation of certain definite integrals by Poisson's summation formula, *Math. Tables and Other Aids to Computation* 9 (1955) 85-92.

[45] B. Fornberg, Numerical differentiation of analytic functions, *ACM Trans. Math. Software* 7 (1981) 512-526.

[46] J. Foster and F. B. Richards, The Gibbs phenomenon for piecewise-linear approximations, *Amer. Math. Monthly* 98 (1991) 47-49.

[47] B. S. Garbow, G. Giunta, J. N. Lyness and A. Murli, Algorithm 662, A FORTRAN software package for the numerical inversion of the Laplace transform based on Weeks' method, *ACM Trans. Math. Software* 14 (1988) 171-176.

[48] W. Gautschi, On the condition of a matrix arising in the numerical inversion of the Laplace transform, *Math. of Computation* 23 (1969) 109-118.

[49] D. P. Gaver, Jr., Observing stochastic processes and approximate transform inversion, *Operations Res.* 14 (1966) 444-459.

[50] D. P. Gaver, Jr., (1968) Diffusion approximations and models for certain congestion problems, *J. Appl. Prob.* 5 (1968) 607-623.

[51] J. Gil-Palaez, Note on the Inversion Theorem, *Biometrika* 38 (1951) 481-2.

[52] I. J. Good, Analogs of Poisson's sum formula, *Amer. Math. Monthly* 69 (1962) 259-266.

[53] F. J. Harris, On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform. *Proceedings of the IEEE* 66 (1978) 51-83.

[54] P. G. Harrison, Laplace transform inversion and passage-time distributions in Markov processes, *J. Appl. Prob.* 27 (1990) 74-87.

[55] H. Heffes, and D. M. Lucantoni, A Markov modulated characterization of packetized

voice and data traffic and related statistical multiplexer performance, *IEEE J. Sel. Areas Commun.* SAC-4 (1986) 856-868.

[56] D. P. Heyman, Mathematical models of database degradation, *ACM Trans. Database Sys.* 7 (1982) 615-631.

[57] G. Honig and U. Hirdes, A method for the numerical inversion of Laplace transforms, *J. Comput. Appl. Math.* 10 (1984) 113-129.

[58] T. Hosono, Numerical inversion of Laplace transform, *J. Inst. Elec. Eng. Jpn.* 54-A64 (1979) 494 (in Japanese).

[59] T. Hosono, Numerical inversion of Laplace transform and some applications to wave optics, *Radio Science* 16 (1981) 1015-1019.

[60] T. Hosono, *Fast Inversion of Laplace Transform by BASIC* (Kyoritsu Publishers, Japan, 1984, in Japanese).

[61] T. Hosono, Numerical algorithm for Taylor series expansion, *Electronics and Communications in Japan* 69 (1986) 10-18.

[62] T. Hosono, K. Yuda and A. Itoh, Analysis of transient response of electromagnetic waves scattered by a perfectly conducting sphere. The cases of back- and forward-scattering, *Electronics and Communications in Japan* 71 (1988) 74-86.

[63] J. T. Hsu and J. S. Dranoff, Numerical inversion of certain Laplace transforms by the direct application of fast Fourier transform (FFT) algorithm, *Comput. & Chem. Engng.* 11 (1987) 101-110.

[64] S. Ichikawa and A. Kishima, Application of Fourier-series technique to inverse Laplace transform (Part I), *Mem. Fac. Eng. Kyoto U.* 34 (1972) 53-67.

[65] S. Ichikawa and A. Kishima, Application of Fourier-series technique to inverse Laplace transform (Part II), *Mem. Fac. Eng. Kyoto U.* 35 (1973) 393-400.

[66] D. L. Jagerman, An inversion technique for the Laplace transform with applications, *Bell System Tech. J.* 57 (1978) 669-710.

[67] D. L. Jagerman, An inversion technique for the Laplace transform, *Bell System Tech. J.* 61 (1982) 1995-2002.

[68] D. L. Jagerman, MATHCALC, AT&T Bell Laboratories, Holmdel, NJ, 1987.

