MULTI-DIMENSIONAL TRANSFORM INVERSION WITH APPLICATIONS TO THE TRANSIENT M/G/1 QUEUE

by

Gagan L. Choudhury,[1] *David M. Lucantoni*[2] *and Ward Whitt*[3]

AT&T Bell Laboratories

June 15, 1993

Revision: November 19, 1993

[1] AT&T Bell Laboratories, Room 3K-603, Holmdel, NJ 07733-3030
gagan@buckaroo.att.com

[2] AT&T Bell Laboratories, Room 3K-601, Holmdel, NJ 07733-3030
dave@buckaroo.att.com

[3] AT&T Bell Laboratories, Room 2C-178, Murray Hill, NJ 07974-0636
wow@research.att.com

Abstract

We develop an algorithm for numerically inverting multi-dimensional transforms. Our algorithm applies to any number of continuous variables (Laplace transforms) and discrete variables (generating functions). We use the Fourier-series method; i.e., the inversion formula is the Fourier series of a periodic function constructed by aliasing. This amounts to an application of the Poisson summation formula. By appropriately exponentially damping the given function, we control the aliasing error. We choose the periods of the multi-dimensional periodic function so that each infinite series is a finite sum of nearly alternating infinite series; then we apply the Euler transformation to compute the infinite series from finitely many terms. The multi-dimensional inversion algorithm enables us, evidently for the first time, to quickly and accurately calculate probability distributions from several classical transforms in queueing theory. For example, we apply our algorithm to invert the two-dimensional transforms of the joint distribution of the duration of a busy period and the number served in that busy period, and the time-dependent of the transient queue-length and workload distributions, in the M/G/1 queue. In other related work, we have applied the inversion algorithms here to calculate time-dependent distributions in the transient BMAP/G/1 queue (with a batch Markovian arrival process) and the piecewise-stationary $M_t/G_t/1$ queue.

Keywords: numerical transform inversion, Laplace transforms, generating functions, multi-dimensional transforms, Fourier transforms, Fourier-series method, Poisson summation formula, M/G/1 queue, transient distributions.

1980 AMS subject classification: 65D30, 60K25.

1. Introduction

In this paper we present an algorithm for numerically inverting multi-dimensional transforms. We are motivated by the desire to compute probability distributions of interest in queues and related stochastic models, but of course there are many other applications. We even allow the inverse transform to be complex-valued. However, in our error analysis we exploit the fact that the modulus of our functions have known bounds, so that the algorithm is particularly appropriate for probability transforms (where the bound is 1). This algorithm is evidently the first multi-dimensional inversion algorithm in the queueing literature. However, we have learned that a different multi-dimensional inversion algorithm intended for queueing models has recently been developed in Russia by Frolov and Kitaev (1992). Their algorithm evidently is similar to a multi-dimensional version of the POST-WIDDER algorithm in Abate and Whitt (1992a). Of course, there is substantial literature on numerical transform inversion, as reviewed in Abate and Whitt (1992a). However, relatively little attention has been given to inversion of multi-dimensional transforms; for some instances, see Singhal, Vlach and Vlach (1975), Huntley and Zinober (1979) and Shephard (1991).

We consider both continuous variables (Laplace transforms) and discrete variables (generating functions). We thus consider three types of two-dimensional transforms: (i) continuous-continuous, (ii) continuous-discrete and (iii) discrete-discrete. We also show how the formulas can be generalized to more than two dimensions with any number of continuous and discrete variables.

The multi-dimensional inversions obviously allows us to compute multivariate probability distributions, as we illustrate here. However, the multi-dimensional inversions also allow us to calculate *time-dependent* probability distributions in queueing models that are not in steady state. As an example, in this paper we invert the classical double transform expressions for the transient workload and queue-length distributions in the M/G/1 queue; see Takács (1962). The special case of the M/M/1 transient queue length has been widely studied in the literature; e.g., see Abate and Whitt (1989). We show that our algorithm in this case is comparable in speed and accuracy to the numerical integration of the integral representation in Abate and Whitt (1989). Moreover, our algorithm applies equally well to the case of non-exponential service times, with no loss of speed or accuracy. In fact, we have applied the algorithm here to calculate time-dependent distributions in the transient BMAP/G/1 queue (with a batch Markovian arrival process) in Lucantoni, Choudhury and Whitt (1994) and the piecewise-stationary $M_t/G_t/1$ queue in Choudhury, Lucantoni and Whitt (1993). We plan to report on applications of the multi-dimensional inversion algorithm to other important queueing problems in the future.

Our algorithms here is a multivariate generalization of the algorithms EULER and LATTICE-POISSON in Abate and Whitt (1992a). We also introduce an enhancement of those algorithms to be able to simultaneously control the aliasing and roundoff errors. As in Abate and Whitt (1992a), we exploit the Fourier-series method. The general approach goes back at least to Fettis (1955); see Abate and Whitt (1992a) for a review. For the multi-dimensional transforms, this means that we apply the multivariate version of the Poisson summation formula, as given for the two-dimensional continuous-continuous case in (5.47) of Abate and Whitt (1992a); also see Good (1962). The approach is closely related to the fast Fourier transform (FFT). The idea is relatively simple: Just as in the one-dimensional case, in the two-dimensional continuous-continuous case we damp the given function by multiplying by a two-dimensional decaying exponential function and then approximate the damped function by a periodic function constructed by aliasing. We use the two exponential parameters to control the aliasing error in this approximation by the periodic function. The inversion formula is then the two-dimensional Fourier series of the periodic function. This yields what we want, because the transform values are the two-dimensional Fourier coefficients. Moreover, the two periods of the periodic function can be chosen so that the two-dimensional Fourier series is a series nested within a second series, each of them being nearly an alternating series. Hence, we can efficiently calculate each infinite series from finitely many terms by exploiting the Euler transformation (or summation). In practice, this usually means that it suffices to compute 100 or fewer terms of each infinite series to achieve a truncation error of the order 10^{-13} or less; see Abate and Whitt (1992a), Johnsonbaugh (1979) and Wimp (1981). When the inverse transform is real, as with probabilities, the overall computation can be reduced by a factor of two.

However, the above choice of the exponential parameters and the periods does not allow us to simultaneously control the aliasing error and the roundoff error. Therefore, we choose the periods such that every l_1^{th} term of the first series and every l_2^{th} term of the second series are nearly alternating. Therefore, the first infinite series may be considered as the sum of l_1 nearly alternating series and the second infinite series may be considered as the sum of l_2 nearly alternating series. Then each alternating series may be efficiently computed using the Euler transformation as mentioned above. The two exponential parameters of the two-dimensional decaying exponential functions along with l_1 and l_2 allow us to simultaneously control the aliasing and roundoff errors, thereby achieving an accurate two-dimensional algorithm.

If one or both the dimensions are discrete, then each such dimension corresponds to the replacement of a continuous function defined over the non-negative real line by a series defined over the non-negative integers. Ideas similar to the continuous case apply to the discrete case, with the decaying exponential function replaced by a decaying geometric series and the Fourier series replaced by a discrete Fourier series. An important difference in the discrete case is that the discrete fourier series, and hence also the corresponding inversion formula, have only finitely many terms. Therefore, we can compute all the terms and do not need to use the Euler transformation. However, if the number of terms in the finite series is very large (several hundreds or more), then we use the Euler transformation in this case as well.

