ON THE LAGUERRE METHOD
FOR NUMERICALLY INVERTING LAPLACE TRANSFORMS

by

Joseph Abate, Gagan L. Choudhury and Ward Whitt

AT&T Bell Laboratories

Room 1L-219, Holmdel, NJ 07733-3030
Email: joe@buckaroo.att.com
Room 1L-238, Holmdel, NJ 07733-3030
Email: gagan@buckaroo.att.com
Room 2C-178, Murray Hill, NJ 07974-0636
Email: wow@research.att.com

April 20, 1995

Revision: September 8, 1995

Abstract

The Laguerre method for numerically inverting Laplace transforms is an old established method based on the 1935 Tricomi-Widder theorem, which shows (under suitable regularity conditions) that the desired function can be represented as a weighted sum of Laguerre functions, where the weights are coefficients of a generating function constructed from the Laplace transform using a bilinear transformation. We present a new variant of the Laguerre method based on: (1) using our previously developed variant of the Fourier-series method to calculate the coefficients of the Laguerre generating function, (2) developing systematic methods for scaling, and (3) using Wynn's ϵ-algorithm to accelerate convergence of the Laguerre series when the Laguerre coefficients do not converge to zero geometrically fast. These contributions significantly expand the class of transforms that can be effectively inverted by the Laguerre method. We provide insight into the slow convergence of the Laguerre coefficients as well as propose a remedy. Before acceleration, the rate of convergence can often be determined from the Laplace transform by applying Darboux's theorem. Even when the Laguerre coefficients converge to zero geometrically fast, it can be difficult to calculate the desired functions for large arguments because of roundoff errors. We solve this problem by calculating very small Laguerre coefficients with low relative error through appropriate scaling. We also develop another acceleration technique for the case in which the Laguerre coefficients converge to zero geometrically fast. We illustrate the effectiveness of our algorithm through numerical examples.

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

Keywords: numerical transform inversion, Laplace transforms, Laguerre polynomials, Weeks' algorithm, accelerated summation, Wynn's ϵ-algorithm

Introduction

Our purpose in this paper is to contribute to the Laguerre method for calculating values of a real-valued function f on the nonnegative real line by numerically inverting its *Laplace transform*

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

which we assume is well defined, e.g., convergent and thus analytic for $Re(s) > 0$; see Doetsch [20]. The Laguerre method is based on the *Laguerre-series representation*

$$f(t) = \sum_{n=0}^\infty q_n l_n(t), \ t \geq 0 \ , \tag{2}$$

where

$$l_n(t) = e^{-t/2} L_n(t), \ t \geq 0 \ , \tag{3}$$

$$L_n(t) = \sum_{k=0}^n \binom{n}{k} \frac{(-t)^k}{k!}, \ t \geq 0 \ , \tag{4}$$

and

$$Q(z) \equiv \sum_{n=0}^\infty q_n z^n = (1-z)^{-1} \hat{f}((1+z)/2(1-z)) \ , \tag{5}$$

with L_n in (4) being the *Laguerre polynomials*, l_n in (3) the associated *Laguerre functions*, q_n in (2) the *Laguerre coefficients* and $Q(z)$ in (5) the *Laguerre generating function* (the generating function of the Laguerre coefficients). The Laguerre functions form an orthonormal basis for the space $L_2[0,\infty)$ of square integrable functions on the nonnegative real line, so that (2) is valid (in the sense of convergence in $L_2[0,\infty)$) for any f in $L_2[0,\infty)$ with

$$q_n = \int_0^\infty f(t) l_n(t) dt \ ; \tag{6}$$

see Laguerre [30] and Szegö [45]. Under extra regularity conditions, the Laguerre series in (2) converges to $f(t)$ pointwise for each t that is a continuity point of f.

The Laguerre-series representation of f in (2) can serve as a basis for inverting the Laplace transform $\hat{f}$ in (1) because the Laguerre generating function Q in (5) is expressed directly in terms of the Laplace transform $\hat{f}$. This fortunate situation occurs because the Laplace transform of the n^{th} Laguerre function has the special form,

$$\hat{l}_n(s) \equiv \int_0^\infty e^{-st} l_n(t) dt = 2(2s-1)^n/(2s+1)^{n+1} \ . \tag{7}$$

By (1), (2) and (7), the Laplace transform $\hat{f}$ can be expressed as

$$\hat{f}(s) = 2 \sum_{n=0}^\infty q_n (2s-1)^n/(2s+1)^{n+1} \ . \tag{8}$$

By using the transformation

$$z = T(s) = \frac{2s-1}{2s+1} \ , \quad s = T^{-1}(z) = \frac{1+z}{2(1-z)} \ , \tag{9}$$

we obtain (5) from (8). The inversion is carried out by calculating the Laguerre coefficients q_n using (5) and approximately summing the series (2). We will have more to say about these two important steps later.

This general approach to inverting Laplace transforms was evidently first proposed in 1935 by Tricomi [46] and Widder [51]. By the early 1960's the basic theory appeared in several textbooks; e.g., p. 250 of Van der Pol and Bremmer [47], p. 471 of Hille [25], p. 333 of Kaplan [26] and p. 426 of Moretti [35]. The first paper to specifically consider numerical inversion was evidently Ward [48] in 1954, but he did not propose a general procedure for computing the Laguerre coefficients q_n. The first full algorithms exploiting the Laguerre method as we know it today seem to have been developed in 1966 by Chen [10], Spinelli [40] and Weeks [50]. At AT&T Bell Laboratories a variant of the Laguerre method was developed by M. Eisenberg (related to the Weeks [50] algorithm, which he helped develop) and has been frequently used for applied probability problems. In 1969 the Laguerre method was reviewed in the perspective of Jacobi polynomials by Luke [31]. In 1971 a new variant of the Laguerre method plus an extensive literature review was provided by Piessens and Branders [36]. (See [36] for more early references.) Other more recent papers on numerical inversion by the Laguerre method (in chronological order) include: Davies and Martin [19], Weber [49], Wu and Ong [54], Lyness and Giunta [32], Garbow, Giunta, Lyness and Murli [22], [23] and Duffy [21]. The Laguerre method is now often called Weeks' method because of his early contribution [50]; see [19], [22], [23] and [32]. The Laguerre method is available from the ACM library of software algorithms in Algorithm 662 by Garbow et al. [23].

There is also a body of related literature related to the relationships (2)–(5) which is not concerned with numerical inversion of the Laplace transform $\hat{f}$. Instead, the sequence of Laguerre coefficients $\{q_n\}$ is regarded as a *Laguerre transform* of f, and attention is focused on how operations on functions f_i can be represented by corresponding operations on the associated Laguerre transforms $\{q_{in}\}$. The notion of the Laguerre transform was introduced in 1960 by McCully [34] and further studied by Keilson and Nunn [27], Keilson, Nunn and Sumita [28], Keilson and Sumita [29], Sumita [41], [42] and Sumita and Kijima [43], [44]. A significant idea of Keilson and Nunn [27] is to focus on the *associated differences* of $\{q_n\}$, i.e. on

$$p_n = q_n - q_{n-1}, \quad n \geq 1, \quad p_0 = q_0 \ . \tag{10}$$

The generating function of $\{p_n\}$ is

$$P(z) \equiv \sum_{n=0}^{\infty} p_n z^n = (1-z)Q(z) = \hat{f}((1+z)/2(1-z)) \ . \tag{11}$$

From (11) it follows that continuous convolution of functions f is associated with discrete convolution of the associated sequences $\{p_n\}$; i.e., if

$$f_3(t) = \int_0^t f_1(t-y) f_2(y) dy \tag{12}$$

and $\{p_{in}\}$ is the Laguerre (difference) transform of f_i, then

$$p_{3,n} = \sum_{k=0}^{n} p_{1,k} p_{2,n-k}, \quad n \geq 0 \ . \tag{13}$$

We will not make use of the differences p_n in (10) and their generating function $P(z)$ in (11). However, the theory of the Laguerre transform helps to understand the numerical inversion problem. In particular, useful background can be found in Keilson and Nunn [27].

We have become interested in the Laguerre method for numerically inverting Laplace transforms as a possible alternative to the Fourier-series method, which we have been using; see Abate and Whitt [3], [5], Abate, Choudhury and Whitt [1], Choudhury, Lucantoni and Whitt [14], Choudhury and Lucantoni [13], Choudhury, Leung and Whitt [11], [12] and Choudhury and Whitt [15] [16]. Earlier comparisons of different inversion algorithms by Davies and Martin [19], Garbow et al. [22] and Duffy [21] indicate that for some problems the Laguerre method performs better than the Fourier-series method (produces higher accuracy in less time), but for other problems it performs worse. Indeed, for 6 of 16 test problems in Davies and Martin [19] the Laguerre method gave poor results, but no clear explanation was given and no remedy was proposed. These conclusions have been verified in two more recent papers: see p. 170 of Garbow et al. [22] and p. 338 of Duffy [21]. Our goal here is to understand why the Laguerre method performs as it does and to propose some improvements.

First, we note that the Fourier-series method is actually quite closely related to the Laguerre method. In both cases the desired function f is represented as an expansion in terms of orthogonal functions, where the coefficients are expressed in terms of the Laplace transform $\hat{f}$. In the Fourier-series method, we begin by replacing f by a periodic function f_p constructed by aliasing, i.e., we let the periodic function be

$$f_p(t) = \sum_{k=-\infty}^{\infty} f(t + (2\pi k/h)) \tag{14}$$

for suitably small h, after damping f exponentially if necessary (i.e., replacing $f(t)$ by $e^{-\sigma t}f(t)$ for $\sigma > 0$) to guarantee that the series in (14) converges. For given t, we choose h sufficiently small so that $f_p(t)$ is sufficiently close to $f(t)$. The difference between f_p and f is the *aliasing error.* Since f_p is periodic, it has a Fourier series expansion, paralleling (2). Moreover, paralleling (5), it turns out that the Fourier coefficients can be expressed directly in terms of the Laplace transform values. Just as with (2), there is a further *truncation error* when we approximately sum the resulting Fourier series. The Fourier series can be expressed so that it tends to eventually be an alternating series. Hence, effective acceleration can be achieved with Euler summation [3], [5].