[69] D. L. Jagerman, The approximation sequence of the Laplace transform, AT&T Bell Laboratories, Holmdel, NJ, 1989.

[70] R. Johnsonbaugh, Summing an alternating series, *Amer. Math. Monthly* 86 (1979) 637-648.

[71] J. Keilson, Exponential spectra as a tool for the study of single-server systems, *J. Appl. Prob.* 15 (1978) 162-170.

[72] D. G. Kendall, A summation formula for finite trigonometric integrals, *Quart. J. Math.* 13 (1942) 172-184.

[73] J. E. Kiefer and G. H. Weiss, A comparison of two methods for accelerating the convergence of Fourier-series, *Comp. & Maths. with Appls.* 7 (1981) 527-535.

[74] Y. Kida, UBASIC Version 8.12, Faculty of Science, Kanazawa University, 1-1 Marunouchi, Kanazawa 920, JAPAN, 1990.

[75] L. Kleinrock, *Queueing Systems, Vol. 1: Theory*, (Wiley, New York, 1975).

[76] H. Kobayashi, *Modeling and Analysis* (Addison-Wesley, Reading Mass., 1978).

[77] S. Koizumi, A new method of evaluation of the heavyside operational expression by

Fourier-series, *Phil. Mag.* 19 (1935) 1061-1076.

[78] V. I. Krylov, and N. S. Skoblya, *A Handbook of Methods of Approximate Fourier Transformation and Inversion of the Laplace Transformation* (Mir Publishers, Moscow, 1977).

[79] Y. K. Kwok and D. Barthez, An algorithm for the numerical inversion of Laplace transforms, *Inverse Problems* 5 (1989) 1089-1095.

[80] E. Lukacs, *Characteristic Functions*, Second ed. (Hafner, New York, 1970).

[81] Y. L. Luke, Simple formulas for the evaluation of some higher transcendental functions, *J. Math. and Phys.* 34 (1955) 298-307.

[82] J. N. Lyness, Differentiation formulas for analytic functions, *Math. Comp.* 22 (1968) 352-356.

[83] J. N. Lyness and G. Giunta, A modification of the weeks method for numerical inversion of the Laplace transform, *Math. Comp.* 47 (1986) 313-322.

[84] J. N. Lyness and C. B. Moler, Numerical differentation of analytic functions, *SIAM J. Numer. Anal.* 4 (1967) 202-210.

[85] W. F. Magnus, F. Oberhettinger and R. P. Soni, *Formulas and Theorems for the Special Functions of Mathematical Physics* (Springer Verlag, New York, 1966).

[86] M. R. Middleton, *Transient Effects in M/G/1 Queues* (Ph.D. dissertation, Stanford University, 1979).

[87] P. L. Mills, Numerical inversion of *z*-transforms with application to polymer kinetics, *Comp. Chem.* 2 (1987) 137-151.

[88] A. Murli and M. Rizzardi, Algorithm 682, Talbot's Method for the Laplace inversion

problem, *ACM Trans. Math. Software* 16 (1990) 158-168.

[89] N. Mullineux and J. R. Reed, Numerical inversion of integral transforms, *Comp. and Math. with Appls.* 3 (1977) 299-306.

[90] R. E. Nance, U. N. Bhat and B. G. Claybrook. Busy period analysis of a time sharing system: transform inversion, *J. ACM* 19 (1972) 453-463.

[91] I. P. Natanson, *Constructive Function Theory, Vol. I, Uniform Approximation* (F. Ungar, New York, 1964).

[92] W. D. Neumann, UBASIC: A public-domain BASIC for mathematics, *Notices Amer. Math. Soc.* 36 (1989) 557-559.

[93] A. H. Nuttall, Numerical evaluation of cumulative probability distribution functions directly from characteristic functions, *IEEE Proc.* 57 (1969) 2071-2072.

[94] F. Oberhettinger, *Fourier Transforms of Distributions and Their Inverses* (Academic, New York, 1973).

[95] W. C. Obi, LAPLACE – A performance analysis library (PAL) module, AT&T Bell Laboratories, Holmdel, NJ, 1987.

