SIAM J. SCI. STAT. COMPUT.
Vol. 3, No. 3, September 1982

0196-5204/82/0303-0006 $01.00/0

AN IMPROVED METHOD FOR NUMERICAL INVERSION OF LAPLACE TRANSFORMS*

F. R. DE HOOG†, J. H. KNIGHT† AND A. N. STOKES†

Abstract. An improved procedure for numerical inversion of Laplace transforms is proposed based on accelerating the convergence of the Fourier series obtained from the inversion integral using the trapezoidal rule. When the full complex series is used, at each time-value the epsilon-algorithm computes a (trigonometric) Padé approximation which gives better results than existing acceleration methods. The quotient-difference algorithm is used to compute the coefficients of the corresponding continued fraction, which is evaluated at each time-value, greatly improving efficiency. The convergence of the continued fraction can in turn be accelerated, leading to a further improvement in accuracy.

Key words. Laplace transforms, Fourier series, acceleration, ε-algorithm, continued fractions, quotient-difference algorithm.

1. Introduction. The Laplace transform of a function $f(t)$, $t \geqq 0$, is defined as

$$F(p)=\int_0^\infty \exp(-pt)f(t)\,dt, \tag{1}$$

and the inverse transform is given by

$$f(t)=\frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} \exp(pt)F(p)\,dp, \tag{2}$$

where γ is such that the contour of integration is to the right of any singularities of $F(p)$.

There are many problems for which the Laplace transform of the solution is readily found, but the transform cannot be easily inverted analytically. For such cases a numerical method of inversion must be used. Unfortunately, different numerical methods give the most accurate answers for various classes of functions and there is no "best" method. Recently, Davies and Martin [4] tested a wide variety of methods on a representative set of sixteen transforms with known inverses and concluded that one of the more successful methods was based on accelerating the convergence of a Fourier series.

It therefore seems worthwhile attempting to improve an already quite good method, and this is the aim of the present paper. In the next section, we describe the Fourier series method and the need to accelerate the convergence of the series. Our improved algorithm is given in § 3, and some numerical comparisons are presented in § 4.

2. Fourier series methods. Since f is a real-valued function for real t, three mathematically equivalent forms can be obtained by manipulating the real and imaginary parts of (2). They are

$$f(t)=\frac{2}{\pi}\exp(\gamma t)\int_0^\infty \operatorname{Re}\{F(p)\}\cos(\omega t)\,d\omega \tag{3}$$

$$=-\frac{2}{\pi}\exp(\gamma t)\int_0^\infty \operatorname{Im}\{F(p)\}\sin(\omega t)\,d\omega \tag{4}$$

$$=\frac{1}{\pi}\exp(\gamma t)\int_0^\infty \operatorname{Re}\{F(p)\exp(i\omega t)\}\,d\omega, \tag{5}$$

* Received by the editors April 21, 1980, and in revised form December 7, 1981.

† Division of Mathematics and Statistics, Commonwealth Scientific and Industrial Research Organization, Canberra, ACT 2601, Australia.

where $p=\gamma+i\omega$. If we now discretize, using the trapezoidal rule with step size π/T, we obtain the approximations

$$f_1(t)=\frac{2}{T}\exp(\gamma t)\left[\frac{F(\gamma)}{2}+\sum_{k=1}^{\infty}\operatorname{Re}\left\{F\left(\gamma+\frac{ik\pi}{T}\right)\right\}\cos\left(\frac{k\pi t}{T}\right)\right], \tag{6}$$

$$f_2(t)=-\frac{2}{T}\exp(\gamma t)\sum_{k=1}^{\infty}\operatorname{Im}\left\{F\left(\gamma+\frac{ik\pi}{T}\right)\right\}\sin\left(\frac{k\pi t}{T}\right), \tag{7}$$

$$f_3(t)=\frac{1}{T}\exp(\gamma t)\left[\frac{F(\gamma)}{2}+\sum_{k=1}^{\infty}\operatorname{Re}\left\{F\left(\gamma+\frac{ik\pi}{T}\right)\exp\left(\frac{ik\pi t}{T}\right)\right\}\right] \tag{8}$$

with various derivations; (6)–(8) are the basis of Fourier series methods examined and used by Dubner and Abate [5], Cooley, Lewis and Welch [2], Silverberg [11], Durbin [6], Crump [3] and others.