Given the relationships (2)–(5), there are two problems in implementing the Laguerre method: (i) calculating the Laguerre coefficients q_n given their generating function Q in (5) and (ii) calculating the sum in (2). It turns out that the first step is usually relatively easy unless q_n needs to be computed accurately even when it is very small, which is typically required for large t (see Sections 6 and 7). In Section 1 we show that our discrete variant of the Fourier-series method, which we call LATTICE-POISSON [3], [4], [14], applies to systematically and effectively invert the Laguerre generating function Q in (5). Thus, *we enhance, the Laguerre method by combining it with the Fourier-series method.*

The second step can be difficult because the Laguerre coefficients q_n can approach 0 slowly. In Section 2 we show that an examination of the nonlinear transformation T in (9) reveals whether q_n approaches 0 geometrically fast or not, as a function of the singularities of the Laplace transform $\hat{f}$. Previous authors such as Garbow et al. [22], [23] require that the good geometric rate of convergence prevails; otherwise the Laguerre method is called "unsuitable." However, we consider the case of non-geometric convergence as well as the case of geometric convergence.

We remark that we are primarily motivated by probability applications, so that the function of interest to us is often a *complementary cumulative distribution function* (ccdf) $F^c(t)$ (representing the probability of the interval (t, ∞)), which is a nonnegative decreasing function on $[0, \infty)$ with $F^c(0) \leq 1$ and $F^c(t) \to 0$ as $t \to \infty$. Our algorithm does not require this special structure, but this special structure can help perform numerical inversion. (We present both probability and non-probability examples in the paper.) However, even with this structure, the standard Laguerre method can encounter difficulties, as we will show.

It might be thought that geometric convergence of $\{q_n\}$ would hold for transforms $\hat{f}$ associated with functions with pure exponential tail, i.e., if

$$f(t) \sim \alpha e^{-\eta t} \text{ as } t \to \infty \ , \tag{15}$$

for constants α and η, with $\eta > 0$, where $a(t) \sim b(t)$ means that $a(t)/b(t) \to 1$ as $t \to \infty$. However, that is *not* the case, as is illustrated by the example

$$\hat{f}(s) = (2\sqrt{1+s} - 1)^{-1} \tag{16}$$

for which $f(t) \sim (1/2)e^{-2t/4}$ as $t \to \infty$; see Section 2. This is an example of an exponential mixture of inverse Gaussian distributions; see [6].

On the other hand, it might be thought that long-tail probability density functions with

$$f(t) \sim \alpha t^{-\beta} \text{ as } t \to \infty \tag{17}$$

necessarily lead to non-geometric convergence of $\{q_n\}$. However, that is *not* necessarily the case either, as is illustrated by the example

$$\hat{f}(s) = 1 - s + \frac{s^2}{2} \log(1 + 2s^{-1}) \ , \tag{18}$$

which is the Laplace transform of a Pareto mixture of exponential (PME) distributions pdf from [1] for which (17) holds with $\alpha = 1$ and $\beta = 3$. It turns out that the tail behavior of the Laguerre coefficients q_n depends not upon the tail behavior of $f(t)$ but upon the *smoothness* of $f(t)$ and its derivatives; alternatively, it depends primarily upon the behavior of the Laplace transform $\hat{f}(s)$ for s at infinity; see Section 2.

When the Laguerre coefficients q_n approach 0 geometrically fast, there usually is little difficulty in summing the Laguerre series, except when we want to calculate $f(t)$ for large t; see Section 6. We treat this case using a new scaling procedure. We also propose an acceleration technique exploiting the geometric convergence; see Section 8. When the Laguerre coefficients approach 0 slowly, we propose another acceleration technique to speed up convergence of (2). In particular, in Section 3 we propose Wynn's [55], [56] ϵ-algorithm for this purpose; e.g., see Chapter 6 of Wimp [53]. The ϵ-algorithm has been used with the Fourier-series method, e.g., see Crump [17] and Piessens and Huysmans [37], but we are unaware of any acceleration technique having been considered previously for the Laguerre method. Indeed, none of the previous work addresses the case of non-geometric convergence of the Laguerre coefficients or the difficulty at large t with geometric convergence. We find that the ϵ-algorithm often helps greatly, but sometimes not very much. However, in the latter case, we have been able to obtain good accuracy by exploiting *both* the ϵ-algorithm and judicious scaling; see Sections 4 and 5.

We summarize our proposed variant of the Laguerre algorithm in Section 9 and state our conclusions in Section 10. In a sequel to this paper we show how the Laguerre method can be extended to invert multidimensional Laplace transforms. As is well known, the basic orthogonal function structure extends immediately, using products of the one-dimensional Laguerre functions as basis elements. In fact, the multivariate generalization of (2) was already studied by Sumita and Kijima [43]. However, we do propose the first effective numerical inversion algorithm for multidimensional transforms via the Laguerre method by applying our multivariate Fourier-series method algorithm [14] to invert the resulting multidimensional Laguerre generating function. Furthermore, all enhancements introduced here can be used in each dimension of the multidimensional inversion.

1. Calculating the Laguerre Coefficients

Using the basic Hilbert space theory of the function space $L_2[0,\infty)$, we know that the squared norm of f is

$$||f||_2^2 \equiv \int_0^\infty f(t)^2 dt = \sum_{n=0}^{\infty} q_n^2 < \infty \tag{19}$$

for each f in $L_2[0,\infty)$, which implies that $q_n \to 0$ as $n \to \infty$. We can also bound q_n by making a further integrability assumption. Since $|l_n(t)| \leq 1$ for all n and t (see 22.14.12 on p. 786 of Abramowitz and Stegun [7] or Szegö [45]), we can apply (6) to deduce that

$$|q_n| \leq \int_0^\infty |f(t)||l_n(t)|dt \leq \int_0^\infty |f(t)|dt \ . \tag{20}$$

To take advantage of (19), we assume that f is integrable as well as square integrable. We are primarily interested in probability applications, where usually there will be nothing extra to verify. For example if f is a *probability density function* (pdf), then $|q_n| \leq 1$ for all n. Similarly, if f is a *complementary cumulative distribution function* (ccdf), i.e., if $f(t) = F^c(t)$ where $F^c(t) = 1 - F(t)$ for a *cumulative distribution function* (cdf) F on the nonnegative real line with mean m, then

$$|q_n| \leq \int_0^\infty F^c(t)dt = m \ . \tag{21}$$

The mean m could be calculated directly to know the bound in (20), e.g., as in Choudhury and Lucantoni [13], but in practice it is not necessary to know the bound in advance.

Moreover, it is not really necessary for the desired function $f(t)$ to be integrable or square integrable, because we can modify it to make it so. In particular, if f is not initially integrable or square integrable, then we can often make it so by *damping* f, i.e., by replacing f by

$$h(t) = e^{-\sigma t} f(t) \ , \quad t \geq 0 \ , \tag{22}$$

for $\sigma > 0$, which has transform $\hat{h}(s) = \hat{f}(s+\sigma)$. (This is a form of scaling, which we discuss further in Section 4. In general, σ has to be chosen so that $\sigma > \sigma_0$ where σ_0 is the real part of the rightmost singularity of $\hat{f}(s)$.) For example, if we were considering the constant function $f(t) = 1$ for all $t \geq 0$, which has Laplace transform s^{-1} (test function #5 in Davies and Martin [19]), then we would need to use damping, after which we have little difficulty. (Even without damping, it turns out that computing q_n would not be difficult in this case, because $q_n = 2(-1)^n,\ n \geq 0$). However, the series converges very slowly due to the very slow decay rate of $l_n(t)$ as $n \to \infty$, which we discuss in the next section.) Since the Laguerre coefficients can be regarded as bounded, inverting the Laguerre generating function to calculate the Laguerre coefficients q_n is relatively straightforward, if we can be satisfied with absolute accuracy of about 10^{-8}.

Given a bound for the Laguerre coefficients, we can apply the LATTICE-POISSON variant of the Fourier-series algorithm for generating functions in [3], [4], to calculate q_n with *guaranteed absolute error bound*. Given the generating function $Q(z)$ in (5), we can obtain expressions for the coefficients via Cauchy contour integrals, i.e.,

$$q_n = \frac{1}{2\pi i} \int_{C_r} Q(z)/z^{n+1} dz \ ,$$

where C_r is a circle about the origin of raidus r, $0 < r < 1$, such that $Q(z)$ is analytic in $\{z : |z| < r\}$. upon making the change of variable $z = re^{iu}$, we obtain the expression

$$q_n = \frac{1}{2\pi r^n} \int_0^{2\pi} Q(re^{iu}) e^{-iku} du \ . \tag{23}$$

If we calculate the integral (23) approximately by the trapezoidal rule, then we obtain

$$\begin{aligned} q_n \approx \bar{q}_n &= \frac{1}{2nr^n} \sum_{j=1}^{2n} (-1)^j Re(Q(re^{\pi ij/n})) \\ &= \frac{1}{2nr^n} \left\{ Q(r) + (-1)^n Q(-r) + 2\sum_{j=1}^{n} (-1)^j Re(Q(re^{\pi ji/n})) \right\} \ . \end{aligned} \tag{24}$$

Moreover, as shown in [3], [4], we can use the discrete Poisson summation formula to derive the *discretization error* $e_a \equiv \bar{q}_n - q_n$. Since the Poisson summation formula is based on creating a periodic extension of the original function by aliasing, e_a is also called the *aliasing error.* The *aliasing error* is

$$e_a = \sum_{j=1}^{\infty} q_{n+2jn} r^{2jn} \ . \tag{25}$$

If $|q_n| \leq C$, then the absolute aliasing error is bounded by

$$|e_a| \leq \frac{Cr^{2n}}{1 - r^{2n}} \approx Cr^{2n} \ . \tag{26}$$

Given C and n, we choose r less than 1 to make the aliasing error bound meet a prescribed error criterion such as 10^{-10}. If C is not known in advance, then we can reduce r later if it becomes evident that we underestimated C.