[96] R. Piessens, A bibliography on numerical inversion of the Laplace transform and its applications, *J. Comput. Appl. Math.* 1 (1975) 115-128.

[97] R. Piessens, and N. D. P. Dang, A bibliography on numerical inversion of the Laplace transform and its applications: A supplement, *J. Comput. Appl. Math.* 2 (1976) 225-228.

[98] R. Piessens and R. Huysmans, Algorithm 619. Automatic numerical inversion of the Laplace transform, *ACM Trans. Math. Softw.* 10 (1984) 348-353.

[99] L. K. Platzman, J. C. Ammons and J. J. Bartholdi, III. A simple and efficient Algorithm

to compute tail probabilities from transforms, *Operations Res.* 36 (1988) 137-144.

[100] S. D. Poisson, Mémoire sur le Calcul Numérique des Integrales Défines, *Mem. Acad. Sci. Inst. France* 6 (1823) 571-602.

[101] E. L. Post, Generalized differentiation, *Trans. Amer. Math. Soc.* 32 (1930) 723-781.

[102] L. R. Rabiner and B. Gold, *Theory and Application of Digital Signal Processing*. (Prentice-Hall, Englewood Cliffs, New Jersey, 1975).

[103] A. A. G. Requicha, Direct computation of distribution functions from characteristic functions using the fast Fourier transform, IEEE Proc. 58 (1970) 1154-1155.

[104] J. Riordan, *Stochastic Service Systems* (Wiley, New York, 1962).

[105] S. O. Rice, Efficient evaluation of integrals of analytic functions by the trapezoidal rule, *Bell System Tech. J.* 52 (1973) 707-722.

[106] S. O. Rice, Numerical evaluation of integrals with infinite limits and oscillating integrands, *Bell System Tech. J.* 54 (1975) 155-164.

[107] S. Ross, *Stochastic Processes*, (Wiley, New York, 1983).

[108] B. Schorr, Numerical inversion of a class of characteristic functions, *BIT* 15 (1975) 94-102.

[109] M. Silverberg, Improving the efficiency of Laplace-transform Inversion for network analysis, *Electronics Letters* 6 (1970) 826-106.

[110] R. M. Simon, M. T. Stroot and G. H. Weiss, Numerical inversion of Laplace transforms with applications to percentage labeled experiments, *Comput. Biomed. Res.* 6 (1972) 596-607.

[111] W. L. Smith, On the distribution of queueing times, *Proc. Camb. Phil. Soc.* 49 (1953)

449-461.

[112] W. Squire, The numerical treatment of Laplace transforms: the Koizumi inversion method, *Int. J. Num. Meth. in Eng.* 20 (1984) 1697-1702.

[113] H. Stehfest, Algorithm 368. Numerical inversion of Laplace transforms, *Comm. ACM* 13 (1970) 479-49 (erratum 13, 624).

[114] F. Stenger, (1981) Numerical methods based on Whittaker cardinal, or sine functions, *SIAM Review* 23 (1981) 165-224.

[115] D. Stoyan, *Comparison Methods for Queues and Other Stochastic Models* (Wiley, Chichester, 1983).

[116] A. Talbot, The accurate numerical inversion of Laplace transforms, *J. Inst. Math. Appl.* 23 (1979) 97-120.

[117] D. ter Haar, An easy approximate method of determining the relaxation spectrum of a viscoelastic material, *J. Polymer Sci.* 6 (1951) 247-250.

[118] H. C. Tijms, *Stochastic Modelling and Analysis: A Computational Approach* (Wiley, Chichester, 1986).

[119] G. P. Tolstov, *Fourier Series*, (Dover, New York, 1976).

[120] B. Van Der Pol and H. Bremmer, *Operational Calculus*, (Cambridge Press, 1955; reprinted, Chelsea, New York, 1987).

[121] J. M. Varah, Pitfalls in the numerical solution of linear ill-posed problems, *SIAM J. Sci. Stat. Comput.* 4 (1983) 164-176.