Although (3) and (4) are mathematically equivalent, the discretized forms (6)–(8) are not. In fact, it can be shown (see Durbin [6]) that for $0\leqq t\leqq 2T$

$$f_1(t)=f(t)+\sum_{k=1}^{\infty}\exp(-2\gamma kT)[f(2kT+t)+\exp(2\gamma t)f(2kT-t)], \tag{9}$$

$$f_2(t)=f(t)+\sum_{k=1}^{\infty}\exp(-2\gamma kT)[f(2kT+t)-\exp(2\gamma t)f(2kT-t)], \tag{10}$$

$$f_3(t)=f(t)+\sum_{k=1}^{\infty}\exp(-2\gamma kT)f(2kT+t). \tag{11}$$

Thus, if γT is large, the discretization error (incurred when the integral is approximated by the Fourier series) is small for f_1 and f_2 when $0\leqq t\leqq T$ and for f_3 when $0\leqq t\leqq 2T$. Furthermore, (9)–(10) indicate that f_3 (which is the average of f_1 and f_2) may be the most useful approximation in practice because it does not contain the exponentially increasing term $\exp(2\gamma t)$ in the discretization error. However, the choice of the most useful discretization depends heavily on how the sums of the infinite series in (6)–(8) are calculated.

To illustrate this, let us consider the evaluation of $f_3(t)$ on an equally spaced partition $t_j=2jT/J$, $j=0,1,\cdots,J-1$. A formal manipulation of $f_3(t_j)$ yields

$$T\exp(\gamma t_j)f_3(t_j)=\operatorname{Re}\left\{\sum_{k=0}^{J-1}a_k\exp\left(\frac{2\pi ikj}{J}\right)\right\}, \tag{12}$$

where

$$a_0=\frac{F(\gamma)}{2}+\sum_{l=1}^{\infty}F\left(\gamma+\frac{ilJ\pi}{T}\right) \tag{13}$$

and

$$a_k=\sum_{l=0}^{\infty}F\left(\gamma+\frac{i(k+lJ)\pi}{T}\right),\qquad k=1,\cdots,J-1.$$

If J is large and a_k, $k=0,\cdots,J-1$, are known, (12) can be evaluated efficiently using a fast Fourier transform. Unfortunately, the formal manipulation used to derive (12) is not valid unless the series defining f_3 converges uniformly. For example, if $F(p)=1/p$, the sums in (13) will diverge.

On the other hand, if f_1 is evaluated on the grid $s_j = jT/J$, $j = 0, \cdots, J$, we obtain

$$\frac{T}{2}\exp(\gamma s_j) f_1(s_j) = \sum_{k=0}^{J} b_k \cos\left(\frac{\pi k j}{J}\right), \qquad j = 0, \cdots, J \tag{14}$$

where

$$b_0 = \frac{F(\gamma)}{2} + \sum_{l=1}^{\infty} \operatorname{Re}\left\{F\left(\gamma + \frac{2\pi i l J}{T}\right)\right\}$$

for $k = 1, \cdots, J-1$,

$$b_k = \sum_{l=0}^{\infty} \operatorname{Re}\left\{F\left(\gamma + \frac{i(2lJ+k)\pi}{T}\right) + F\left(\gamma + \frac{i(2lJ+J-k)\pi}{T}\right)\right\} \tag{15}$$

and

$$b_J = \sum_{l=0}^{\infty} \operatorname{Re}\left\{F\left(\gamma + \frac{i(2l+1)J\pi}{T}\right)\right\}.$$

Cooley, Lewis and Welch [2] have noted that (14) can also be evaluated using a fast cosine transform and that the sums defining $b_0, \cdots, b_j$ will usually converge. They, therefore, choose (15) and (16) as the basis of their inversion algorithm even though a comparison of (9) and (11) shows that the discretization error of f_3 will usually be smaller than that of f_1. A method for choosing an appropriate value of γ (assuming that the b_k, $k = 0, \cdots, J$ are known to machine precision) is also given by these authors. However, accurate calculation of the b_k is by no means trivial. For example, if $F(p) = 1/p$, then

$$b_0 = \frac{1}{2\gamma} + \gamma T^2 \sum_{l=1}^{\infty} \frac{1}{\gamma^2 T^2 + 4\pi^2 l^2 J^2},$$

which converges very slowly. In general, a very large number of evaluations of $F(p)$ may be required for an accurate inversion, and it is clear that this part of the calculation can be the major source of rounding error and require the most computer time, even when $F(p)$ has a simple form.