Moreover, the *relative aliasing error* is bounded by

$$|\frac{e_a}{q_n}| \le \sum_{j=1}^{\infty} |\frac{q_{n+2jn}}{q_n}| r^{2jn} \ . \tag{27}$$

Note that for the Laguerre method to be applicable $|q_n|$ should not grow fast with n and should actually decay with n for large n. This implies that $|q_{n+2jn}/q_n|$ should be small compared to 1 for large n and should remain an $O(1)$ quantity even for small n. Therefore, we should have $|q_{n+2jn}/q_n| \le K$ for all n for some constant K, which implies that

$$|\frac{e_a}{q_n}| \le \frac{Kr^{2n}}{1-r^{2n}} \approx Kr^{2n} \ . \tag{28}$$

Therefore, we can effectively control the relative aliasing error by choosing r^{2n} suitably small, say 10^{-10}.

Computing $\bar{q}_n$ via (24) involves a potential roundoff problem, because the factor $1/2nr^n$ in (24) can be small; see Remark 5.8 of [3]. We can address this problem by using the modification of the LATTICE-POISSON algorithm in Choudhury, Lucantoni and Whitt [14]. In particular, instead of (24) and (25), we can use $q_n = \bar{q}_n - e_a$, where

$$\bar{q}_n = \frac{1}{2lnr^n} \sum_{j=0}^{l-1} \sum_{k=-n}^{n-1} (-1)^j \exp(-\pi ik/l) Q(re^{(\pi i(j+lk)/nl)}) \tag{29}$$

with relative aliasing error

$$\frac{e_a}{q_n} = \sum_{j=1}^{\infty} \frac{q_{n+2jln}}{q_n} r^{2jln} \ . \tag{30}$$

If $|q_{n+2jln}/q_n| \le K$, then

$$|e_a/q_n| \le Kr^{2ln}/(1-r^{2ln}) \approx Kr^{2ln} \ . \tag{31}$$

Just as with (28), for C and n given, we choose r^{2l} such that the aliasing error bound meets a prescribed error criterion. This choice leaves a degree of freedom in the choice of r and l. As can be seen from (29), the required computation is proportional to l, but increasing l tends to reduce the roundoff error, because it reduces the factor $1/2lnr^n$ in (29). For one-dimensional inversions, usually $l = 1$ is sufficient to obtain about 10^{-8} accuracy. If not, $l = 2$ or $l = 3$ is usually sufficient. However, the roundoff error cannot be reduced below the machine precision, which is about 10^{-14}

for standard double-precision computations, even with large l. So there is little point in increasing l above 3.

Moreover, repeating the calculation with different values of l serves as an accuracy check. Different values of l correspond to numerical integration along different circles in the complex plane, so they represent totally different calculations.

Our proposed variant of the Fourier-series method to compute the Laguerre coefficients is quite close to what Piessens and Branders [36] and Garbow et al. [22], [23] do. However, the seemingly minor differences are important. First, Piessens and Branders [36] follow the procedure above but with $r = 1$, so they do not attain a guaranteed error bound when inverting the Laguerre generating function $Q(z)$. Garbow et al. [22] recognize the value of having $r < 1$, but they leave r as a control parameter. They do not have the (r, l) procedure to obtain an explicit error bound with roundoff error control. Instead, they use an elaborate procedure involving integration along several circles, which is more difficult to understand and implement than our (24) or (29).

2. Convergence of the Laguerre Series

Typically, the most challenging problem in the Laguerre method, if there is a problem, is to calculate the Laguerre series $\Sigma_{n=0}^{\infty} q_n l_n(t)$ in (2) given the Laguerre coefficients q_n. There is little difficulty in evaluating the Laguerre functions $l_n(t)$ to an absolute accuracy of about 10^{-8}; they can be calculated in a numerically stable way using the recursion

$$l_n(t) = \left(\frac{2n-1-t}{n}\right) l_{n-1}(t) - \left(\frac{n-1}{n}\right) l_{n-2}(t) \ , \tag{32}$$

starting with $l_0(t) = e^{-t/2}$ and $l_1(t) = (1-t)e^{-t/2}$; see [45]. As we have noted before, $|l_n(t)| \le 1$ for all n and t, and (assuming that f is integrable) $|q_n| \le C$ for all n. In addition, $l_n(t)$ approaches 0 as $n \to \infty$ for each positive t; in particular,

$$l_n(t) \sim \frac{1}{\pi^{1/2}(nt)^{1/4}} \cos(2\sqrt{nt} - (\pi/4)) \text{ as } n \to \infty \ ; \tag{33}$$

see p. 245 of Magnus, Oberhettinger and Soni [33].

However, note that the convergence rate of $l_n(t)$ as $n \to \infty$ in (33) is very slow. Hence, in practice $|l_n(t)|$ becomes nearly constant for large n. Hence, for an effective algorithm, we must rely on the decay rate of q_n as $n \to \infty$.

In many cases there is no difficulty at all. For example, consider the function

$$f(t) = e^{-(1+(t/2))} I_0(2\sqrt{t}), \ t \ge 0 \ , \tag{34}$$

where $I_0(t)$ is the modified Bessel function, with Laplace transform

$$\hat{f}(s) = (s + (1/2))^{-1} \exp(-(2s-1)/(2s+1)) \ . \tag{35}$$

The function f is related to the *Bessel pdf* in (11.2) of Abate and Whitt [6] with $r = 0$; i.e., $f(t) = e^{t/2}\beta_0(t)$ where $\beta_0(t)$ is the second Bessel pdf. The Laplace transform $\hat{f}$ in (35) above has the property that the associated Laguerre generating function is just

$$Q(z) = e^{-z} \ , \tag{36}$$

so that $q_n = (-1)^n/n!$ and the Laguerre algorithm works spectacularly well; e.g., $q_{20} = 0.4 \times 10^{-18}$.

However, the Laguerre coefficients do not always converge to 0 so rapidly. Indeed, a serious difficulty is that the Laguerre coefficients q_n may approach 0 slowly. We propose to address this problem when it exists by using appropriate scaling and an acceleration technique, in particular, the ϵ-algorithm, which we describe in the next section. Now we aim to understand why this slow convergence can occur.

First, assuming that $f \in L_2[0,\infty)$,which may require damping as in (22), we have $q_n \to 0$ as $n \to \infty$ by (19). The essential reason why $q_n \to 0$ slowly as $n \to \infty$ is *lack of smoothness* of f. Indeed, as shown on p. 326 of Keilson and Nunn [27], if f is not continuous, then $\sum_{n=0}^{\infty} |q_n| = \infty$, which implies that $q_n \to 0$ slowly. Unlike the Fourier-series method (see Section 6 of [3]), lack of smoothness evidently cannot be directly addressed by convolution smoothing, because convolution smoothing makes $f(t)$ defined for negative t. An important point is that lack of continuity of f is *not* the only reason the Laguerre coefficients q_n can converge to zero slowly. As indicated in Section 6 of Keilson and Nunn [27], lack of smoothness of derivatives of f also leads to slow convergence of q_n to zero. The more continuous derivatives f has, the faster q_n converges to 0.

Since the Laplace transform $\hat{f}(s)$ is presumably available, while the function $f(t)$ is not, it is natural to approach the slow convergence of q_n to zero via the transform $\hat{f}(s)$. From this perspective, a key to understanding the slow convergence is the *conformal mapping* T in (9). The mapping T maps the complex plane into itself. It is sometimes called a *bilinear map*, because it can be expressed via the equation

$$sz + \frac{z}{2} - s + \frac{1}{2} = 0 \ , \tag{37}$$

which is linear with respect to both s and z. The mapping T maps the positive half plane $\{s : Re(s) > 0\}$ into the *interior* of the unit circle, i.e., into $\{z : |z| < 1\}$, maps the negative half plane $\{s : Re(s) < 0\}$ outside the unit circle, i.e., into $\{z : |z| > 1\}$, and maps the imaginary axis $\{s : Re(s) = 0\}$ onto the unit circle $\{z : |z| = 1\}$. We consider Laplace transforms $\hat{f}$ that are analytic in the positive half plane, so that the Laguerre generating function $Q(z)$ in (5) is analytic inside the unit circle. Hence, the radius of convergence of $Q(z)$ is $R \geq 1$; e.g., see Sec. 2.4 of Wilf [52]. If $R > 1$, then the convergence of $\{q_n\}$ is geometric; i.e., $q_n = o(\epsilon^n)$ as $n \to \infty$ for any $\epsilon > R^{-1}$. On the other hand, if $R = 1$, then the convergence will not be geometric.

In summary, non-geometric convergence of q_n can occur only if the Laguerre generating function Q has a singularity on the unit circle $\{z : |z| = 1\}$. This can happen in two ways: first, if $\hat{f}$ has a singularity on the imaginary axis $\{s : Re(s) = 0\}$ and, second, if $\hat{f}$ has a singularity at infinity. At first, the possibility of this second case may not be obvious, but it can occur because, from (9), $|T(s)| \to 1$ as $|s| \to \infty$. Since $s^{-1} \to 0$ as $|s| \to \infty$, the second case occurs if and only if $\hat{f}(1/s)$ has a singularity at $s = 0$. If these two possibilities can be ruled out, then $R > 1$ and q_n converges to 0 geometrically fast, as noted in Theorem 3.1 of Keilson and Nunn [27]. (Keilson and Nunn [27] rule out a singularity on the imaginary axis by the condition that $e^{at}f(t)$ be integrable for some positive a; this implies that $\hat{f}(s)$ is analytic for $Re(s) > -a$.)

For further discussion, we restrict attention to transforms $\hat{f}$ whose singularities fall on the negative real axis. This class covers most transforms arising in probability applications. For s real, clearly $T(s)$ is also real. Note that $T(0) = -1$ and T is strictly increasing on $(-1/2, 0]$ and strictly decreasing on $(-\infty, -1/2)$ with $T(s) \to 1$ as $s \to -\infty$. Hence, for this class of transforms, Q can have a singularity on the unit circle only if $\hat{f}$ has a singularity at $s = 0$ or $s = -\infty$.