The ideas above apply to arbitrary dimensions and any mixture of discrete and continuous variables. Indeed, one important contribution of this paper is the seamless combination of discrete and continuous variables.

Here is how the rest of this paper is organized. In Sections 2, 3 and 4, respectively, we develop the two-dimensional inversion formulas for the continuous-continuous, continuous-discrete and discrete-discrete cases. In Section 5 we show how the formulas can be generalized to more than two dimensions with any number of continuous and discrete variables. It is significant that the overall algorithm for *n* dimensions reduces to the iterative application of the one-dimensional algorithm *n* times, in any order.

In Section 6 we apply the inversion algorithm to specific examples associated with the M/G/1 queue. We illustrate each of the variants of the algorithm in Sections 2-4. We intend to indicate in a subsequent paper how to calculate moments and asymptotic parameters of time-dependent probability distributions, extending the algorithm in Choudhury and Lucantoni (1994).

2. Two-Dimensional Inversion with Continuous Variables

In this section we develop the variant of our algorithm to numerically invert a two-dimensional Laplace transform. Let $f(t_1, t_2)$ be a complex-valued function of nonnegative real variables t_1 and t_2, and let its two-dimensional *Laplace transform* be

$$\tilde{f}(s_1, s_2) = \int_0^\infty \int_0^\infty e^{-(s_1 t_1 + s_2 t_2)} f(t_1, t_2)\, dt_1\, dt_2 \ , \tag{2.1}$$

which we assume is well defined; e.g., see Ditkin and Prudnikov (1962) or Van der Pol and Bremmer (1955). In (2.1) s_1 and s_2 are complex variables with $Re(s_1) > 0$ and $\mathrm{Re}(s_2) > 0$. We will show how to calculate $f(t_1, t_2)$ using values of $\tilde{f}(s_1, s_2)$.

2.1 Developing the Algorithm

We start by considering Fourier transforms. Let $F(t_1, t_2)$ be a complex-valued function on R^2 with a well-defined *bivariate Fourier transform*

$$\phi(u_1, u_2) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{i(t_1 u_1 + t_2 u_2)} F(t_1, t_2)\, dt_1\, dt_2 \;, \tag{2.2}$$

(If F is a probability density function, then ϕ is its characteristic function; see pp. 521-525 of Feller (1971).) Under regularity conditions, F can be recovered by the *Fourier inversion formula*

$$F(t_1, t_2) = \frac{1}{4\pi^2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} e^{-i(t_1 u_1 + t_2 u_2)} \phi(u_1, u_2)\, du_1\, du_2 \;; \tag{2.3}$$

In the Fourier theory, F and ϕ constitute a *Fourier pair*, see Chapter 8 of Champeney (1987). It is significant that (2.2) and (2.3) hold in great generality provided the integrals are interpreted properly; in particular, F need not be bounded and continuous. The regularity conditions in the one-dimensional case are discussed in §5 of Abate and Whitt (1992a); we will not discuss the regularity conditions here.

We now exploit the two-dimensional *Poisson summation formula*

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} F\left[t_1 + \frac{2\pi j}{h_1} \;,\; t_2 \;+\; \frac{2\pi k}{h_2}\right] = \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} \frac{h_1 h_2}{4\pi^2} \phi(j h_1, k h_2)\, e^{-(j h_1 t_1 + k h_2 t_2)} \;. \tag{2.4}$$

The left side of (2.4) is constructed by *aliasing* to be a periodic function of t_1 and t_2 with periods h_1^{-1} and h_2^{-1}, respectively. (Aliasing means that the new function is constructed by adding translated versions of the original function.) Assuming that the series on the left in (2.4) converges and that this periodic function has a proper *Fourier series*, the Fourier series is given by the right side of (2.4). Hence, given that the aliased function on the left side is well defined, the validity of (2.4) depends on the classical theory of Fourier series; see §5 of Abate and Whitt (1992a) and Tolstov (1976). For our inversion problem, the key point is that the Fourier transform values $\phi(jh_1, kh_2)$ from (2.2) appear as the Fourier coefficients in (2.4); see (5.47) of Abate and Whitt (1992a) and p. 163 of Champeney (1987). Note that the right side of (2.4) can be regarded as a *trapezoidal rule* form of numerical integration applied to the inversion integral (2.3).

In order to control the aliasing error, we do *exponential damping*; i.e., if f is our original function of interest in (2.1), then we replace $F(t_1, t_2)$ above by the function $f(t_1, t_2)\, e^{-(a_1 t_1 + a_2 t_2)}$ when $t_1 \geq 0,\ t_2 \geq 0$ and 0 elsewhere. Then $\phi(u_1, u_2) = \tilde{f}(a_1 - iu_1,\ a_2 - iu_2)$ for $\tilde{f}$ in (2.1), and the right side of (2.4) can be expressed in terms of the Laplace transform values. If, in addition, we let $h_1 = \pi/(t_1 l_1)$ and $h_2 = \pi/(t_2 l_2)$, where $l_1,\ l_2 \geq 1$, then (2.4) becomes

$$\sum_{j=0}^{\infty} \sum_{k=0}^{\infty} e^{-[a_1(1+2jl_1)t_1 + a_2(1+2kl_2)t_2]} f((1+2jl_1)t_1,\ (1+2kl_2)t_2)$$

$$= \frac{1}{4 l_1 t_1 l_2 t_2} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} e^{-i\left[\frac{j\pi}{l_1} + \frac{k\pi}{l_2}\right]} \tilde{f}\left[a_1 - \frac{ij\pi}{l_1 t_1}\ ,\ a_2 - \frac{ik\pi}{l_2 t_2}\right] . \qquad (2.5)$$

If, furthermore, we let $a_1 = A_1/(2t_1 l_1)$ and $a_2 = A_2/(2t_2 l_2)$, then we get

$$\sum_{j=0}^{\infty} \sum_{k=0}^{\infty} e^{-(A_1 j + A_2 k)} f((1+2jl_1)t_1, (1+2kl_2)t_2)$$

$$= \frac{\exp\left[\frac{A_1}{2l_1} + \frac{A_2}{2l_2}\right]}{4 l_1 t_1 l_2 t_2} \sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} e^{-i\left[\frac{j\pi}{l_1} + \frac{k\pi}{l_2}\right]} \tilde{f}\left[\frac{A_1}{2l_1 t_1} - \frac{ij\pi}{l_1 t_1}\ ,\ \frac{A_2}{2l_2 t_2} - \frac{ik\pi}{l_2 t_2}\right] . \quad (2.6)$$

Note that we can rewrite (2.6) as $f(t_1, t_2) = \bar{f}(t_1, t_2) - \bar{e}$, where the value to be calculated is

$$\bar{f}(t_1, t_2) = \frac{\exp(A_1/2l_1)}{2 l_1 t_1} \sum_{j=-\infty}^{\infty} e^{-ij\pi/l_1} \left\{ \frac{\exp(A_2/2l_2)}{2 l_2 t_2} \sum_{k=-\infty}^{\infty} e^{-ik\pi/l_2} \right.$$

$$\left. \times \left[\tilde{f}\left[\frac{A_1}{2l_1 t_1} - \frac{ij\pi}{l_1 t_1}\ ,\ \frac{A_2}{2l_2 t_2} - \frac{ik\pi}{l_2 t_2}\right]\right]\right\} \qquad (2.7)$$

and the error is

$$\bar{e} \equiv \bar{e}(t_1,t_2,A_1,A_2,l_1,l_2) = \sum_{\substack{j=0 \\ not\ j=k=0}}^{\infty} \sum_{k=0}^{\infty} e^{-(A_1 j + A_2 k)} f((1 + 2jl_1)t_1,\ (1+2kl_2)t_2)\ . \quad (2.8)$$

From (2.7) we see that the two-dimensional formula is the iterated one-dimensional formulas. In particular, if $l_1 = l_2 = 1$, then the expression within the braces in (2.7) can be regarded as the one-dimensional EULER algorithm in (5.26) of Abate and Whitt (1992a) with the one-dimensional transform replaced by the two-dimensional transform $\tilde{f}$. Moreover, the entire expression (2.7) can be regarded as the one-dimensional EULER algorithm with the one-dimensional transform replaced by the quantity in braces.