Of course, slow convergence, with its attendant problems of loss of accuracy and excessive computation time, can be expected for any algorithm which evaluates (6)–(8) directly. If a Fourier scheme is to be useful, it would seem to be necessary to accelerate the convergence of the appropriate infinite series. Durbin [6] reported that he had tried various convergence acceleration methods, apparently without major improvement. Crump [3] used the epsilon-algorithm of Wynn [12] to accelerate the convergence of the sum in (5) with good results.

To accelerate the convergence of a sequence of partial sums using the epsilon-algorithm

$$S_0 = \frac{F(\gamma)}{2}, \qquad S_k = S_{k-1} + \operatorname{Re}\left\{F\left(\gamma + \frac{ik\pi}{T}\right)\exp\left(\frac{ik\pi t}{T}\right)\right\}, \qquad k = 1, \cdots, 2M,$$

we define $\varepsilon_{-1}^{(m)} = 0$, $\varepsilon_0^{(m)} = S_m$, $m = 0, 1, \cdots, 2M$, and then put

$$\varepsilon_{p+1}^{(m)} = \varepsilon_{p-1}^{(m+1)} + [\varepsilon_p^{(m+1)} - \varepsilon_p^{(m)}]^{-1}. \tag{16}$$

The sequence $\varepsilon_0^{(0)}, \varepsilon_2^{(0)}, \varepsilon_4^{(0)}, \cdots, \varepsilon_{2M}^{(0)}$ gives successive approximations to the sum of the series. Crump's accelerated estimate of $f(t)$ we denote by

$$\hat{f}_1(t, \gamma, T, M) = \frac{1}{T} \exp(\gamma t) \varepsilon_{2M}^{(0)}.$$

For comparison, we denote the unaccelerated estimate by

$$\hat{f}_0(t, \gamma, T, M) = \frac{1}{T} \exp(\gamma t)\left[\frac{F(\gamma)}{2} + \sum_{k=1}^{2M} \operatorname{Re}\left\{F\left(\gamma + \frac{ik\pi}{T}\right) \exp\left(\frac{ik\pi t}{T}\right)\right\}\right].$$

3. Improved acceleration. A convergence acceleration technique can be applied to any arbitrary sequence of numbers, with uncertain results, but most such techniques assume that the sequence has some particular structure, and they work best for sequences close to the assumed form. As emphasized by Wynn [13], applying the epsilon-algorithm to the partial sums of a power series in a variable z is equivalent to constructing successive rational approximations to the power series. These rational approximations have special properties being particular Padé approximations to the power series. (For their properties see Baker [1] or Henrici [8].) The sequence $\{\varepsilon_{2m}^{(0)}\}$, $m = 0, 1, \cdots, M$, gives diagonal elements of the Padé table

$$\varepsilon_{2m}^{(0)} = \left[\sum_{n=0}^{m} b_n z^n\right] \Big/ \left[\sum_{n=0}^{m} c_n z^n\right], \qquad c_0 = 1$$

with

$$\sum_{n=0}^{2m} a_n z^n - \varepsilon_{2m}^{(0)} = O(z^{2m+1}).$$

The Padé approximations are often better than the original power series at representing a function and valid in a larger domain. Accordingly, the greater improvement in convergence can be expected when the algorithm is applied to a sequence which happens to be the partial sums of some power series. For example, given a cosine series $\sum a_n \cos(n\theta)$ to accelerate, Wynn [13] appends the conjugate sine series $\sum a_n i \sin(n\theta)$, applies the epsilon-algorithm to the complex power series in the variable $z = \exp(i\theta)$ and takes the real part of the result. Because the algorithm is nonlinear, this is not the same as accelerating the cosine series directly.

For our problem of Laplace transform inversion, the natural way to obtain a power series is to retain the original complex form

$$g(t) = \frac{F(\gamma)}{2} + \sum_{k=1}^{\infty} F\left(\gamma + \frac{ik\pi}{T}\right) \exp\left(\frac{ik\pi t}{T}\right) = \sum_{k=0}^{\infty} a_k z^k \tag{17}$$

with $a_0 = \frac{1}{2}F(\gamma)$, $a_k = F(\gamma + ik\pi/T)$, $k = 1, 2, \cdots$ and $z = \exp(i\pi t/T)$.