Thus the source of non-geometric convergence of q_n must be a singularity of $\hat{f}$ at $s = 0$ or $s = -\infty$. However, it is not necessary that the Laguerre generating function Q inherit the singularity of the Laplace transform $\hat{f}$ under the transformation T. For example, the PME Laplace transform in (18) has a singularity at 0, but the Laguerre generating function Q turns out to be analytic at $T(0) = -1$, and has radius of convergence $R > 1$. In this case we easily obtain high accuracy with small n in (2). Moreover, a singularity at 0 can always be removed by doing exponential damping; i.e., if $\hat{f}(s)$ has a singularity at $s = 0$, then $\hat{f}(s + \sigma)$ for $\sigma > 0$ has a singularity at $s = -\sigma$.

The more difficult case is the singularity of $\hat{f}$ at $s = -\infty$. For example, the Laplace transform $\hat{f}$ in (16) has a branch point singularity at $s = -\infty$, and Q has an algebraic singularity at $T(-\infty) = 1$, so that q_n does not converge geometrically to 0. Moreover, it is fairly common for Laplace transforms to have singularities at $s = -\infty$, as we will show below. Furthermore, these singularities evidently cannot be so easily removed as the singularities on the imaginary axis.

We can see the impact of singularities of Q on the unit circle by doing asymptotic analysis. The asymptotic analysis yields the asymptotic form of q_n as $n \to \infty$. In some cases, the generating function Q will have a pole on the unit circle. For example, if $f(t) = t$, $t \geq 0$, then $\hat{f}(s) = s^{-1}$ and $Q(z) = 2(1+z)^{-1}$, so that Q has a simple pole at $z = -1$ and $q_n = 2(-1)^n$. However, in many applications Q will have an *algebraic singularity* on the unit circle. With an algebraic singularity, we can apply Darboux's theorem; see Darboux [18], p. 179 of Wilf [52] and pp. 445–450 of Henrici [24].

We will actually apply only a simple form of Darboux's theorem. Suppose that Q has one singularity on the unit circle, which is an algebraic singularity at $z = 1$ (corresponding to a singularity of f at $s = -\infty$). Then $Q(z) = (1-z)^\beta G(z)$ where β is non-integer and G is analytic in the circle $\{z : |z| < 1 + \eta\}$ for some $\eta > 0$. In this setting, Darboux's theorem concludes that

$$q_n \sim G(1)/n^{1+\beta}\Gamma(-\beta) \text{ as } n \to \infty \ , \tag{38}$$

where $\Gamma(x)$ is the gamma function. The full Darboux's theorem gives more terms, but we only make use of the single term in (38).

To illustrate, we give some probability examples, drawing on Abate and Whitt [6]. In these examples we know the function f as well as its transform, so that these examples are intended to provide insight rather than to be serious candidates for inversion. In these examples we can directly see that the function $f(t)$ lacks smoothness at the origin. For harder examples, for which only $\hat{f}(s)$ is available, we would really need Darboux's theorem. In the first four examples, we can apply Darboux's theorem to show that (38) holds for $0 < \beta < 1$, so that the rate of convergence of q_n to 0 is very slow in each case. In each example we discuss the pdf's. Afterwards we indicate how to relate pdf's and ccdf's.

Example 2.1. First consider the *gamma pdf* with (positive) shape parameter ν and (positive) mean ν/λ, namely,

$$f(t) = \Gamma(\nu)^{-1}\lambda^\nu t^{\nu-1}e^{-\lambda t} \ , \quad t \geq 0 \ , \tag{39}$$

which has Laplace transform

$$\hat{f}(s) = (1 + s/\lambda)^{-\nu} \ . \tag{40}$$

Clearly f in (39) is a relatively nice and well-behaved function for each ν and λ, but $\hat{f}$ has branch point singularities at $s_1 = -1/\lambda$ and $s_2 = -\infty$ for all non-integer ν. Note that $f(t)$ lacks smoothness at $t = 0$ for $0 < \nu < 1$ and that higher derivatives of $f(t)$ lack smoothness at $t = 0$ for noninteger $\nu > 1$.

To be more specific, let $\lambda = 1$ and $\nu = 1/2$. Then $\hat{f}(s) = (1+s)^{-1/2}$ and $Q(z) = 1/\sqrt{2(1-z)}$, so that Darboux's theorem yields

$$q_n = (1/4^k\sqrt{2})\binom{2k}{k} \sim 1/\sqrt{2\pi n} \text{ as } n \to \infty \ . \tag{41}$$

We remark that this gamma (1/2) pdf example is a damped version of the transform $s^{-1/2}$ with inverse $(\pi t)^{-1/2}$, which is Example 9 in Davies and Martin [19], for which the Laguerre method performed poorly; also see Duffy [21].

Example 2.2. Next let $f(t)$ be the density of the first-moment cumulative distribution function of regulated Brownian motion, as in (1.10) and (4.5) of Abate and Whitt [2] and in (8.6) and (9.15) of Abate and Whitt [6]; i.e., let

$$f(t) = (2\pi t)^{-1/2}e^{-t/4} - 1 + \Phi(\sqrt{t/2}), \ t \geq 0 \ , \tag{42}$$

where $\Phi(t)$ is the standard (mean 0, variance 1) normal cdf, which has Laplace transform

$$\hat{f}(s) = 2/(1+\sqrt{1+4s}) \ . \tag{43}$$

This example is similar to (16). In this case

$$Q(z) = \frac{1}{\sqrt{1-z}}\left(\frac{2}{\sqrt{1-z}+\sqrt{3+z}}\right) \ , \tag{44}$$

so that Darboux's theorem yields

$$q_n \sim 1/\sqrt{n\pi} \text{ as } n \to \infty \ . \tag{45}$$

Given (42), (45) is not a surprise because $f(t) \to \infty$ as $t \to 0$.

Example 2.3. As in Section 10 of [6], let f be *Feller's first Bessel pdf*; i.e.,

$$f(t) = \sqrt{\mu/2\pi t^3}\exp(-t/2\mu)[(1-\exp(-(\mu^2-1)t/2\mu))/(\mu-1)], \ t \geq 0 \ , \tag{46}$$

and

$$\hat{f}(s) = (\mu-1)^{-1}(\sqrt{\mu^2+2\mu s} - \sqrt{1+2\mu s}) \tag{47}$$

for $\mu > 1$. To be more specific, let $\mu = 2$. Then

$$\hat{f}(s) = 2\sqrt{1+s} - \sqrt{1+4s} \ , \tag{48}$$

$$\begin{aligned} Q(z) &= (1-z)^{-3/2}(\sqrt{6-2z} - \sqrt{3+z}) \\ &= (1-z)^{-1/2}\left(\frac{3}{\sqrt{6-2z}+\sqrt{3+z}}\right) \end{aligned} \tag{49}$$

and Darboux's theorem yields

$$q_n \sim 3/4\sqrt{\pi n} \text{ as } n \to \infty \ . \tag{50}$$

Example 2.4. As in Section 11 of [6], let f be *Feller's second Bessel pdf* with parameter $r = -1/2$; i.e.,

$$f(t) = (\pi t)^{-1/2} e^{-(1+t)} \cosh(2\sqrt{t}), \ t \geq 0 \ , \tag{51}$$

and

$$\hat{f}(s) = (1+s)^{-1/2} \exp(-s/(1+s)) \ . \tag{52}$$

(Also use 10.2.14 of Abramowitz and Stegun [7].) In this case,

$$Q(z) = (1-z)^{-1/2}((3-z)/2)^{1/2} \exp(-(1+z)/(3-z)) \ , \tag{53}$$

so that Darboux's theorem yields

$$q_n \sim 1/e\sqrt{\pi n} \text{ as } n \to \infty \ . \tag{54}$$

Example 2.5. As in (2.2) of [6], let f be the *exponential-integral pdf*; i.e.,

$$f(t) \equiv E_1(t) \equiv \int_t^\infty x^{-1} e^{-x} dx = \int_0^1 \mu^{-1} e^{-t/\mu} d\mu \tag{55}$$

and

$$\hat{f}(s) = \log(1+s)/s \ . \tag{56}$$

In this case,

$$Q(z) = (2/(1+z)) \log((3-z)/2(1-z)) \ . \tag{57}$$

In this case $Q(z)$ has singularities at $z = 1$ and 3, which are the branch points of the logarithm. (Note that $Q(z)$ does not have a singularity at $z = -1$, in fact, $Q(-1) = 1/2$.) Darboux's theorem can not be used in this example because it only applies to algebraic singularities. However, note that as $z \to 1$, $Q(z) \to -\log(1-z)$. Therefore,

$$q_n \sim 1/n \text{ as } n \to \infty \ . \tag{58}$$

Example 2.6. As in Section 13 of [6], let f be the *first theta pdf*, i.e., the pdf of the first passage time for standard reflected Brownian motion from 0 to $\sqrt{2}/2$; i.e.,

$$f(t) = \pi \sum_{n=1}^{\infty} (-1)^{n+1} (2n-1) \exp(-\pi^2 (n-(1/2))^2 t) \tag{59}$$

and

$$\hat{f}(s) = 1/\cosh(\sqrt{s}) \ . \tag{60}$$

In this case

$$Q(z) = 1/((1-z)\cosh(\sqrt{(1+z)/2(1-z)}) \ . \tag{61}$$

The transform $\hat{f}$ in (60) has a simple pole at $s = -\infty$, but the corresponding singularity in $Q(z)$ at $z = 1$ is an essential singularity. Hence, Darboux's theorem does not apply to this example. However, we will show that q_n approaches 0 slowly numerically in Section 5.