[122] F. Veillon, Une nouvelle methode de calcul de la transformee inverse d'une fonction au sens de Laplace et de la deconvolution de deux fonctions, *R.A.I.R.O.* 6 (1972) 91-98.

[123] F. Veillon, Algorithm 486. Numerical inversion of Laplace transform, *Comm. ACM 17*, (1974) 587-589.

[124] W. T. Weeks, Numerical inversion of Laplace transforms using Laguerre functions, *J. ACM.* 13 (1966) 419-426.

[125] D. V. Widder, The inversion of the Laplace integral and the related moment problem, *Trans. Amer. Math. Soc.* 36 (1934) 107-200.

CONTENTS

1. Introduction 4
2. The Transforms: Definitions and Basic Properties 8
3. Inversion Integrals 11
4. Numerical Integration 15
5. The Poisson Summation Formula 20
6. Truncation, Convolution Smoothing and Alternative Methods of Summation . . . 38
7. Summaries of Variants of the Fourier-Series Method 55
8. Methods Based on the Post-Widder Formula 60
9. Examples: The M/G/1 Waiting Time CDF 67
10. The Time-Dependent Mean of RBM 78
11. The Number Served in an M/M/1 Busy Period 82
12. Convolutions of Exponential Distributions 83
13. Renewal Functions 86
14. Examples to Reveal the Limitations of the Fourier Series Method 87
15. Literature Review 91

Prescribed Error ε	Discretization h^{-1}		Truncation $T = Nh$		Number of Terms N for Error of 2ε	
	Estimate (9.6)	Chebychev Bound (9.10)	Standard Estimate (9.16) with $t = 6$	Bound (9.17)	Two Estimates	Two Bounds
10^{-2}	9.4	15.8	1.3	2.5	13	40
10^{-3}	11.7	37.6	2.7	7.8	32	294
10^{-4}	14.0	107	5.9	24.5	83	2,630
10^{-5}	16.4	324	12.6	77.3	207	25,050
10^{-6}	18.7	1013	27.1	245	507	249,000
10^{-7}	21.1	3190	58.4	773	1232	2,470,000
10^{-8}	23.4	10,075	126	2450	2949	24,700,000

Table 1. Required discretization parameter h^{-1} and truncation parameter $T = Nh$ to achieve prescribed errors of 10^{-k} (overall error of 2×10^{-k}) as a function of k for the $M/H_2/1$ waiting-time example in Section 9 using algorithm POISSON based on estimates and bounds.

t	$F^c(t)$ exact (9.3)	asymptotic approximation (9.5)	$\varepsilon = 2 \times 10^{-7}$ $h^{-1} = 21.1$		$\varepsilon = 2 \times 10^{-5}$ $h^{-1} = 16.4$		$\varepsilon = 2 \times 10^{-3}$ $h^{-1} = 11.7$	
			$t = 1$ $N = 2238$	$t = 6$ $N = 1232$	$t = 1$ $N = 375$	$t = 6$ $N = 206$	$t = 1$ $N = 58$	$t = 6$ $N = 32$
0.1	0.97589492	0.921	−6	+34	+1	+26	+1	+3
0.3	0.93237079	0.892	+3	+2	+3	−15	+2	+6
0.5	0.89383953	0.865	−2	−7		+10		+4
1.0	0.81254551	0.799	+1	−2	−1	+5	−1	−2
1.5	0.74498610	0.739	−1	+4	−1	+3		−2
2.0	0.68595171	0.6833		−2		+3		+1
4.0	0.49932501	0.49921			−1	+1		
6.0	0.36474142	0.364737		+1		+1		−1
9.0	0.22778160	0.22778156				+1		
12.0	0.14225171	0.14225171				+1		
15.0	0.08883752	0.08883752						
18.0	0.05547986	0.05547986		−1				+1
24.0	0.02163781	0.02163781		−1	−1		−1	−1
30.0	0.00843900	0.00843900	−1	−1	−1	−1	−1	−1

Table 2. A comparison of approximations obtained by numerical inversion using POISSON with exact values for the $M/H_2/1$ complementary conditional waiting-time cdf $F^c(t)$ in Section 9. For the numerical inversions, the displayed numbers are the errors multiplied by 10^k when the prescribed error is 2×10^{-k}; no entry means less than 0.5.