We regard $\bar{e}$ in (2.8) as the error term, which will not be explicitly computed. if $|f(t_1,t_2)| \leq C$ for some constant C and all t_1,t_2 ($C = 1$ if $f(t_1,t_2)$ is a probability), then the error can be bounded as follows:

$$|\bar{e}| \leq \frac{C(e^{-A_1} + e^{-A_2} - e^{-(A_1+A_2)})}{(1-e^{-A_1})(1-e^{-A_2})} \approx C(e^{-A_1} + e^{-A_2})\ . \quad (2.9)$$

In order to be able to exploit the Euler summation technique for nearly alternating series, we rewrite (2.7) as

$$\bar{f}(t_1,t_2) = \frac{\exp(A_1/2l_1)}{2l_1 t_1} \sum_{j_1=1}^{l_1} \sum_{j=-\infty}^{\infty} (-1)^j e^{-\frac{ij_1\pi}{l_1}} \left\{ \frac{\exp(A_2/2l_2)}{2l_2 t_2} \sum_{k_1=1}^{l_2} \sum_{k=-\infty}^{\infty} (-1)^k \right.$$

$$\left. \times \frac{e^{-ik_1\pi}}{l_2} \tilde{f}\left[\frac{A_1}{2l_1 t_1} - \frac{ij_1\pi}{l_1 t_1} - \frac{ij\pi}{t_1}\ ,\ \frac{A_2}{2l_2 t_2} - \frac{ik_1\pi}{l_2 t_2} - \frac{ik\pi}{t_2} \right] \right\}\ . \quad (2.10)$$

So far, we allowed f to be complex-valued. (Hence, $|f|$ and $|\bar{e}|$ should be interpreted as the modulus.) However, if f is real-valued, then we can reduce the computations by a factor of 2 by noting that $\tilde{f}(\bar{s}_1, \bar{s}_2) = \overline{\tilde{f}(s_1, s_2)}$, where $\bar{s}$ is the complex conjugate of s. Then (2.10) can be expressed as

$$\bar{f}(t_1, t_2) = \frac{\exp\left[\frac{A_1}{2l_1} + \frac{A_2}{2l_2}\right]}{4t_1 l_1 t_2 l_2} \left\{ \tilde{f}\left[\frac{A_1}{2t_1 l_1}, \frac{A_2}{2t_2 l_2}\right] \right.$$

$$+ 2 \sum_{k_1=1}^{l_2} \sum_{k=0}^{\infty} (-1)^k \operatorname{Re}\left[e^{-ik_1\pi/l_2} \tilde{f}\left[\frac{A_1}{2t_1 l_1}, \frac{A_2}{2t_2 l_2} - \frac{ik_1\pi}{t_2 l_2} - \frac{ik\pi}{t_2}\right]\right]$$

$$+ 2 \sum_{j_1=1}^{l_1} \sum_{j=0}^{\infty} (-1)^j \operatorname{Re}\left[\sum_{k_1=1}^{l_2} \sum_{k=0}^{\infty} (-1)^k e^{-\left[\frac{ij_1\pi}{l_1} + \frac{ik_1\pi}{l_2}\right]} \right.$$

$$\left. \times \tilde{f}\left[\frac{A_1}{2t_1 l_1} - \frac{ij_1\pi}{t_1 l_1} - \frac{ij\pi}{t_1}, \frac{A_2}{2t_2 l_2} - \frac{ik_1\pi}{t_2 l_2} - \frac{ik\pi}{t_2}\right]\right]$$

$$+ 2 \sum_{j_1=1}^{l_1} \sum_{j=0}^{\infty} (-1)^j \operatorname{Re}\left[e^{-ij_1\pi/l_1} \tilde{f}\left[\frac{A_1}{2t_1 l_1} - \frac{ij_1\pi}{t_1 l_1} - \frac{ij\pi}{t_1}, \frac{A_2}{2t_2 l_2}\right]\right.$$

$$+ \sum_{k_1=1}^{l_2} \sum_{k=0}^{\infty} (-1)^k e^{-\left[\frac{ij_1\pi}{l_1} - \frac{ik_1\pi}{l_2}\right]}$$

$$\left.\left. \times \tilde{f}\left[\frac{A_1}{2t_1 l_1} - \frac{ij_1\pi}{t_1 l_1} - \frac{ij\pi}{t_1}, \frac{A_2}{2t_2 l_2} + \frac{ik_1\pi}{t_2 l_2} + \frac{ik\pi}{t_2}\right]\right]\right\}. \tag{2.11}$$

Note that (2.11) contain infinite sums of the form $S = \sum_{k=0}^{\infty} (-1)^k a_k$ where a_k is real or complex. Also, equation (2.10) contains infinite sums of the form $\sum_{k=-\infty}^{\infty} (-1)^k a_k$, which can be written as the sum of two separate sums over the nonnegative integers. Section 6 of Abate and

Whitt (1992a) explains the Euler transformation for computing infinite sums of the above form when a_k is real; also see Davis and Rabinowitz (1984), Johnsonbaugh (1979) and Wimp (1981). Specifically, the *Euler sum* with parameters *n* and *m* is given by,

$$E(m,n) = S_n + (-1)^{n+1} \sum_{k=0}^{m-1} (-1)^k 2^{-(k+1)} \Delta^k a_{n+1} = \sum_{k=0}^{m} \binom{m}{k} 2^{-m} S_{n+k} , \qquad (2.12)$$

where,

$$S_j = \sum_{k=0}^{j} (-1)^k a_k , \qquad (2.13)$$

$\Delta a_j = a_{j+1} - a_j$ and Δ^k is obtained by *k*-fold application of the forward-difference operator Δ.

Unfortunately, we do not have general error bounds associated with the computation $E(m,n)$. As reviewed in Abate and Whitt (1992a) and Johnsonbaugh (1979), it is known that if

$$(-1)^m \Delta^m a_{n+k} \text{ is decreasing in } k \text{ for } k \geq 1 , \qquad (2.14)$$

then

$$|E(m,n) - S| \leq \frac{\Delta^m a_{n+1}}{2^m} = |E(m,n) - E(m-1,n)| , \qquad (2.15)$$

and an upper bound on the error in Euler sum can be obtained by computing $E(m,n)$ and $E(m-1,n)$. In case a_k is complex and both its real and imaginary parts satisfy condition (2.14), then (2.15) also gives a bound for complex Euler sums. However, in general it is difficult to verify condition (2.14). Our numerical experience shows, though, that unless we compute the inverse transform near a discontinuity, usually $E(m,n)$ computes *S* with an error of the order of 10^{-13} or less with the choice $n = 38$ and $m = 11$, i.e., requiring the computation of only 50 terms. In contrast, a straightforward computation of the infinite series by truncation after *K* terms would often require *K* to be 10,000 or more.