Our examples above have been for pdf's, but we are usually interested in ccdf's instead. However, it is easy to relate the asymptotic behavior of the Laguerre coefficients of a ccdf to the asymptotic behavior of the associated Laguerre coefficients of the pdf. For this purpose, suppose the pdf f satisfies (2) and let the ccdf be represented as

$$F^c(t) = \int_t^\infty f(u)du = \sum_{n=0}^\infty q'_n l_n(t) \ . \tag{62}$$

with $Q'(z) = \sum q'_n z^n$. Since $\hat{F}^c(s) = (1 - \hat{f}(s))/s$,

$$Q'(z) = \frac{-2(1-z)}{(1+z)}(Q(z) - 1) \ . \tag{63}$$

Hence, if Q has radius of convergence $R > 1$ with a singularity at R (which will be the case if $\hat{f}$ has all its singularities on the negative real axis and has no singularities at $s = -\infty$ or $s = 0$), then $q'_n \sim [2(R-1)/(1+R)]q_n$ as $n \to \infty$. On the other hand, if Q has a radius of convergence 1 with only one algebraic singularity on the unit circle at $z = 1$, then the asymptotic behavior of q'_n is the same as the coefficients of $(1-z)Q(z)$, which are $q_{n-1} - q_n$. Hence, if $q_n \sim Cn^{-\beta}$ as $n \to \infty$ for $\beta > 0$, then $q'_n \sim C\beta n^{-(1+\beta)}$ $n \to \infty$. (This can be deduced directly or by Darboux's theorem.)

In summary, when there is geometric rate of convergence of the Laguerre coefficients, this rate applies to *both* the pdf and the ccdf. On the other hand, when there is non-geometric rate of convergence, the Laguerre coefficients of the ccdf tend to approach 0 somewhat faster than the Laguerre coefficients of the pdf. In particular, for ccdf's (but not for pdf's) we can guarantee that the truncation error is asymptotically negligible, i.e.,

$$|\sum_{n=m}^\infty q_n l_n(t)| \le \sum_{n=m}^\infty |q_n| < \infty \tag{64}$$

because $q_n \sim Cn^{-(1+\beta)}$ as $n \to \infty$ for $\beta > 0$. However, the rate of convergence of q_n for the ccdf's in the examples above is still quite slow.

3. Application of Wynn's ϵ-Algorithm

We propose using Wynn's [55], [56] ϵ-algorithm to accelerate convergence of the Laguerre series (2) when the rate of convergence is slow. We will show in Section 5 that this algorithm together with suitable scaling can indeed greatly improve the accuracy. The ϵ-algorithm is defined by the recursion

$$\epsilon_{k+1}^{n} = \epsilon_{k-1}^{n+1} + (\epsilon_{k}^{n+1} - \epsilon_{k}^{n})^{-1} , \tag{65}$$

where $\epsilon_{-1}^{n} = 0$ and $\epsilon_{0}^{n} = S_n$ for all n, where S_n is the n^{th} partial sum of the series (2); see p. 138 of Wimp [53]. The final approximation is ϵ_{2m}^{n} for suitable n and m. We typically use $n = 100$ and $m = 6$. There is typically significant improvement as m increases from 1 to 2 and 3, but improvement slows by about $m = 6$.

As discussed in Wimp [53], pp. 3–5 and Chapters 5 and 6, the ϵ-algorithm is a *nonlinear lozenge algorithm* implementing the *Schmidt transformation*; as in Shanks [38]. Moreover, the ϵ-algorithm result ϵ_{2m}^{n} for a specific t is equivalent to the $[n + m/m]$ *Padé approximant*; see pp. 112, 131, 138 of Wimp [53]. Knowing these connections, useful additional background can be obtained from Baker [8] and Bender and Orszag [9]. In contrast to Euler summation used in [3]–[4] and [14], the ϵ-algorithm is often effective for non-alternating series. See Smith and Ford [39] for numerical comparisons. In Section 5 we show that the ϵ-algorithm is effective for the examples in Section 2 in which the sequence $\{q_n\}$ converges to zero slowly.

It should go without saying that there are no guarantees in advance with the ϵ-algorithm. To understand why, note that the recursion (65) can behave badly if S_n converges too quickly. In particular, if $S_n = S_{n+1}$, then the calculation of e_1^n involves a division by 0. However, we can be fairly confident about the success of the ϵ-algorithm after the computation if accuracy checks indicate consistent answers. By computing ϵ_{2m}^{n} for $m = 0, 1, \ldots, 6$, we can clearly see the impact of the ϵ-algorithm. Moreover, we can use the scaling discussed in the next section to obtain independent accuracy checks.

4. Scaling

Ever since Weeks [50], it has been recognized that the Laguerre method can be enhanced by scaling; e.g., see Garbow et al. [22], [23]. We scale with two positive real parameters σ and b.

$$f_{\sigma,b}(t) = e^{-\sigma t} f(t/b), \ t \geq 0 \ . \tag{66}$$

We then apply the Laguerre algorithm to $f_{\sigma,b}$ and recover f by

$$f(t) \equiv f_{0,1}(t) = e^{\sigma bt} f_{\sigma,b}(bt) \ . \tag{67}$$

The Laguerre generating function associated with $f_{\sigma,b}$ is

$$Q_{\sigma,b}(z) = \frac{b}{1-z}\hat{f}\left(\frac{b(1+z)}{2(1-z)} + b\sigma\right) \ . \tag{68}$$

The scaling parameter σ moves the singularities of $\hat{f}$ to the left by σ. Hence, σ is important for eliminating the bad effect of a singularity in $\hat{f}$ at 0, as discussed in Section 2. However, σ cannot help with singularities at infinity, as in the six examples in Section 2.

Weeks [50] and Garbow et al. [22] [23] all noted that the scaling parameter b can help, but a systematic procedure for setting b has yet to be developed. Weeks [50] noted that the Laguerre method should be more effective if the Laguerre functions are evaluated where they are oscillating. This leads to the restriction $bt \le 4n$, where n is the number of terms used. However, this restriction does not adequately explain the benefits of b. We have found that in the cases of slow convergence the ratio $|q_n|/|q_0|$ can be decreased significantly by increasing b, say from $b = 1$ to $b = 10$. Even though the new sequence $\{q_n\}$ still converges to 0 non-geometrically, the smaller ratio $|q_n|/|q_0|$ significantly improves the accuracy.

Another reason for using the scaling parameters σ and b is that they provide a powerful accuracy check. Since the computation is very different with different scaling pairs (σ, b), we can estimate the accuracy of the results by doing the same computations with two or more different scaling pairs.

5. Numerical Examples Treating Slow Convergence

In this section we present numerical examples of the Laguerre algorithm applied to treat the case in which the Laguerre coefficients q_n converge to 0 slowly. We show that using both the ϵ-algorithm in Section 3 and scaling with b (but not σ) in Section 4 can greatly improve the results. We show the scaling improvement with $b = 10$, but the exact choice is not critical. In fact, $b = 15$ or $b = 20$ provides similar (usually slightly better) improvement.

In all numerical examples we use the notation AeB to indicate $A \times 10^{-B}$.

Example 5.1. We start by considering the gamma distribution with shape parameter 1/2 in Example 2.1. In particular, we consider the ccdf $F^c(t)$, which has Laplace transform

$$\hat{F}^c(s) = s^{-1}(1 - \hat{f}(s)) = s^{-1}(1 - (1+2s)^{-1/2}) \ , \tag{69}$$

where $\hat{f}$ is the Laplace transform of the pdf in (40) with $\lambda = 1$ and $\nu = 1/2$. As indicated in Section 2, $q_n \sim 1/2\sqrt{2\pi n^3}$ as $n \to \infty$.

For this example, we compare exact values of the ccdf with numerical values from four variants of the Laguerre algorithm. We do the inversion with and without the ϵ-algorithm, and with and without scaling. Scaling means using $b = 10$, while no scaling means using $b = 1$. Without the ϵ-algorithm, we use 100 terms in the Laguerre series (2). When we use the ϵ-algorithm, we use the sixth-order version; i.e., we use e_{2m}^n in (65) for $m = 6$ and $n = 100$. This requires that we compute q_n for $0 \leq n \leq 113$.

The exact values are obtained using the Fourier-series method, using the accuracy check based on computations with different l parameters, as in [14]. The exact values have been checked to all digits given.

		$b = 1$	$b = 1$	$b = 10$	$b = 10$
t	exact	without epsilon	with epsilon	without epsilon	with epsilon
0	1.000000000	.960	.9824	.9875	.9945
1	.317310508	.31745	.317289	.3173096	.31731040
2	.157299207	.15742	.157304	.157308	.15729922
3	.83264517e-1	.8333e-1	.832674e-1	.83273e-1	.83264521e-1
4	.45500264e-1	.4562e-1	.455014e-1	.45495e-1	.45500250e-1
5	.25347319e-1	.2537e-1	.253457e-1	.25352e-1	.25347324e-1
6	.14305878e-1	.1422e-1	.143037e-1	.14310e-1	.14305902e-1
8	.46777349e-2	.4647e-2	.467711e-2	.46741e-2	.46777307e-2
10	.15654022e-2	.15626e-2	.156495e-2	.15623e-2	.15654026e-2

Table 1. A comparison of four variants of the Laguerre algorithm with exact results for the gamma (1/2) ccdf in Example 5.1. The number of terms is 100 without the ϵ-algorithm and 113 with the ϵ-algorithm.

From Table 1, we see that the accuracy of the basic Laguerre algorithm without refinements is not good (only 2 or 3 digits) for this example. The epsilon algorithm without scaling and scaling without the epsilon algorithm both improve the results in roughly the same way (one to two extra digits). For small t, scaling has more benefit; for large t, the epsilon algorithm has more benefit. However, taking both measures clearly provides dramatic improvement.

Examples 2.2–2.4 have a structure similar to Example 2.1 and yield similar numerical results, so we do not display them.

Example 5.2. We next consider the exponential-integral ccdf in Example 2.5, for which the Laplace transform has a logarithmic singularity. In particular, we consider the Laplace transform

$\hat{F}^c(s) = s^{-1}(1 - \hat{f}(s))$ for $\hat{f}$ in (56). In this case, the Laguerre coefficients q_n of the ccdf decay somewhat faster, in particular, $q_n \sim 1/n^2$ as $n \to \infty$. We carry out the same experiment for the exponential-integral ccdf, once again using the Fourier-series method to compute the exact values.

Numerical results for the exponential-integral ccdf are given in Table 2. The basic Laguerre algorithm performs even worse than for the gamma (1/2) ccdf in Table 1, at least for the larger times. Here again we see that both scaling and the ϵ-algorithm provide significant improvement separately. However, as before, both together again provide dramatic improvement.