t	$F^c(t)$ exact (9.21)	asymptotic approximation (9.26)	$\varepsilon = 2 \times 10^{-7}$ $h^{-1} = 20.6$		$\varepsilon = 2 \times 10^{-5}$ $h^{-1} = 16.1$		$\varepsilon = 2 \times 10^{-3}$ $h^{-1} = 11.5$	
			$t = 1$ $N = 4729$	$t = 6$ $N = 2309$	$t = 1$ $N = 585$	$t = 6$ $N = 285$	$t = 1$ $N = 66$	$t = 6$ $N = 32$
0.1	0.9784447	0.948	–6		–6	+14	+2	+4
0.3	0.9408811	0.918	+3	–8	–1	+3	+1	+4
0.5	0.9068208	0.889		+8	+1	–6	–1	+2
1.0	0.8305714	0.820	–1	+1		+2		–2
1.5	0.7630251	0.756		–1		+1		–1
2.0	0.7020169	0.697	+1	–1	–1	–2		+1
4.0	0.5060261	0.5050						
6.0	0.3659234	0.36566		+1		+1		+1
9.0	0.2253310	0.22529						+1
12.0	0.1388133	0.138807				+1		
15.0	0.0855228	0.0855217						
18.0	0.0526919	0.0526917	–1	–1				
24.0	0.0200020	0.0200020	–1	–1				
30.0	0.0075928	0.0075928	–1	–1	–1	–1		–2

Table 3. A comparison of approximations obtained by numerical inversion using POISSON with exact values for the $M/\Gamma_{1/2}/1$ complementary conditional waiting-time cdf $F^c(t)$ in Section 9. For the numerical inversions, the displayed numbers are the errors multipled by 10^k when the prescribed error is 2×10^{-k}; no entry means less than 0.5.

t	$F^c(t)$ exact	$N = 2000$, $h^{-1} = 4.41$			$N = 500$, $h^{-1} = 4.41$		
		POISSON (4.4)	convolution smoothing: Gaussian window (6.19)	convolution smoothing: Hanning window (10.7)	POISSON (4.4)	convolution smoothing: Gaussian window (6.19)	convolution smoothing: Hanning window (10.7)
0.1	0.5870048	–22	+60	+13	+391	+1277	+197
0.3	0.3829011	+18	+10	+2	+19	+168	+36
0.5	0.2798589	–12	+4	+1	–69	+70	+15
0.7	0.2143120	+6	+3	+1	–67	+38	+8
1.0	0.1506796	+6	+1		+1	+19	+4
1.5	0.0900794	–3			+22	+8	+2
2.0	0.0567901	+2			–24	+4	+1
3.0	0.0246974	+1				+1	
4.0	0.0115375				+12		
5.0	0.0056341					+1	
6.0	0.0028368	–1			–9		
7.0	0.0014610	–1					
8.0	0.0007656	–1	–1	–1	+6	–1	–1
9.0	0.0004069	–1	–1	–1	–1	–1	–1
10.0	0.0002187	–1	–1	–1	–7	–1	–1

Table 4. A comparison of three numerical inversion approximations based on POISSON with the exact values of the RBM complementary cdf $F^c(t)$ in §10. For the numerical inversions, the displayed numbers are the errors multiplied by 10^5; no entry means less than 0.5.

	$N = 5000$			$N = 100$				$N = 16$
		convolution smoothing			convolution smoothing			
t	POISSON (4.4)	Gaussian window (6.19)	Hanning window (10.7)	POISSON (4.4)	Gaussian window (6.19)	Hanning window (10.7)	EULER $A = 7$ (4.6)	GAVER (8.6) and (8.9)
0.001	–2			–6				
0.01	+1			–61				
0.1	+3			+138				
1.0	–3			–137				
3.0	+2			+111				−710
5.0	–1			–74		+1	+48	x
5.7	–21			–1145	–251	+51	−483	x
5.9	+23			+2269	x	–1787	−3773	x
5.99	+1292		–13	x	x	x	x	x
5.999	+4224	x	x	x	x	x	x	x
6.001	–4230	x	x	x	x	x	x	x
6.01	–1300		+13	x	x	x	x	x
6.1	–24			–2332	x	+1787	+3520	x
6.3	+25			+1338	+251	–51	–2201	x
6.5	+25			–1273		+13	+1146	x
6.7	+26			+1267		–5	−971	+3765
7.0							−534	+4179
7.5	+25			+1273		–13	−183	+6408