We propose to apply the epsilon-algorithm to the partial sums of the series (17) to give an improved estimate

$$\hat{f}_2(t, \gamma, T, M) = \frac{1}{T} \exp(\gamma t) \operatorname{Re}\{\varepsilon_{2M}^{(0)}\}. \tag{18}$$

If the epsilon table is written out in the form

$$\begin{array}{cccc}
 & \varepsilon_0^{(0)} & & \\
\varepsilon_{-1}^{(1)} & & \varepsilon_1^{(0)} & \\
 & \varepsilon_0^{(1)} & & \varepsilon_2^{(0)} \\
\varepsilon_{-1}^{(2)} & & \varepsilon_1^{(1)} & \\
 & \varepsilon_0^{(2)} & & \\
\varepsilon_{-1}^{(3)} & & &
\end{array}$$

for each J, a new diagonal $\varepsilon_j^{(J-j)}$, $j=-1, 0, \cdots, J$, can be calculated using the "rhombus rule" in the form

$$\varepsilon_j^{(J-j)}=\varepsilon_{j-2}^{(J+1-j)}+[\varepsilon_{j-1}^{(J+1-j)}-\varepsilon_{j-1}^{(J-j)}]^{-1}, \qquad j=1,\cdots,J \tag{19}$$

with the initial values

$$\varepsilon_{-1}^{(J+1)}=0, \qquad \varepsilon_0^{(J)}=\sum_{k=0}^{J} a_k z^k.$$

Every second such diagonal yields an improved estimate $\varepsilon_{2m}^{(0)}$, $m=1, 2, \cdots, M$.

When applied to the partial sums of a power series in this way, it can be seen that the procedure calculates $\varepsilon_1^{(J-1)}=[a_J z^J]^{-1}$, for example, by subtracting successive partial sums to retrieve the Jth term of the power series. Table entries can be numerically large and the rhombus rule often involves the subtraction of two large quantities to find a relatively small difference. It is not surprising that calculation of table elements is numerically unstable and can be seriously affected by roundoff error for large J. For each new value of z, all the entries in the table must be recalculated, and $M(2M-1)$ entries are needed to give the estimate $\varepsilon_{2M}^{(0)}$ although no more than two diagonals need to be stored at any stage.

Given that our proposed acceleration procedure is in fact computing a diagonal Padé approximation corresponding to the power series (17), it is profitable to investigate other methods which may be more efficient. Wynn [13] suggests that, while it is more economical to use the epsilon-algorithm for an exploratory survey with a limited number of sample values of z, if results are required for many values it is better to use a method such as the quotient-difference algorithm of Rutishauser [10]. This makes the rational approximation available in the form of a continued fraction for substitution of different values of z. Therefore, for the inversion of Laplace transforms, we propose to use the quotient-difference algorithm to calculate a (trigonometric) rational approximation to the series (17) in the form of a continued fraction, and to evaluate it by recursion at any value of the time.

In our outline of the quotient-difference algorithm, we follow Henrici [7], [8], [9]. Given the power series (17), we wish to calculate the corresponding continued fraction

$$v(z)=d_0/(1+d_1 z/(1+d_2 z/(1+\cdots)))$$

with the same formal power series development. Given

$$u(z,M)=\sum_{k=0}^{2M} a_k z^k, \qquad v(z,M)=d_0/(1+d_1 z/(1+\cdots+d_{2M}z)),$$

$$u(z,M)-v(z,M)=O(z^{2M+1})$$

so $v(z, M)$ is the same diagonal Padé approximation as $\varepsilon_{2M}^{(0)}$. The coefficients d_k, $k=0, 1, \cdots, 2M$ can be calculated using the quotient-difference algorithm as follows. We set $e_0^{(i)}=0$ for $i=0, \cdots, 2M$ and $q_1^{(i)}=a_{i+1}/a_i$ for $i=0, \cdots, 2M-1$. Then successive columns of the array are formed according to the rules for

$$(20)\qquad r=1, \cdots, M, \qquad e_r^{(i)}=q_r^{(i+1)}-q_r^{(i)}+e_{r-1}^{(i+1)}, \qquad i=0, \cdots, 2M-2r,$$

for

$$r=2, \cdots, M, \qquad q_r^{(i)}=q_{r-1}^{(i+1)}e_{r-1}^{(i+1)}/e_{r-1}^{(i)}, \qquad i=0, \cdots, 2M-2r-1.$$

The array is written out in the form

$$\begin{array}{ccccc}
 & q_1^{(0)} & & & \\
e_0^{(1)} & & e_1^{(0)} & & \\
 & q_1^{(1)} & & q_2^{(0)} & \\
e_0^{(2)} & & e_1^{(1)} & & e_2^{(0)}, \\
 & q_1^{(2)} & & q_2^{(1)} & \\
e_0^{(3)} & & e_1^{(2)} & & \\
 & q_1^{(3)} & & & \\
e_0^{(4)} & & & &
\end{array}$$

and it can be seen that successive diagonals can be built up in the same way the epsilon table is constructed using the relations (20).