		$b = 1$	$b = 1$	$b = 10$	$b = 10$
t	exact	without epsilon	with epsilon	without epsilon	with epsilon
0	1.000000000	.9902	.9895	.99902	.99979
1	.148495507	.14856e-1	.148483	.14849532	.148495486
2	.37534266e-1	.37595e-1	.375382e-1	.3753416e-1	.37534293e-1
3	.10641925e-1	.10678e-1	.106435e-1	.1064188e-1	.106419267e-1
4	.31982292e-2	.326e-2	.319845e-2	.31973e-2	.31982288e-2
5	.99646904e-3	.1008e-2	.9958e-3	.9972e-3	.99646916e-3
6	.318257463e-3	.274e-3	.31849e-3	.31893e-3	.318257469e-2
7	.103509844e-3	.137e-3	.10371e-3	.10341e-3	.103509846e-3
8	.34137645e-4	.18e-4	.3382e-4	.3357e-4	.34137666e-4

Table 2. A comparison of four variants of the Laguerre algorithm with exact results for the exponential-integral ccdf in Example 5.2. The number of terms is 100 without the ϵ-algorithm and 113 with the ϵ-algorithm.

Example 5.3. Finally, we consider the theta ccdf in Example 2.6, for which the Laguerre generating function Q has an essential singularity at $z = 1$. In particular, we consider the transform $\hat{F}^c(s) = s^{-1}(1 - \hat{f}(s))$ for $\hat{f}$ in (60). We repeat the previous experiment for this example and display the results in Table 3. In this case, scaling (by $b = 10$) separately is more effective than the ϵ-algorithm alone, but both together again provide dramatic improvement.

		$b = 1$	$b = 1$	$b = 10$	$b = 10$
t	exact	without epsilon	with epsilon	without epsilon	with epsilon
0.0	1.000000000	.9913	.99918	1.000037	.9999945
0.5	.370777430	.37132	.37083	.37077708	.370777430
1.0	.107977044	.10818	.107968	.10797695	.107977031
1.5	.314443118e-1	.3117e-1	.31463e-1	.3144415e-1	.31444363e-1
2.0	.91569903e-2	.928e-2	.91503e-2	.915687e-2	.91569942e-2
2.5	.26666340e-2	.2608e-2	.26681e-2	.266673e-2	.26666334e-2
3.0	.77655831e-3	.844e-3	.7755e-3	.77647e-3	.776491e-3
3.5	.22614382e-3	.12e-3	.2294e-3	.22622e-3	.2261445e-3
4.0	.65856006e-4	.20e-3	.619e-4	.65899e-4	.658572e-4

Table 3. A comparison of four variants of the Laguerre algorithm with exact results for the theta ccdf in Example 5.3. The number of terms is 100 without the ϵ-algorithm and 113 with the ϵ-algorithm.

We conclude this section by pointing out how the ratio $|q_{100}|/|q_0|$ decreased by increasing b from 1 to 10. For the examples in Tables 1, 2 and 3, the ratio decreased, respectively, from 3.4×10^{-4} to 4.5×10^{-5}, from 2.6×10^{-4} to 7.7×10^{-5}, and from 5.0×10^{-4} to 2.9×10^{-7}.

6. Difficulties with Large Arguments

In this section we show that the basic Laguerre algorithm can encounter serious difficulties even when q_n has a fast geometric decay rate.

Example 6.1. To make this point, we consider the first test function included in ACM Algorithm 662 [23], namely,

$$\hat{f}(s) = \frac{1}{s+0.5} + \frac{1}{s^2} + \frac{1}{1+(s+0.2)^2} \ , \tag{70}$$

which has the known inverse

$$f(t) = e^{-0.5t} + t + e^{-0.2t} \sin t \ . \tag{71}$$

This test example is the sum of test examples 3, 4 and 6 from Davies and Martin [19], for which the Laguerre method performed very well. However, Davies and Martin [19] only considered relatively small times, in particular, they considered 30 time points, all with $t \leq 15$. In contrast, we will also consider larger times.

Because of the s^{-2} term, the transform $\hat{f}$ has a singularity at 0, which can be removed by exponential damping. Accordingly, the ACM algorithm uses the scaling $(\sigma = 1, \ b = 1)$ for this example. We compare the results of applying the ACM algorithm with this scaling to the exact values of $f(t)$ for several values of t in Table 4.

		ACM algorithm
t	exact	$\sigma = 1,\ b = 1$
0.05	1.074792	1.074792
0.10	1.149086	0.149086
0.50	1.712603	1.712603
1.	2.295469	2.295469
5.	4.729316	4.729316
10.	9.933113	9.933113
15.	15.03293	15.03293
30.	29.99755	29.9790
50.	49.99999	−1.2e7
90.	90.00000	−3.8e23
200.	200.00000	—
400.	400.00000	—
800.	800.00000	—
1200.	1200.00000	—

Table 4. A comparison of ACM algorithm 662 with the exact solution for the three-term test function in (67). Our modified algorithm matches the exact values up to all displayed digits.

From Table 4, we see that the ACM algorithm performs very well for smaller t, but begins to break down at $t = 30$ and completely fails for $t \geq 50$. The difficulty is not due to slow convergence of q_n. The ACM algorithm uses $n = 63$ and obtains $q_{63} = -3.6 \times 10^{-14}$. (The true $|q_{63}|$ is even smaller.) No improvement is obtained by increasing n to 100 or 300. Furthermore, the choice of the scaling parameters ($\sigma = 1,\ b = 1$) is not too critical. For four other scaling settings used in the ACM algorithm, the algorithm also always crashes, with the critical time falling between $t = 15$ and $t = 50$. The problem is not just with the ACM algorithm either. We tried the same example with our LATTICE-POISSON algorithm in Section 1 and got essentially the same results.

The difficulty is that, *for large t, we need low relative error in q_n*. For large t, the dominant terms $e^{\sigma bt} q_n l_n(bt)$ occur for relatively large n, where q_n is very small, but it is offset by very large values of $e^{\sigma bt}$, while $l_n(bt)$ is near 1. For smaller $n, l_n(bt)$ is much smaller than 1, making the whole term $e^{\sigma bt} q_n l_n(bt)$ negligible. The computation will be satisfactory if we can calculate q_n with low relative error.

In many examples, such as the ccdf's, the function $f(t)$ will be very small when t is large, so that we may not care about large t. However, even then, we will encounter difficulties if we want to calculate $f(t)$ for large t with suitably small relative error.

We conclude this section by pointing out that the problem when we apply LATTICE-POISSON is the relative roundoff error, not the relative aliasing error. Since we will tend to have $|q_{n+2jln}| \leq$

$|q_n|$, the relative aliasing error in (27) should not be substantially worse than the absolute aliasing error bound in (26).

However, l in formula (24) tends to control only the *absolute roundoff error* in q_n. We need to take extra measures to control the *relative roundoff error.*

7. Achieving Low Relative Error in the Laguerre Coefficients

In this section we indicate how to calculate q_n with low relative error, so that we can calculate $f(t)$ for large t. Following Choudhury and Lucantoni [13], we introduce the scaled generating function

$$Q_\alpha(z) \equiv \sum_{n=0}^{\infty} q_n^{(\alpha)} z^n = \sum_{n=0}^{\infty} q_n \alpha^n z^n = Q(\alpha z) \ , \tag{72}$$

where α is chosen so that $q_n^{(\alpha)}$ remains $O(1)$ for all n. This scaling evidently is new in the context of the Laguerre method. The standard inversion algorithm maintains low absolute error in $q_n^{(\alpha)}$, implying that it will also maintain low relative error in q_n.

If $|q_n| \sim C\beta^n$ as $n \to \infty$, then a natural candidate for α should be β^{-1}. However, in general the asymptotic decay rate β is unknown. Moreover, even if it is known, it is often not appropriate for all n. As in Choudhury and Lucantoni [13], we can estimate the appropriate scaling from the most recent two computed values; i.e., when we compute q_n, we use the scale parameter

$$\alpha_n = |q_{n-2}/q_{n-1}| \ . \tag{73}$$

Given that $|q_n| \sim C\beta^n$ as $n \to \infty$, then $\alpha_n \to 1/\beta$ as $n \to \infty$, as it should.

Through extensive numerical experiments, we observed that the scaling (72) and (73) works in the current context as well, provided σ is not too large. Also we do not use this scaling for the first few values of n when q_n is $O(1)$ but changes erratically. Also if the ratio is less than 1, then we use $\alpha = 1$.

In particular, we observed that we can compute all the exact values for Example 6.1 in our Table 4 with our Laguerre algorithm provided we choose $\sigma = 0.05$ and $b = 1$ and use the scaling in (69) and (70) whenever $|q_{n-1}| < 0.1$. This happens for $n - 1 > 40$.

In this case, $q_{162} = 9.189 \times 10^{-12}$, $q_{197} = -9.814 \times 10^{-15}$ and $q_{350} = 8.0609 \times 10^{-28}$. Hence, without scaling, there will be significant errors for $n \geq 162$ and serious errors for $n \geq 197$. For small t, the convergence occurs for $n < 162$, so the standard algorithms compute accurately. However, for $t = 1200$,

$$|e^{\sigma bt} q_n l_n(bt)| < 10^{-30} \text{ for all } n \leq 197 \ . \tag{74}$$

Therefore, it is clear that all the significant contribution to the final answer comes from values of n for $n > 197$. It turns out that, with the new scaling in this section, $n = 500$ yields the correct answer for $t = 1200$.

We observed that the scaling introduced in this section is not effective if the decay rate of $|q_n|$ is slower than geometric. However, it remains pretty effective if the decay rate is faster than geometric. In particular, for the functions $f(t), \hat{f}(s)$ and $Q(z)$ given in (34), (35) and (36), $q_n = (-1)^n/n!$. In this case, with decay rate faster than geometric, we observed that the scaling in (73) is very effective and may be used to compute $f(t)$ accurately even for very large t.