Table 5. A comparison of several numerical inversion approximations of the complementary cdf of a unit point mass at $t = 6$, as discussed in §14. The displayed numbers are the errors multiplied by 10^5; no entry means less than 0.5 in this scale. An x means greater than 10^4. The computations involving POISSON are based on $h^{-1} = 7/\pi$.

Appendices

to

THE FOURIER-SERIES METHOD FOR INVERTING TRANSFORMS

OF PROBABILITY DISTRIBUTIONS

by

Joseph Abate and Ward Whitt

Appendix A. Applying the Gil-Palaez Inversion Formula

In this appendix we indicate how to apply the Poisson summation formula to the Gil-Palaez inversion formula (3.5). For an even function γ, we have from (2) of Fettis (1955) that

$$\int_0^\infty \gamma(u)\, du = h\left[\frac{\gamma(0)}{2} + \sum_{k=1}^\infty \gamma(kh)\right] - 2\sum_{k=1}^\infty g\left[\frac{2\pi k}{h}\right], \tag{A-1}$$

where

$$g(t) = \int_0^\infty \gamma(u) \cos tu\, du . \tag{A-2}$$

We can apply this to (3.5) in the form

$$\frac{1}{2} - F(t) = \int_0^\infty \gamma(u)\, du \tag{A-3}$$

with

$$\gamma(u) = [\cos ut\ Im(\phi)(u) - \sin ut\ Re(\phi)(u)]/\pi u , \tag{A-4}$$

because γ in (A-4) is an even function. Moreover, by applying the trigonometric identities

$$2 \sin x \cos y = \sin (x-y) + \sin (x+y) \quad \text{and} \quad 2 \cos x \cos y = \cos (x+y) + \cos (x-y), \tag{A-5}$$

we see that the discretization error simplifies; in particular,

$$e_d = -2\sum_{k=1}^\infty g\left[\frac{2\pi k}{h}\right] = 2\sum_{k=1}^\infty \left[1 - F\left[t + \frac{2\pi k}{h}\right] - F\left[t - \frac{2\pi k}{h}\right]\right]$$

$$= 2\sum_{k=1}^\infty F^c\left[t + \frac{2\pi k}{h}\right] - 2\sum_{k=1}^\infty F\left[t - \frac{2\pi k}{h}\right]. \tag{A-6}$$

Unfortunately, in this case it is not as easy to bound or estimate the resulting discretization error as it was in Sections 5.2 and 5.3. Of course, if

$$F(t) \le ae^{-b|t|} \quad \text{for} \quad t \le t_1 \le 0 \quad \text{and} \quad F^c(t) \le ce^{-dt} \quad \text{for} \quad t \ge t_2 \ge 0 , \tag{A-7}$$

then

$$|e_d| \le \max\left\{\frac{2ae^{-b(t+2\pi/h)}}{1 - e^{-b2\pi/h}},\ \frac{2ce^{-d(t+2\pi/h)}}{1 - e^{-d2\pi/h}}\right\} \tag{A-8}$$

provided that $t - 2\pi/h \le t_1$ and $t + 2\pi/h \ge t_2$.