As in Abate and Whitt (1992a), we use $|E(m,n) - E(m-1,n)|$ as an *estimate* of the error produced by applying Euler summation.

2.2 Error Control In the Inversion Algorithm

There are three sources of error in the inversion algorithms (2.10) and (2.11). We now explain them and show how to control them. The error term $\bar{e}$ in (2.8) and (2.9) can be interpreted either as an *aliasing error*, since the periodic function on the left side of (2.4) is constructed by aliasing, or as a *discretization error*, since the right side of (2.4) can be interpreted as a trapezoidal rule form of numerical integration. For the rest of the paper we refer to $\bar{e}$ only as aliasing error. This error may be reduced by increasing the parameters A_1 and A_2 in (2.9). For example, if $C = 1$ (as in probability applications) then we can limit $|\bar{e}|$ to 10^{-8} by choosing $A_1 = A_2 = 19.1$ and limit it to 10^{-12} by choosing $A_1 = A_2 = 28.3$.

The second source of error comes from approximating each infinite series in (2.10) and (2.11) by a finite number of terms. We call this the *truncation error*, even though we do not do straightforward truncation. As explained earlier, unless we attempt to compute the inverse transform near discontinuities, we can usually reduce the truncation error to 10^{-13} or lower by using the Euler summation technique with about 50 terms. As indicated above, we estimate the truncation error using $|E(m,n) - E(m-1,n)|$.

The third source of error is *roundoff error*, which is primarily due to multiplying large numbers by small ones. Specifically, the quantity $\exp\left[\frac{A_1}{2l_1} + \frac{A_2}{2l_2}\right] / (4 l_1 t_1 l_2 t_2)$ appearing in both (2.10) and (2.11) can be large. However, there are four parameters to control it: A_1, A_2, l_1, and l_2. Since we have already used A_1 and A_2 to control the aliasing error, we use l_1 and l_2 to control the roundoff error. (The one-dimensional Euler algorithm in Abate and Whitt (1992a) did not use any parameter like l_1 and l_2 and hence could not control the roundoff and aliasing errors simultaneously.) Table 1 shows how the quantity $\exp\left[\frac{A_1}{2l_1} + \frac{A_2}{2l_2}\right] / (4 l_1 t_1 l_2 t_2)$ decreases (thereby decreasing the roundoff error) with increasing l_1 and l_2 (assuming $t_1 = t_2 = 1$). We consider two cases with aliasing error bounds of 10^{-8} and 10^{-12}, respectively. This bound fixes A_1 and A_2 (assuming $A_1 = A_2$) and we change l_1, l_2 to control the roundoff error. Note that the cost of reducing the roundoff error is the increase in computation time which is proportional to the product of l_1 and l_2. For any choice of l_1 and l_2, we choose A_1 and A_2 such that the aliasing and roundoff errors are about the same order of magnitude. Our numerical experience indicates that with $l_1 = l_2 = 1$ we can usually achieve an overall accuracy of 5 or 6 digits, and with $l_1 = l_2 = 2$ we can usually achieve an overall accuracy of 10 or more digits. This is based on a double-precision arithmetic (i.e., about 16-digit precision). For two-dimensional inversion, usually $l_1 = l_2 = 2$ is adequate. However, in order to achieve high accuracy with higher dimensional inversions (to be described in Section 5) we may need bigger l_1 and l_2. In Choudhury et al. (1993) we solved a problem with two and one dimensional inversions, but the inversions were nested, thereby effectively amounting to an n-dimensional inversion where n could be as large as 22. We could accurately solve that problem by choosing each l_i to be 7.

3. Two-Dimensional Inversion with Discrete Variables

Let $p_{n_1 n_2}$ be a double sequence of complex numbers defined on the pairs (n_1, n_2) of nonnegative integers, and let $G(z_1, z_2)$ be its two-dimensional *generating function*, which we assume is well defined; i.e., paralleling (2.1), we have

$$G(z_1, z_2) = \sum_{n_1=0}^{\infty} \sum_{n_2=0}^{\infty} p_{n_1 n_2} z_1^{n_1} z_2^{n_2} \ . \tag{3.1}$$

We will show how to compute $p_{n_1 n_2}$ using the values of $G(z_1, z_2)$.

As in §2.1, we start by considering general Fourier transforms. Let a_{n_1, n_2} be a sequence of complex numbers on the pairs (n_1, n_2) of integers, and let $\phi(u_1, u_2)$ be its *discrete Fourier transform*, where

$$\phi(u_1,u_2) \;=\; \sum_{n_1=-\infty}^{\infty} \sum_{n_2=-\infty}^{\infty} a_{n_1 n_2} e^{i(u_1 n_1 + u_2 n_2)} \;. \tag{3.2}$$

Paralleling (2.4), we obtain the *discrete Poisson summation formula*

$$\sum_{j=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} a_{n_1+jm_1, n_2+km_2} = \frac{1}{m_1 m_2} \sum_{j=\frac{-m_1}{2}}^{\frac{m_1}{2}-1} \sum_{k=\frac{-m_2}{2}}^{\frac{m_2}{2}-1} \phi\left[\frac{2\pi j}{m_1}, \frac{2\pi k}{m_2}\right] e^{-i\left[\frac{2\pi j n_1}{m_1} + \frac{2\pi k n_2}{m_2}\right]} \tag{3.3}$$

The left side of (3.3) is constructed by aliasing to be a bivariate periodic sequence with periods m_1 and m_2, respectively. We assume that m_1 and m_2 are even positive integers. The right side of (3.3) is the two-dimensional discrete Fourier series of the periodic sequence on the left. In order to control the aliasing error, we assume that $a_{n_1 n_2}$ is defined in terms of our original sequence $p_{n_1 n_2}$ by

$$a_{n_1 n_2} \;=\; \begin{cases} p_{n_1 n_2} \;\; r_1^{n_1} r_2^{n_2} \;\; \text{for } n_1 \geq 0 \;,\; n_2 \geq 0 \\ \\ 0 \qquad \text{otherwise} \;, \end{cases} \tag{3.4}$$

where r_i is a real number with $0 < r_i < 1$ for $i = 1,2$. The term $r_1^{n_1} r_2^{n_2}$ in (3.4) constitutes a geometric damping, paralleling the exponential damping in §2. With (3.4), the generating function G in (3.1) is related to the transform ϕ in (3.2) by

$$\phi(u_1,u_2) \;=\; G(r_1 e^{iu_1} \;,\; r_2 e^{iu_2}) \;.$$

From (3.3), after some manipulations, we get $p_{n_1 n_2} \;=\; \bar{p}_{n_1 n_2} - \bar{e}$, where

$$\bar{p}_{n_1,n_2} \;=\; \frac{1}{m_1 r_1^{n_1}} \sum_{j=\frac{-m_1}{2}}^{\frac{m_1}{2}-1} e^{-\frac{2\pi i j n_1}{m_1}} \left\{ \frac{1}{m_2 r_2^{n_2}} \sum_{k=-\frac{m_2}{2}}^{\frac{m_2}{2}-1} e^{\frac{-2\pi i k n_2}{m_2}} G(r_1 e^{\frac{2\pi i j}{m_1}} \;,\; r_2 e^{\frac{2\pi i k}{m_2}}) \right\} \tag{3.5}$$

and

$$e \equiv e(m_1, m_2, r_1, r_2) = \sum_{\substack{j=0 \\ \text{not } j=k=0}}^{\infty} \sum_{k=0}^{\infty} p_{n_1 + jm_1, n_2 + km_2} r_1^{jm_1} r_2^{km_2} . \tag{3.6}$$