8. Extrapolation with a Geometric Rate

The frequently occurring asymptotic geometric rate of convergence of q_n allows us to introduce a further refinement to the Laguerre algorithm. With the (σ, b)-scaling in Section 4, the inversion formula is

$$f_n(t) = e^{\sigma bt} \sum_{k=0}^{n} q_k l_k(bt) \ , \tag{75}$$

where the *truncation error* is

$$e_n(t) \equiv f(t) - f_n(t) = e^{\sigma bt} \sum_{k=n+1}^{\infty} q_k l_k(bt) \ . \tag{76}$$

We can exploit the geometric rate of convergence (when it holds) to approximate the truncation error by replacing q_n with $C\beta^n$, where C and β are chosen to satisfy $q_k = C\beta^k$ for $k = n-1$ and $k = n$, which yields

$$\beta = q_n/q_{n-1} \text{ and } C = q_n\beta^{-n} \ . \tag{77}$$

Now the approximate truncation error can be written as

$$e_n(t) \approx e^{\sigma bt} \sum_{k=0}^{\infty} C\beta^k l_k(bt) - e^{\sigma bt} \sum_{k=0}^{n} C\beta^k l_k(bt) \ . \tag{78}$$

Using the generating function of Laguerre polynomials, 22.9.15 of Abramowitz and Stegun [7], we can sum the first term in (78) to obtain

$$e_n(t) \approx \frac{C}{1-\beta} \exp(bt(\sigma - \frac{1}{2} - \frac{\beta}{1-\beta})) - e^{\sigma bt} \sum_{k=0}^{n} C\beta^k l_k(bt) \ . \tag{79}$$

Combining (75), (76) and (79), we get the refined inversion formula

$$f_n^{(r)}(t) = e^{\sigma bt} \sum_{k=0}^{n} (q_k - C\beta^k) l_k(bt) + \frac{C}{1-\beta} \exp(bt(\sigma - \frac{1}{2} - \frac{\beta}{1-\beta})) \ . \tag{80}$$

Formula (80) is usually substantially more accurate than formula (75) for a given n. For example, to get 9 significant digits for $t = 1200$ in Example 6.1, we need $n = 500$. In contrast, the same accuracy is obtained with (80) with only $n = 350$. For smaller t, usually $n = 100$ suffices for (75), while $n = 50$ suffices for (80).

We illustrate the extrapolation refinement with another example.

Example 8.1. As in [1], we now consider the ccdf $G^c(t)$ of the steady-state waiting time in the M/G/1 queue with arrival rate 0.7, mean service time 1.0 and PME service-time distribution with Laplace transform (18). This service-time distribution is interesting because it has a long tail. In particular, the service-time pdf has the asymptotic form $f(t) \sim t^{-3}$ as $t \to \infty$.

The ccdf of interest $G^c(t)$ has Laplace transform $\hat{G}^c(s) = (1 - \hat{g}(s))/s$ for

$$\hat{g}(s) = \frac{0.3}{1 - 0.7[1 - \hat{f}(s)]/s} \tag{81}$$

and $\hat{f}$ in (18). Table 5 compares the ACM algorithm to exact values for both large and small t.

t	exact Fourier-series method	ACM Algorithm 662 $\sigma = 0.1,\ b = 1$
1	0.539934	0.539934
10	0.162295	0.162295
100	0.131553e-1	0.131553e-1
200	0.624080e-2	0.62424e-2
400	0.302786e-2	0.77e-12
800	0.148865e-2	—
1200	0.986383e-3	—

Table 5. A comparison of ACM algorithm 662 with exact values of the steady-state waiting-time ccdf for the M/G/1 queue with long-tail service-time distribution in Example 8.1. Our modified algorithm matches the exact values to all displayed digits.

The exact values in Table 5 are computed by the Fourier-series method [1], [14] using $l = 2$ and $l = 3$ for an accuracy check. The ACM algorithm is done with scaling $\sigma = 0.1$ and $b = 1$. As can be seen from Table 5, the ACM algorithm is excellent when t is not too large, but it starts losing accuracy at $t = 200$ and fails for $t \geq 400$. In contrast, with the scaling in Section 7 the Laguerre algorithm produces accuracy to all displayed digits (and more) for all values of t. Moreover, extrapolation using (80) significantly improves accuracy for a given number of terms.

We close this section by pointing out that the extrapolation can produce errors, rather than improvement, if $\{q_n\}$ does not converge to zero geometrically fast. However, such errors will invariably be detected if accuracy checks are performed. In such cases it is better to use the ϵ-algorithm considered in Section 3. However, in the presence of geometric convergence, the acceleration technique introduced in this section usually is much more effective than the ϵ-algorithm.

9. Summary of the Algorithm

We first describe a simple basic algorithm assuming that the problem is well behaved, i.e., assuming that $\hat{f}(s)$ has all singularities strictly to the left of the imaginary axis, $f(t)$ is square integrable, $|q_n|$ decays geometrically or faster with n, we are not interested in $f(t)$ for large t, and we are not interested in very small (less than 10^{-8}) values of $|f(t)|$. Later we will describe refinements to treat cases when one or more of these assumptions does not hold. We will also show how to check the validity of these assumptions.

Basic Algorithm

Step 1: Compute and store the Laguerre coefficients q_n for $n = 0, 1, 2, \ldots, n_0$ from $Q(z)$ in (5) using the basic LATTICE-POISSON algorithm in Section 1, in particular, with (23) using $r = (0.1)^{4/n}$ yielding roughly errors of the order 10^{-8} or smaller using standard double-precision computation. The truncation point n_0 is the minimum of 100 and the n for which $|q_k| < 10^{-8}$ for all $k \geq n$. (In particular, we can observe when $|q_n| < 10^{-8}$ for several successive n to avoid one chance occurrence. For well-behaved problems, n_0 can often be significantly smaller than 100.)

Step 2: For each desired t, first compute and store the Laguerre functions $l_n(t)$ for $n = 0, 1, \ldots n_0$ using the recursion (32) and then use (2) with truncation point n_0 to compute $f(t)$.

Refinements

Refinement 1: (better accuracy and accuracy check in computing q_n): This is done by replacing Step 1 of the basic algorithm with the enhanced LATTICE-POISSON algorithm with $r = (0.1)^{6/n}$ and $l = 2$ and/or 3. Also, decide the truncation point n_0 based on $|q_n| < 10^{-12}$. This will roughly yield errors of the order 10^{-12} or smaller. Also, if both $l = 2$ and $l = 3$ are used and the results agree, then that provides an accuracy check for q_n since the contours of integration in the two cases are different. If only an accuracy check is needed, then the above comparison can be done with just $l = 1$ and 2.

Refinement 2: (accuracy check in the computation of $f(t)$): Use the (σ, b) scaling algorithm in Section 4. The basic algorithm uses $\sigma = 0$ and $b = 1$. Redoing the computations with slightly perturbed values such as $(\sigma = 0.1, b = 1)$ or $(\sigma = 0, b = 1.1)$ yields an independent accuracy check because the entire computation changes significantly.

What To Do Next?

If the accuracy checks in Refinements 1 and 2 above are both satisfied for all desired values of t, then we are done. If not, then there are problems in the basic method and we have to consider further refinements. To identify which refinement will help, the best thing to do is to look at the decay rate of $|q_n|$ with n. This is easy to do since the values of q_n are already stored as part of the basic algorithm. There are two scenarios:

Scenario 1: $|q_n|$ either does not decay with n or decays too slowly and remains much above 10^{-8} even at $n = 100$.

Scenario 2: $|q_n|$ decays rapidly and is around or less than 10^{-8} by $n = 100$.

Under Scenario 1 we try Refinements 3, 4 and 5 below in turn; under Scenario 2 we try Refinement 6.

Refinement 3: (first refinement to consider under Scenario 1): Increase n_0 to 200 or 400 and see if $|q_{n_0}|$ falls below 10^{-8}. If yes, then perform accuracy checks as in refinements 1 and 2. If the checks are satisfied, then we are done. If any of the above is negative, go to Refinement 4.

Refinement 4: (second refinement to consider under Scenario 1): Assume that the problem is due to the presence of singularities of $\hat{f}(s)$ on the imaginary axis or at infinity. (In some cases the user may know this and come directly to this step.) Redo the algorithm with successive higher values of σ. For example, it is natural to consider geometrically increasing values, such as 0.1, 0.2, 0.4, 0.8, ... etc. See if there exists some critical σ above which $|q_k|$ starts decaying fast. If yes, then use some suitable σ above this critical value and redo the basic algorithm with refinements 1 and 2. If no, then go to refinement 5.

Refinement 5: (third refinement under Scenario 1): Assume that the difficulty is due to a singularity of $\hat{f}(s)$ at $s = -\infty$ as in Section 2. Use the scaling parameter $b = 10$ and the ϵ-algorithm as in Sections 4–6. If this is indeed the difficulty, then it should be possible to obtain high accuracy, as demonstrated in Section 5.

Refinement 6: (Refinement under Scenario 2). In Scenario 2 the decay rate of $|q_n|$ with n is fast and so the algorithm should not have any difficulty for small values of t. So the difficulty here should only be for large values of t and be caused due to large relative errors in computing q_n when it is very small but still contributes significantly in the expression (2) or (27). As explained in Sections 7 and 8, high accuracy is attainable using the scaling in (75). In this setting, improvement can usually be obtained by extrapolation exploiting the geometric rate as in (82).

We conclude by summarizing the controls that can be used with the refined Laguerre algorithm in Table 6.

10. Conclusions

In this paper we have investigated the Laguerre method for numerically inverting Laplace transforms. We have obtained new insight into when the Laguerre method performs well and when it has difficulties. Using the six refinements described in Section 9, it is possible to treat a significantly larger class of problems via the Laguerre method than was possible before.