	$N = 5000$	$N = 100$	$N = 15,\ m = 11$	$m =$

t	POISSON (4.4)	convolution smoothing: Gaussian window (6.19)	convolution smoothing: Hanning window (10.7)	POISSON (4.4)	convolution smoothing: Gaussian window (6.19)	convolution smoothing: Hanning window (10.7)	EULER $A = 7$ (4.6)	EULER $A = 7$ (4.6a)	POST- ((
0.001	–2			–6					
0.01	+1			–61					
0.1	+3			+138					
1.0	–3			–137					
3.0	+2			+111					
5.0	–1			–74		+1	+48	+86	
5.7	–21			–1145	–251	+51	–483	–371	
5.9	+23			+2269	x	–1787	–3773	–3592	
5.99	+1292		–13	x	x	x	x	x	
5.999	+4224	x	x	x	x	x	x	x	
6.001	–4230	x	x	x	x	x	x	x	
6.01	–1300		+13	x	x	x	x	x	
6.1	–24			–2332	x	+1787	+3520	+3409	
6.3	+25			+1338	+251	–51	–2201	–2247	

6.5	+25			−1273		+13	+1146	+1258	
6.7	+26			+1267		−5	−971	−1057	
7.0							−534	−666	
7.5	+25			+1273		−13	−183	−339	

Table 5. A comparison of several numerical inversion approximations of the complementary cdf of a uni mass at $t = 6$, as discussed in §14. The displayed numbers are the errors multiplied by 10^5; no entry mea than 0.5 in this scale. An x means greater than 10^4. The computations involving POISSON are ba $h^{-1} = 7/\pi$.

```
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 (1.1)
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=19,1
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)ˆN*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)ˆN*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)
```

```
1   ’The Algorithm TRIGAMMA
2   ’
3   ’A variant of the Fourier-series method
4   ’for lattice distributions using the trigamma function,
5   ’applied to the cdf of the number of customers
6   ’served in an M/M/1 busy period
7   ’
20  input "Lattice Point=";N
21  M=150;N=N+1
```

```
22  H=#pi/M
23  U=2*M*#piˆ2
24  ’
30  Sum=0
31      for K=1 to 2*M−1:Y=K*H:W=K/(2*M)
32      Sum+=fnTrigam(W)*fnGemz(Y)*sin(N*Y):next
33  Fun=1-N/M-Sum/U
34  ’
40  print
41  print "Lattice point=";N−1,"Functions=";using(,6),Fun
42  end
43  ’
80  fnTrigam(W)
81  Tri=0
82      for J=0 to 6
83      Tri+=1/(J+W)ˆ2:next
84  V=1/(7+W)
85  Tri+=V+Vˆ2/2+Vˆ3/6−Vˆ5/30+Vˆ7/42
86  return(Tri)
87  ’
90  fnGemz(Y)
91  S=#i*Y
92  Z=exp(−S)
```

```
93  Rho=0.75:Bt=4*Rho/(1+Rho)^2
94  Gen=(1−sqrt(1−Bt*Z))/sqrt (Bt*Rho)
95  Gz=im((1−Z)*Gen)
96  return(Gz)
```

```
 1    'The Algorithm GAVER
 2    '
 3    'The Gaver-Stehfest method
 4    'for inverting Laplace transforms,
 5    'applied to the M/G/1 transform (1.1)
 6    '
10    NN=16
11    dim A (2*NN),B(2*NN),G(NN)
12    '
20    input ''TIME='';T:Alfa=log(2)/T
21    '
30    for K=1 to 2*NN
31    A(K)=K*Alfa*fnFs(K*Alfa):next
32    '
40    for J=1 to NN
41        for M=J to 2*NN-J
42        B(M)=(1+M/J)*A(M)-(M/J)*A(M+1)
43        next M
```

```
44              for I=J to 2*NN-J
45              A(I)=B(I):next I
46    G(J)=B(J):next J
47    '
50    'Fun=0
51       for N=1 to NN
52       Wt=(-1)^(NN-N)*N^NN/(!(N)*!(NN-N))
53       Fun+=Wt*G(N):next
54    '
60    print
61    print "TIME=";T,"FUNCTION=";using(2,7),Fun
62    end
63    '
90    fnFs(S)
91    Rho=0.75:Mean=1
92    Gs=1/sqrt(1+2*S)
93    Gse=(1−Gs)/(Mean*S)
94    F=(1−Gse)/(S*(1−Rho*Gse))
95    return(F)
```