If $|p_{n_1,n_2}| \le C_2$ then

$$|\bar{e}| \le \frac{C(r_1^{m_1} + r_2^{m_2} - r_1^{m_1} r_2^{m_2})}{(1 - r_1^{m_1})(1 - r_2^{m_2})} \approx C(r_1^{m_1} + r_2^{m_2}) . \tag{3.7}$$

Assuming that $m_1 = 2l_1 n_1$ and $m_2 = 2l_2 n_2$, we can rewrite (3.5) as

$$\begin{aligned} \bar{p}_{n_1,n_2} = \frac{1}{2l_1 n_1 r_1^{n_1}} \sum_{j_1=0}^{l_1-1} \sum_{j=-n_1}^{n_i-1} (-1)^j e^{-\frac{\pi i j_1}{l_1}} \Bigg\{ \frac{1}{2l_2 n_2 r_2^{n_2}} \sum_{k_1=0}^{l_2-1} \sum_{k=-n_2}^{n_2-1} (-1)^k \\ \times e^{-\frac{\pi i k_1}{l_2}} G(r_1 e^{\frac{\pi i (j_1 + l_1 j)}{l_1 n_1}}, r_2 e^{\frac{\pi i (k_1 + l_2 k)}{l_2 n_2}}) \Bigg\} \end{aligned} \tag{3.8}$$

and the upper bound in (3.7) as $C(r_1^{2l_1 n_1} + r_2^{2l_2 n_2})$.

If p_{n_1,n_2} is real-valued, then it is possible to reduce the computations by a factor of 2 by using the fact that $\overline{G(r_1 e^{iu_1}, r_2 e^{iu_2})} = G(r_1 e^{-iu_1}, r_2 e^{-u_2})$, but we do not show that expression.

Note that (3.8) can be considered as an iterative application of two one-dimensional algorithms. When $l_1 = l_2 = 1$, Formula (3.8) is the two-dimensional generalization of the algorithm LATTICE-POISSON in Abate and Whitt (1992a,b). We use l_1 and l_2 to be able to simultaneously control the aliasing and roundoff errors.

Paralleling §2, the aliasing error is controlled by reducing $C(r_1^{2l_1 n_1} + r_2^{2l_2 n_2})$, while the roundoff error is controlled by reducing the factor $1/(4l_1 l_2 n_1 n_2 r_1^{n_1} r_2^{n_2})$, using the four parameters l_1, l_2, r_1 and r_2. Since (3.8) has only finite sums, there is no truncation error. However, if n_1 and n_2 are very large, then we can also use the Euler summation. The sums in (3.8) are expressed as nearly alternating series with this in mind.

4. One Discrete and One Continuous Variable

Now let the function of interest be $f(t,n)$, where t is a nonnegative continuous variable and n is a nonnegative integer. We wish to calculate $f(t,n)$ by numerically inverting the two-dimensional transform

$$\tilde{f}(s,z) = \int_0^\infty \sum_{n=0}^\infty f(t,n) e^{-st} z^n dt \ . \tag{4.1}$$

As before, we work with Fourier transforms. For this purpose, let $F(t,n)$ be defined for real t and integer n and let $\phi(u_1, u_2)$ its fourier transform, i.e.,

$$\phi(u_1,u_2) = \int_{-\infty}^\infty \sum_{n=-\infty}^\infty F(t,n) e^{i(u_1 t + u_2 n)} dt \ . \tag{4.2}$$

The *bivariate mixed Poisson summation formula* is

$$\sum_{j=-\infty}^\infty \sum_{k=-\infty}^\infty F\left(t + \frac{2\pi j}{h}, n + km\right) = \frac{h}{2\pi m} \sum_{j=-\infty}^\infty \sum_{k=-\frac{m}{2}}^{\frac{m}{2}-1} \phi\left(jh, \frac{2\pi k}{m}\right) e^{-i\left[jht + \frac{2\pi kn}{m}\right]} \ . \tag{4.3}$$

The left side of (4.3) is constructed to be periodic by aliasing. The right side is a Fourier series with respect to variable t and a discrete Fourier series with respect to n. In order to control the aliasing error we do exponential/geometric damping as follows:

$$F(t,n) = \begin{cases} f(t,n) e^{-at} r^n & \text{for } t \geq 0 \ , \ n \geq 0 \\ 0 & \text{otherwise} \ , \end{cases} \tag{4.4}$$

where, $a > 0$ and $0 < r < 1$. Then $\phi(u_1,u_2) = \tilde{f}(a - iu_1, re^{iu_2})$. Letting $h = \pi/(tl_1)$, $m = 2l_2 n$ and $a = A/(2tl_1)$, after some manipulations on (4.3), we get $f(t,n) = \bar{f}(t,n) - \bar{e}$, where

$$\bar{f}(t,n) \;=\; \frac{\exp\left[\dfrac{A}{2l_1}\right]}{2l_1 t} \sum_{j_1=1}^{l_1} \sum_{j=-\infty}^{\infty} (-1)^j e^{-\frac{ij_1\pi}{l_1}}$$

$$\times \left\{ \frac{1}{2l_2 n r_2^n} \sum_{k_1=0}^{l_2-1} \sum_{k=-n}^{n-1} (-1)^k e^{-\frac{ik_1\pi}{l_2}} \tilde{f}\left[\frac{A}{2l_1 t} - \frac{ij_1\pi}{l_1 t} - \frac{ij\pi}{t} \;,\; re^{\frac{\pi i(k_1+l_2 k)}{l_2 n}} \right] \right\} \tag{4.5}$$

and

$$\bar{e} \;=\; \sum_{\substack{j=0 \\ \text{not } j=k=0}}^{\infty} \sum_{k=0}^{\infty} e^{-Aj} r^{2kl_2 n} f((1+2jl_1)t \;,\; (1+2kl_2)n) \;. \tag{4.6}$$

Now the aliasing error can be bounded by

$$|\bar{e}| \;\le\; \frac{C(e^{-A} + r^{2l_2 n} - e^{-A} r^{2l_2 n})}{(1-e^{-A})(1-r^{2l_2 n})} \;\simeq\; C(e^{-A} + r^{2l_2 n}) \;, \tag{4.7}$$

assuming that $|f| \le C$. The computations in (4.5) can be further reduced by a factor of 2 if $f(t,n)$ is real, but we do not show the resulting expression.

Both the aliasing and roundoff errors may be controlled by the parameters A, r, l_1 and l_2. The infinite sum may be efficiently computed by the Euler summation technique. If n is very large, then the Euler summation technique may be used on the finite sum as well.