The continued fraction coefficients d_k, $k=0, \cdots, 2M$ are given by

$$d_0=a_0, \quad d_{2m-1}=-q_m^{(0)}, \quad d_{2m}=-e_m^{(0)}, \qquad m=1, \cdots, M.$$

Calculating these using the $q-d$ algorithm involves about the same effort as calculating the epsilon table but does not have to be redone for a new value of z. For any z the successive convergents of the continued fraction can be evaluated using the recurrence relations

$$(21)\qquad \begin{aligned} A_n &= A_{n-1}+d_n z A_{n-2}, \\ B_n &= B_{n-1}+d_n z B_{n-2}, \end{aligned} \qquad n=1, \cdots, 2M$$

with the initial values $A_{-1}=0$, $B_{-1}=1$, $A_0=d_0$, $B_0=1$, then $v(z, M)=A_{2M}/B_{2M}$.

To apply the procedure to the series (17), as before we put $z=\exp(i\pi t/T)$, $a_0=\frac{1}{2}F(\gamma)$, $a_k=F(\gamma+ik\pi/T)$, $k=1, \cdots, 2M$ and the estimate of $f(t)$ is

$$(22)\qquad \hat{f}_3(t, \gamma, T, M)=\frac{1}{T}\exp(\gamma t)\,\mathrm{Re}\left\{\frac{A_{2M}}{B_{2M}}\right\}.$$

If the numerical calculations were exact, $\hat{f}_2$ and $\hat{f}_3$ would be identical since the underlying rational approximations are the same. Calculating the elements of the $q-d$ array is also numerically unstable, but it is to be expected that roundoff error can affect $\hat{f}_2$ and $\hat{f}_3$ in different ways.

The continued fraction has been obtained as the result of applying an acceleration procedure to the power series (17), and now we can consider applying an acceleration procedure to the continued fraction itself. Writing $v(z)$ as a terminating fraction with remainder,

$$v(z)=d_0(1+d_1z/(1+\cdots+d_nz/(1+r_{n+1})\cdots)),$$

where $r_{n+1}(z)$ is the remainder. In the usual method of evaluation, the nth convergent A_n/B_n is obtained with $r_{n+1}(z)$ taken as zero, but better estimates of the remainder give more accurate approximations for $v(z)$. The simplest improvement is achieved by assuming that $d_{n+m}=d_{n+1}$ for all positive m, which leads to an estimate of the remainder satisfying

$$r_{n+1}=d_{n+1}z/(1+r_{n+1}).$$

However, in many continued fractions, the coefficients form a pattern which repeats in pairs so a better and more general assumption is that for all nonnegative m

$$d_{n+2m}=d_n \quad \text{and} \quad d_{n+2m+1}=d_{n+1}.$$

This leads to an estimate $R_{n+1}(z)$ of the remainder satisfying

$$R_{n+1}=d_{n+1}z/(1+d_nz/(1+R_{n+1}))$$

or

$$R_{n+1}^2+[1+(d_n-d_{n+1})z]R_{n+1}-d_{n+1}z=0.$$

For convergence, we want the root of smaller magnitude which is

$$R_{n+1}(z)=-h_{n+1}[1-(1+d_{n+1}z/h_{n+1}^2)^{1/2}],$$

where h_{n+1} denotes $\frac{1}{2}[1+(d_n-d_{n+1})z]$, and the complex square root has argument $\leqq\pi/2$.

This further form of acceleration can be done in conjunction with the iterative method of forward evaluation used above, if on the last evaluation of the recurrence relations $d_{2M}z$ is replaced by $R_{2M}(z)$,

$$(23)\qquad R_{2M}(z)=-h_{2M}[1-(1+d_{2M}z/h_{2M}^2)^{1/2}], \qquad h_{2M}=\tfrac{1}{2}[1+(d_{2M-1}-d_{2M})z]$$

giving

$$(24)\qquad A'_{2M}=A_{2M-1}+R_{2M}A_{2M-2}, \qquad B'_{2M}=B_{2M-1}+R_{2M}B_{2M-2}.$$