Given our previous experience with the Fourier-series method, it is natural to compare the Laguerre method to the Fourier-series method. We noted at the outset that the Fourier-series method and Laguerre method proceed in a similar way. However, there are significant differences. First, in order to control the aliasing error in approximating $f(t)$ by $f_p(t)$ in the Fourier-series method, we focus on a single value of t for each calculation. In contrast, the Laguerre method is more likely to yield one function that is a good approximation for a large set of t. However, this capability is a mixed blessing, because if $f(t)$ is badly behaved at just one value of t, then the Laguerre method has difficulties at all values of t. As a simple example, the distribution of the steady-state waiting time in a stable queueing system usually has an atom at the origin, which causes large errors in the Laguerre method unless the atom is explicitly taken out. This can be a nuisance since it is often not easy to identify this atom for non-Poisson arrival processes. In contrast, the atom has negligible effect on the accuracy of the Fourier-series method, so that there is no need to take it out. Moreover, even if $f(t)$ is differentiable everywhere, but its derivative has a discontinuity at the origin, then the Laguerre method has difficulty, whereas the Fourier-series method does not. Nevertheless, the Laguerre method can perform much better for smooth functions when function values are sought for a large number of time points.

Another difficulty with the Laguerre method is that the Laguerre coefficients are not expressed directly in terms of Laplace transform values. Instead, the Laguerre generating function is connected to the Laplace transform via the nonlinear function T in (9). We showed that this causes the Laguerre method to have difficulties for some functions. We have shown that it is possible to resolve many of the difficulties that arise with the Laguerre method, but the remedies make the Laguerre algorithm more complicated. Overall, our experience indicates that the Fourier-series method tends to be more robust, i.e., provides satisfactory accuracy without special tuning for a wider class of functions. However, for well-behaved functions, the Laguerre method can be more efficient, especially when many function values are required.

control	control parameter	default values	source of more information
truncation point	n_0	100 or smallest n with $\|q_n\| < 10^{-8}$	(2) in Section 0
aliasing error in LATTICE-POISSON	r	$(0.1)^{4/ln}$ for q_n	(27) in Section 1
roundoff error in LATTICE-POISSON	l	1	(25) in Section 1
exponential damping scale parameter	σ	0	Section 4
time scale parameter	b	1	Sections 4 and 5
ϵ-algorithm	use	not use	Section 3
scaling of Laguerre generating function $Q(z)$ to achieve low relative error in q_n	use	not use	Section 7
extrapolation exploiting geometric rate of convergence	use	not use	Section 8

Table 6. A summary of the controls that can be used with the refined Laguerre algorithm.

References

1. J. Abate, G. L. Choudhury and W. Whitt, 1994. Waiting-time tail probabilities in queues with long-tail service-time distributions, *Queueing Systems*, 16, 311–338.

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

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

4. J. Abate and W. Whitt, 1992. Numerical inversion of probability generating functions, *Oper. Res. Letters* 12, 245–251.

5. J. Abate and W. Whitt, 1995. Numerical inversion of Laplace transforms of probability distributions, *ORSA J. on Computing* 7, 36–43.

6. J. Abate and W. Whitt, 1996. An operational calculus for probability distributions via Laplace transforms, *Adv. Appl. Prob.*, to appear.

7. M. Abramowitz and I. A. Stegun, 1972. *Handbook of Mathematical Functions*, National Bureau of Standards, Washington, D.C.

8. G. A. Baker, Jr., 1975. *Essentials of Padé Approximants*, Academic, New York.

9. C. M. Bender and S. A. Orszag, 1978. *Advanced Mathematical Methods for Scientists and Engineers*, McGraw-Hill, New York.

10. C. F. Chen, 1966. A new formula for obtaining the inverse Laplace transformation in terms of Laguerre functions, *IEEE Intern. Convention Record* 14, 281–287.

11. G. L. Choudhury, K. K. Leung and W. Whitt, 1995. An algorithm to compute blocking probabilities in multi-rate multi-class multi-resource loss models, *Adv. Appl. Prob.*, to appear.

12. G. L. Choudhury, K. K. Leung and W. Whitt, 1995. An inversion algorithm to compute blocking probabilities in loss networks with state-dependent rates, *IEEE/ACM Trans. Networking*, to appear.

13. G. L. Choudhury and D. M. Lucantoni, 1995. Numerical computation of a large number of moments with applications to asymptotic analysis, *Opns. Res.,* to appear.

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

15. G. L. Choudhury and W. Whitt, 1995. Q^2: a new performance analysis tool exploiting numerical transform inversion, *Proc. Third Int. Workshop on Modeling, Analysis and Simul. of Computer and Telecomm. Systems* (MASCOTS '95), Durham, NC, 411–415.

16. G. L. Choudhury and W. Whitt, 1995. Computing distributions and moments in polling models by numerical transform inversion, *Perf. Eval.*, to appear.

17. K. S. Crump, 1976. Numerical inversion of Laplace transforms using a Fourier-series approximation, *J. ACM* 23, 89–96.

18. G. Darboux, 1878. Mémoire sur l'approximation des fonctions de tres grands nombres, *J. de Mathématiques* 4, 5–56 and 377–416.

19. B. Davies and B. L. Martin, 1979. Numerical inversion of the Laplace transform: A survey and comparison of methods, *J. Comp. Phys.* 33, 1–32.

20. G. Doetsch, 1974. *Introduction to the Theory and Application of the Laplace Transformation*, Springer-Verlag, New York.

21. D. G. Duffy, 1993. On the numerical inversion of Laplace transforms: comparison of three new methods on characteristic problems from applications, *ACM Trans. Math. Software* 19, 333–359.

22. B. S. Garbow, G. Giunta, J. N. Lyness and A. Murli, 1988. Software for an implementation of Weeks' method for the inverse Laplace transform problem, *ACM Trans. Math. Software* 14, 163–170.

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

24. P. Henrici, 1977. *Applied and Computational Complex Analysis*, vol. 2, Wiley, New York.

25. E. Hille, 1962. *Analytic Function Theory*, Vol. 2, Ginn, Boston.

26. W. Kaplan, 1962. *Operational Methods*, Addison, Reading.

27. J. Keilson and W. R. Nunn, 1979. Laguerre transformation as a tool for the numerical solution of integral equations of convolution type, *Appl. Math. Comput.* 5, 313–359.

28. J. Keilson, W. R. Nunn and U. Sumita, 1981. The bilateral Laguerre transform, *Appl. Math. Comput.* 8, 137–174.

29. J. Keilson and U. Sumita, 1982. Waiting time distribution response to traffic surges via the Laguerre transform, *Applied Probability – Computer Science: The Interface*, ed. R. L. Disney and T. J. Ott, Birkhäuser, Boston, 109–130.

30. E. Laguerre, 1876. Sur l'approximation des fonctions, *Bulletin de la Société Mathématique* 5, 78–92.

31. Y. L. Luke, 1969. *The Special Functions and Their Approximations*, vol. 2, Academic, New York.

32. J. N. Lyness and G. Giunta, 1986. A modification of the Weeks method for numerical inversion of the Laplace transform, *Math. Comp.* 47, 313–322.

33. W. Magnus, F. Oberhettinger and R. P. Soni, 1966. *Formulas and Theorems for the Special Functions of Mathematical Physics*, Springer, New York.

34. J. McCully, 1960. The Laguerre transform, *SIAM Review* 2, 185–191.

35. G. Moretti, 1964. *Functions of a Complex Variable*, Prentice Hall, Englewood Cliffs.

36. R. Piessens and M. Branders, 1971. Numerical inversion of the Laplace transform using generalized Laguerre polynomials, *Proc. IEEE* 118, 1517–1522.

37. R. Piessens and R. Huysmans, 1984. Algorithm 619. Automatic numerical inversion of the Laplace transforms, *ACM Trans. Math. Software* 10, 348–353.

38. D. Shanks, 1955. Non-linear transformations of divergent and slowly convergent sequences, *J. Math. and Phys.* 34, 1–42.

39. D. A. Smith and W. F. Ford, 1982. Numerical comparisons of nonlinear convergence accelerators, *Math. Comp.* 38, 481–499.

40. R. A. Spinelli, 1966. Numerical inversion of a Laplace transform, *SIAM J. Numer. Anal.* 3, 636–649.

41. U. Sumita, 1984. The Laguerre transform and a family of functions with nonnegative Laguerre coefficients, *Math. Oper. Res.* 9, 510–521.

42. U. Sumita, 1984. The matrix Laguerre transform, *Appl. Math. Comput.* 15, 1–28.

43. U. Sumita and M. Kijima, 1985. The bivariate Laguerre transform and its applications: numerical exploration of bivariate processes, *Adv. Appl. Prob.* 17, 683–708.

44. U. Sumita and M. Kijima, 1988. Theory and algorithms of the Laguerre transform. part I: theory, *J. Opns. Res. Soc. Japan* 31, 467–494.

45. G. Szegö, 1975. *Orthogonal Polynomials*, 4^{th} ed., American Math. Soc. Colloq. Pub. 23.

46. F. Tricomi, 1935. Transformazione di Laplace e polinami di Laguerre, *R. C. Accad. Nat. dei Lincei* 21, 232–239.

47. B. Van Der Pol and H. Bremmer, 1955. *Operational Calculus*, Cambridge Univ. Press. (Reprinted in 1987 by Chelsea Press, New York).

48. E. E. Ward, 1954. The calculation of transients in dynamical systems, *Proc. Camb. Phil. Soc.* 50, 49–59.

49. H. Weber, 1981. Numerical computation of the Fourier transform using Laguerre functions and the fast Fourier transform, *Numer. Math.* 36, 197–209.

50. W. T. Weeks, 1966. Numerical inversion of Laplace transforms using Laguerre functions, *J. ACM* 13, 419–426.

51. D. V. Widder, 1935. An Application of Laguerre polynomials, *Duke Math. J.* 1, 126–136.

52. H. S. Wilf, 1994. *generatingfunctionology*, second ed., Academic, Boston.

53. J. Wimp, 1981. *Sequence Transformations and Their Applications*, Academic, New York.

54. W. T. Wu and H. R. Ong, 1983. On a functional approximation for inversion of Laplace transforms, *J. Chinese Inst. Eng.* 6, 265–268.

55. P. Wynn, 1956. On a device for computing the $e_m(S_n)$ transformation, *Math. Tables Aids Comput.* 10, 91–96.

56. P. Wynn, 1966. On the convergence and stability of the epsilon algorithm, *SIAM J. Numer. Anal.* 3, 91–122.