5. Arbitrary Number of Dimensions

The formulas in Sections 2-4 above can easily be generalized to an arbitrary number of dimensions. Let $f(\mathbf{t})$ be a complex-valued function of a vector $\mathbf{t} \equiv (t_1, \ldots, t_l)$ of l nonnegative real variables. We allow the variables to be either continuous or discrete (integer). Let T_k be a variable indicating the *type of variable k*; i.e., $T_k = 1$ if t_k is continuous and $T_k = 2$ if t_k is discrete. For $1 \le k \le l$, let I_k be the appropriate integral or sum operator for the variable t_k; i.e., let

$$I_k = \begin{cases} \int_0^\infty dt_k & \text{if } T_k = 1 \\ \sum_{t_k=0}^{\infty} & \text{if } T_k = 2 \ . \end{cases} \tag{5.1}$$

Let $\mathbf{s} \equiv (s_1, \ldots, s_l)$ be the vector of l complex transform variables. For $1 \leq k \leq l$, let

$$\alpha_k(s_k, t_k) = \begin{cases} e^{-s_k t_k} & \text{if } T_k = 1 \\ s_k^{t_k} & \text{if } T_k = 2 \ . \end{cases} \tag{5.2}$$

Then the multi-dimensional transform of f can be expressed as

$$\tilde{f}(\mathbf{s}) = \Big(\prod_{k=1}^{l} I_k\Big) f(\mathbf{t}) \prod_{k=1}^{l} \alpha_k(s_k, t_k) \ . \tag{5.3}$$

The multi-dimensional inversion formula can then be defined recursively. For this purpose, let A_k and r_k be positive constants, l_k a positive integer and $|r_k| < 1$. For $1 \leq k \leq l$, let $\hat{j}_k$ be the k-vector $(j_1, \ldots, j_k)$ associated with the l-vector $\mathbf{j} \equiv (j_1, \ldots, j_l)$. Similarly, for $1 \leq k \leq l$, let $\hat{p}_k$ be the k-vector $(p_1, \ldots, p_k)$ associated with the l-vector $\mathbf{p} = (p_1, \ldots, p_l)$.

Then the inversion formula is $f(\mathbf{t}) = \bar{f}(\mathbf{t}) - \bar{e}$, where $\bar{f}(t) \equiv F_{0,\hat{j}_0,\hat{p}_0}$ and for $1 \leq k \leq l$,

$$F_{k-1,\hat{j}_{k-1},\hat{p}_{k-1}} = \begin{cases} \dfrac{e^{A_k/2l_k}}{2t_k l_k} \displaystyle\sum_{p_k=1}^{l_k} \sum_{j_k=-\infty}^{\infty} (-1)^{j_k} e^{\frac{-ip_k\pi}{l_k}} F_{k,\hat{j}_k,\hat{p}_k} & \text{if } T_k = 1 \\ \dfrac{1}{2l_k t_k r_k^{t_k}} \displaystyle\sum_{p_k=0}^{l_k-1} \sum_{j_k=-t_k}^{t_k-1} (-1)^{j_k} e^{-\frac{ip_k\pi}{l_k}} F_{k,\hat{j}_k,\hat{p}_k} & \text{if } T_k = 2 \ , \end{cases} \tag{5.4}$$

where

$$F_{l,\hat{j}_l,\hat{p}_l} = \tilde{f}(\boldsymbol{\xi}) \ , \tag{5.5}$$

with $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_l)$ and

$$\xi_k = \begin{cases} \dfrac{A_k}{2t_k l_k} - \dfrac{ip_k\pi}{t_k l_k} - \dfrac{ij_k\pi}{t_k} & \text{if } T_k = 1 \\ r_k e^{\frac{\pi i(p_k + l_k j_k)}{l_k t_k}} & \text{if } T_k = 2 \; . \end{cases} \tag{5.6}$$

The error term $\bar{e}$ is then given by (in the notation of (5.3))

$$\bar{e} \equiv \sum_{\substack{j_1=0 \\ \text{not } j_1=\ldots=j_l=0}}^{\infty} \cdots \sum_{j_l=0}^{\infty} f(\boldsymbol{\tau})\Big(\prod_{k=1}^{l} \beta_k\Big) \; , \tag{5.7}$$

where $\boldsymbol{\tau} = (\tau_1, \ldots, \tau_l)$,

$$\tau_k = t_k(1 + 2j_k l_k) \tag{5.8}$$

and

$$\beta_k = \begin{cases} e^{-j_k A_k} & \text{if } T_k = 1 \\ r_k^{2l_k t_k j_k} & \text{if } T_k = 2 \; . \end{cases} \tag{5.9}$$

If $|f(\mathbf{t})| \le C$ for all allowed values of $\boldsymbol{t}$, then

$$|\bar{e}| \le \hat{e} \approx C \sum_{k=1}^{l} \gamma_k \; , \tag{5.10}$$

where

$$\gamma_k = \begin{cases} e^{-A_k} & \text{if } T_k = 1 \\ r_k^{2l_k t_k} & \text{if } T_k = 2 \; . \end{cases} \tag{5.11}$$

Note that the continuous and discrete variables in the formulas here can be ordered in an arbitrary way. Also note that the results of Sections 2-4 are all special cases of the formulas in this section. As before, if f is real-valued, then it is possible to reduce the computations somewhat, but the formulas get complicated.

6. Numerical Examples

The main motivation for our work has been the desire to compute probability distributions of interest in queueing models. In this section we provide a few examples associated with the M/G/1 queue. For the most part, the transforms can all be found in Takács (1962). Some additional details can be found in Lucantoni et al. (1994).

6.1 The Busy Period: Duration and Number Served

We start with the joint distribution of the number served, N, and the duration, X, of a busy period in the M/G/1 queue. Let $G_1(n) = P(N = n)$, $G_2(x) = P(X \leq x)$ and $G(n,x) = P(N = n,\ X \leq 2)$. We define the one-dimensional and two-dimensional transforms

$$\overline{G}_1(z) = \sum_{n=0}^{\infty} z^n G_1(n) \ , \tag{6.1}$$

$$\hat{G}_2(s) = \int_0^{\infty} e^{-sx} dG_2(x) \ , \tag{6.2}$$

$$\tilde{G}(z,s) = \sum_{n=0}^{\infty} \int_0^{\infty} e^{-sx} z^n d_x G(n,x) \ . \tag{6.3}$$

Note that $\hat{G}(s) = \tilde{G}(1,s)$ and $\overline{G}(z) = \tilde{G}(z,0)$. Numerically, it is easier to work with the Laplace transforms of the complimentary cumulative distribution functions rather than the cumulative distribution functions (CDFs) themselves (because there is less aliasing error). Therefore, we invert the transforms $\tilde{G}^c(z,s)$ and $\hat{G}^c(s)$, where

$$\tilde{G}^c(z,s) = \sum_{n=0}^{\infty} \int_0^{\infty} e^{-sx} z^n G^c(n,x)\, dx \ , \tag{6.4}$$

$$\hat{G}^c(s) = \int_0^{\infty} e^{-sx} G_2^c(x)\, dx \ , \tag{6.5}$$

$G^c(n,x) = P(N = n,\ X > x)$ and $G_2^c(x) = P(X > x)$. It can be shown that

$$\tilde{G}^c(z,s) = \frac{1}{s}(\overline{G}(z) - \tilde{G}(z,s)) \tag{6.6}$$

$$\hat{G}^c(s) = \frac{1}{s}(1 - \hat{G}(s)) \ . \tag{6.7}$$

It is well known that $\tilde{G}(z,s)$, $\hat{G}(s)$ and $\overline{G}(z)$ satisfy the functional equations

$$\tilde{G}(z,s) = z\hat{h}(s + \lambda - \lambda\tilde{G}(z,s)) \tag{6.8}$$

$$\hat{G}(s) = \hat{h}(s + \lambda - \lambda\hat{G}(s)) \tag{6.9}$$

$$\overline{G}(z) = z\hat{h}(\lambda - \lambda\overline{G}(z)) \ , \tag{6.10}$$

where $\hat{h}(s)$ represents the Laplace-Stieltjes transform of service-time CDF; see Takács (1962). We compute the transforms iteratively. In Choudhury, Lucantoni and Whitt (1994) we prove that all the iterations converge (even when server utilization is bigger than 1) if we start them at 0. We invert the one-dimensional transforms using the algorithms in Abate and Whitt (1992a) and the two-dimensional transform using the algorithm in Section 4.

In Figure 1 we plot, in log scale, the conditional busy-period distribution $P(X > x | N = n) = G^c(n,x)/G_1(n)$ for $n = 1$, $n = 5$ and $n = 25$ when the arrival rate is 0.8 and the service-time distribution is gamma with mean 1 and shape parameter 1/4. Then the squared coefficient of variation (SCV, variance divided by the square of the mean) is 4. We also show the unconditional distribution $P(X > x)$. Note that the conditional and the unconditional busy-period distributions are quite different.

Also note that the conditional distributions are not straightforward to find by alternate means. In particular, the conditional busy-period distribution is not the n-fold convolution of the service-time distribution. However, in the special case of deterministic service times, the conditional busy-period distribution is just a point mass at n times the constant service time. This case is difficult to invert numerically since the inverse transform is discontinuous. However, we have considered the E_k (Erlang of order k) service-time distribution with k up to a few hundreds and observed that as k increases, the conditional busy-period distribution approaches that of the point mass mentioned above. This provides a check on the algorithm. We have also calculated the distribution of number served conditioned on the length of the busy period, but we do not show that here.

6.2 The Transient Queue-Length Distribution

Next we consider the transient queue-length distribution in an M/G/1 queue. Let, $Q(t)$ represent the queue length at time t (including the one in service, if any). Let there be a departure at time $t = 0$ and at that instant let there be i_0 customers in the system. Let, $Y_{i_0}(n,t) = P(Q(t) = n \mid Q(0) = i_0)$. Consider the two-dimensional transform

$$\tilde{Y}_{i_0}(z,s) = \sum_{n=0}^{\infty} \int_{t=0}^{\infty} e^{-st} z^n Y_{i_0}(n,t)\, dt \; . \tag{6.11}$$

It can be shown that

$$\tilde{Y}_{i_0}(z,s) = \frac{z^{i_0+1}(1-\hat{h}(s+\lambda-\lambda z))}{(s+\lambda-\lambda z)(z-\hat{h}(s+\lambda-\lambda z))} + \frac{(z-1)\hat{p}_{i_0 0}(s)\hat{h}(s+\lambda-\lambda z)}{z-\hat{h}(s+\lambda-\lambda z)} \; , \tag{6.12}$$

where

$$\hat{p}_{i_0 0}(s) = \frac{\{\hat{G}(s)\}^{i_0}}{s+\lambda-\lambda\hat{G}(s) \; .} \tag{6.13}$$

and $\hat{G}_2(s)$ is defined in (6.2) and obtained iteratively using (6.9); see Takács (1962) and Lucantoni et al. (1994).

Using the results in Section 4, we invert the transform in (6.12) and get the transient queue-length distribution. In Figure 2 we plot in log scale this distribution at $t = 5$ with $i_0 = 10$ for three different service-time distributions, each with mean 1: M (exponential), $E_4 = \Gamma_4$ (Erlang or gamma with SCV = 1/4), and $\Gamma_{1/4}$ (gamma with SCV = 4). We note that greater service-time variability causes a greater variability in the queue-length distribution as well.

In Figure 3 we concentrate on the gamma service-time distribution and show the transient distribution at $t = 1$, $t = 5$ and $t = 100$. The steady-state distribution is also shown. Note that the transient behavior is quite different from the steady-state behavior. Also note that the transient tail decays faster than the steady-state tail (the latter is known to be geometric in this case). It is interesting to note that at $t = 100$ the transient and steady-state distributions are very close for small *n*, but at large *n* the transient tail decays much faster than the steady-state tail.

The special case of M/M/1 transient queue length has been studied extensively and several algorithms have been proposed. Abate and Whitt (1989) recommend using Theorem 1 in Section 1.2 (p. 23) of Takács (1962), which gives $Y_{i_0}(n,t)$ as a finite integral. We implemented this algorithm using a fifth-order Romberg integration, as described in Section 4.3 of Press, Flannery, Teukolsky and Vetterling (1988). Using double precision arithmetic, we observed that for the example in Figure 2 this algorithm agrees with our numerical inversion algorithm up to 11 or more significant places. Also, the two algorithms are comparable in speed (both took a few seconds on a SUN 2 workstation to compute ten points of the distribution). Of course, the transform inversion algorithm works for general service-time distributions as well without any loss of speed or accuracy. (We are unaware of alternate algorithms in the M/G/1 case). We also observed that the algorithm based on integration has problems (gets too slow or inaccurate) if *t* is very large or if the server utilization is close to 1 or exceeds 1. The transform inversion algorithm did not have problems in any of these cases. (Of course, it is possible to address the problems in the integration-based algorithm by fine tuning it based on the properties of the integrand, but we did not do this.)

6.3 The Transient Workload Distribution

Next we consider the transient workload distribution in an M/G/1 queue. Let $W(t)$ represent the workload (remaining service time of all customers in the system) at time *t* and let $W(t,x)$ be its CDF. Consider the two-dimensional transform

$$\tilde{w}(\xi,s) = \int_0^\infty \int_0^\infty e^{-\xi t} e^{-sx} d_x W(t,x)\, dt \ . \tag{6.14}$$

It can be shown that

$$\tilde{w}(\xi,s) = \frac{\{\hat{h}(s)\}^{i_0} - s\hat{P}_{i_0 0}(s)}{\xi - s + \lambda - \lambda\hat{h}(s)} \ , \tag{6.15}$$

where i_0 is the initial queue length (assuming that there has been a departure at $t = 0$) and $\hat{p}_{i_0 0}(s)$ is as given in (6.13). We actually invert the double transform of

$$\tilde{w}^c(\xi,s) = \int_0^\infty \int_0^\infty e^{-\xi t} e^{-sx} W^c(t,x)\, dx \ dt \ , \tag{6.16}$$

where $W^c(t,x) = 1 - W(t,x)$. It can be shown that

$$\tilde{w}(\xi,s) = \frac{1}{s\xi} - \frac{\tilde{w}(\xi,s)}{s} \ . \tag{6.17}$$

We do the transform inversion using the continuous-continuous variant of the algorithm in Section 2. In Figure 4 we plot the transient workload distribution at times $t = 2$, $t = 10$ and $t = 50$, assuming that the system starts empty at $t = 0$. The service-time distribution is gamma with mean 1 and SCV = 4 and the server utilization is 1.5, so that $W(t) \to \infty$ as $t \to \infty$. However, the transient workload is finite and Figure 4 shows how it progresses with time.

6.4 The Conditional Queue Length At Arrivals

We conclude with a discrete-discrete example to illustrate Section 3. For this purpose, let Q_j be the queue length observed by (just prior to) the j^{th} arrival. We shall calculate the conditional probability

$$p_{ik}^{(n)} = P(Q_{n+m} = k \,|\, Q_m = i) \tag{6.18}$$

in the M/M/1 queue. The double transform of $p_{ik}^{(n)}$ is given in Theorem 4 on p. 28 of Takács

(1962). We observed that there are two typographical errors in the formula. After correcting these, we get

$$P(z,\omega) \equiv \sum_{n=0}^{\infty} \sum_{k=0}^{\infty} p_{ik}^{(n)} z^k \omega^n = -\frac{g(\omega)[\mu-(\lambda+\mu)z]z^i}{[z-g(\omega)][\mu-\lambda z\omega g(\omega)]}$$
$$+\frac{(1-z)[\mu-(\lambda+\mu)g(\omega)][g(\omega)]^{i+1}}{[1-g(\omega)][z-g(\omega)][\mu-\lambda z\omega g(\omega)]}, \qquad (6.19)$$

where λ is the arrival rate, μ is the service rate and

$$g(\omega) = \frac{(\lambda+\mu)-\sqrt{(\lambda+\mu)^2-4\lambda\mu\omega}}{2\lambda\omega}. \qquad (6.20)$$

Figure 5 plots in log scale the conditional probability distribution of the queue length observed by the $(n+1)^{\text{st}}$ customer given that the first customer saw 10 in the queue (including the one in service). We consider four cases: $n = 2$, $n = 10$, $n = 100$ and $n = \infty$. The transient distributions approach the steady-state distribution as n gets large. it is interesting to note that the distribution drops to zero (shown by dotted line) whenever k exceeds $(n+10)$. This is because there cannot be more than $(n+10)$ in the queue at the arrival instant of the $(n+1)^{\text{st}}$ customer since the first arrival found 10 in the system.

We can also study this conditional distribution in the more general M/G/1 case using Theorem 11 on p. 70 of Takács (1962), but we do not do that.

Acknowledgment Our algorithm here builds on work on one-dimensional algorithms by Joseph Abate. We thank him for showing the way.

References

ABATE, J. and WHITT, W. (1989) Calculating time-dependent performance measures for the M/M/1 queue. *IEEE Trans. Commun.* 37, 1102-1104.

ABATE, J. and WHITT, W. (1992a) The Fourier-series method for inverting transforms of probability distributions. *Queueing Systems* 10, 5-88.

ABATE, J. and WHITT, W. (1992b) Numerical inversion of probability generating functions. *Opns. Res. Letters* 12, 245-251.

CHAMPENEY, D. C. (1987) *A Handbook of Fourier Theorems.* Cambridge University Press, Cambridge, England.

CHOUDHURY, G. L. and LUCANTONI, D. M. (1994) Numerical computation of a large number of moments with applications to asymptotic analysis. *Opns. Res.,* to appear.

CHOUDHURY, G. L., LUCANTONI, D. M. and WHITT, W. (1993) Numerical solution of $M_t/G_t/1$ queues, submitted.

CHOUDHURY, G. L, LUCANTONI, D. M. and WHITT, W. (1994) The busy period in the BMAP/G/1 queue, in preparation.

DAVIS, P. J. and RABINOWITZ, P. (1984) *Methods of Numerical Integration*, second ed., Academic Press, New York.

DITKIN, V. A. and PRUDNIKOV, A. P. (1962) *Operational Calculus in Two Variables and its Applications,* Pergamon Press, (English translation of 1958 Russian edition).

FELLER, W. (1971) *An Introduction to Probability Theory and its Applications,* vol. II, second ed., Wiley, New York.

FETTIS, H. E. (1955) Numerical calculation of certain definite integrals by Poisson's summation formula. *Math. Tables Other Aids Comput.* 9, 85-92.

FROLOV, G. and KITAEV, M. (1992) Personal communication.

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

HUNTLEY, E. and ZINOBER, A. S. I. (1979) Applications of numerical double Laplace transform algorithms to the solution of linear partial differential equations. *Computing* 21, 245-258.

JOHNSONBAUGH, R. (1979) Summing an alternating series. *Amer. Math. Monthly,* 86, 637-648.

LUCANTONI, D. M., CHOUDHURY, G. L. and WHITT, W. (1994) The transient BMAP/G/1 queue. *Stochastic Models* 10, to appear.

PRESS, W. H., FLANNERY, B. P., TEUKOLSKY, S. A. and VETTERLING, W. T. (1988) *Numerical Recipes, FORTRAN version,* Cambridge University Press, Cambridge, England.

SHEPHARD, N. G. (1991) Numerical integration rules for multivariate inversions. *J. Statist. Comput. Simul.* 39, 37-46.

SINGHAL, K., VLACH, J. and VLACH, M. (1975) Numerical inversion of multidimensional Laplace transforms. *Proc. IEEE* 63, 1627-1628.

TAKACS, L., (1962) *Introduction to the Theory of Queues,* Oxford University Press, New York.

TOLSTOV, G. P., (1976) *Fourier Series,* Dover, New York.

VAN DER POL, B. and BREMMER, H. (1955) *Operational Calculus,* Cambridge Press (reprinted, Chelsea Press, New York, 1987).

WIMP, J. (1981) *Sequence Transformations and Their Applications,* Academic Press, New York.

Aliasing error bound	$A_1 (= A_2)$	$\exp\left[\frac{A_1}{2l_1} + \frac{A_2}{2l_2}\right] / (4 l_1 t_1 l_2 t_2)$		
		$l_1 = l_2 = 1$	$l_1 = l_2 = 2$	$l_1 = l_2 = 3$
10^{-8}	19.114	5×10^7	8.8×10^2	16.2
10^{-12}	28.324	5×10^{11}	8.8×10^4	350

Table 1. Controlling the roundoff error by the choice of l_1 and l_2. (Here we assume that $t_1 = t_2 = 1$.)

Figure 1. The conditional busy-period distribution $P(X > x | N = n)$ in the M/G/1 queue in log scale, as a function of n when the arrival rate is 0.8 and the service-time distribution is gamma with mean 1 and SCV 4.

Figure 2. The transient queue-length distribution $P(Q(5) = n | Q(0) = 10)$ in the M/G/1 queue, in log scale, as a function of the service-time distribution when the arrival rate is 0.8 and the mean service time is 1.

Figure 3. The transient queue-length distribution $P(Q(t) = n | Q(0) = 10)$ in the M/G/1 queue, in log scale, as a function of time t when the arrival rate is 0.8 and the service-time distribution is gamma with mean 1 and SCV 4.

Figure 4. The transient workload complementary CDF $P(W(t) > x | W(0) = 0)$ for the unstable M/G/1 queue with arrival rate 1.5 and gamma service-time distribution with mean 1 and SCV 4.

Figure 5. The conditional probability of queue lengths at arrival epochs, $P(Q_{n+1} = k | Q_1 = 10)$, in the M/M/1 queue as a function of n when the traffic intensity is $\rho = 0.8$.