A doubly accelerated estimate for $f(t)$ is given by

$$(25)\qquad \hat{f}_4(t,\gamma,T,M)=\frac{1}{T}\exp(\gamma t)\,\mathrm{Re}\left\{\frac{A'_{2M}}{B'_{2M}}\right\}.$$

4. Numerical results. The three main sources of error in approximating $f(t)$ by the sum of a Fourier series instead of by the integral (5) are (i) discretization error, (ii) truncation error, caused by taking only a finite sum of N terms, and (iii) roundoff error. The discretization error is governed by the values of the parameters γ and T and the roundoff error by the properties of the machine used. It is effectively the truncation error that we are trying to improve, so for a given transform we will use identical values of γ and T for all methods and compute in double precision where necessary to prevent significant roundoff error. Of course, for fixed word length, stability to roundoff error is an important attribute on which we will comment later.

Numerical results were obtained on a CDC Cyber 76 computer with about 14 decimal digits in single precision and about 28 in double precision. Two transforms with known inverses were used to evaluate the improvements to the accelerated

TABLE 1
Errors in estimates of $g_1(t)$, $T=7.5$, $\gamma=-0.5+0.4\ \ln\,(10.0)$.

estimate: precision:	$\hat{f}_1(t, \gamma, T, 9)$ double	$\hat{f}_2(t, \gamma, T, 9)$ double	$\hat{f}_3(t, \gamma, T, 9)$ single	$\hat{f}_4(t, \gamma, T, 9)$ single	$\hat{f}_4(t, \gamma, T, 14)$ single
t					
0.0	7.2×10^{-3}	8.2×10^{-3}	8.2×10^{-3}	5.9×10^{-3}	1.5×10^{-3}
0.5	-1.8×10^{-3}	2.9×10^{-5}	2.9×10^{-5}	4.2×10^{-6}	6.8×10^{-7}
1.0	3.8×10^{-4}	1.5×10^{-6}	1.5×10^{-6}	7.8×10^{-7}	6.7×10^{-7}
2.0	4.5×10^{-5}	3.5×10^{-7}	3.5×10^{-7}	3.5×10^{-7}	3.5×10^{-7}
3.0	8.9×10^{-6}	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}	3.1×10^{-8}
4.0	-1.6×10^{-6}	$-1.1\times10^{\ 7}$	-1.1×10^{-7}	-1.1×10^{-7}	-1.1×10^{-7}
5.0	-6.2×10^{-8}	-9.5×10^{-8}	-9.5×10^{-8}	-9.5×10^{-8}	-9.5×10^{-8}
6.0	-3.6×10^{-8}	-3.5×10^{-8}	-3.5×10^{-8}	-3.5×10^{-8}	-3.5×10^{-8}
7.0	7.0×10^{-9}	7.0×10^{-9}	7.0×10^{-9}	7.0×10^{-9}	7.0×10^{-9}
8.0	1.9×10^{-8}	1.9×10^{-8}	1.9×10^{-8}	1.9×10^{-8}	1.9×10^{-8}
9.0	1.2×10^{-8}	1.2×10^{-8}	$1.2\times10^{\ 8}$	1.2×10^{-8}	1.2×10^{-8}
10.0	8.8×10^{-9}	2.6×10^{-9}	2.6×10^{-9}	2.6×10^{-9}	2.6×10^{-9}
11.0	-6.5×10^{-9}	-2.4×10^{-9}	-2.4×10^{-9}	-2.4×10^{-9}	-2.4×10^{-9}
12.0	-1.9×10^{-5}	-3.8×10^{-9}	-3.8×10^{-9}	-2.9×10^{-9}	-2.8×10^{-9}
13.0	2.1×10^{-3}	-4.5×10^{-8}	-4.5×10^{-8}	-5.9×10^{-9}	-1.3×10^{-9}
13.5	2.9×10^{-2}	4.5×10^{-7}	4.5×10^{-7}	4.7×10^{-8}	-5.9×10^{-10}
14.0	-1.8×10^{-1}	-8.1×10^{-7}	-8.1×10^{-7}	-2.9×10^{-7}	-3.5×10^{-11}
14.5	-3.0×10^{-1}	-1.1×10^{-3}	-1.1×10^{-3}	$-1.4\times10^{\ 4}$	-2.7×10^{-8}
15.0	4.0	4.5	4.5	3.3	8.5×10^{-1}

Fourier series method of Crump [3]. The first is $G_1(p)=(p^2+p+1)^{-1}$ which is the transform of the damped sinusoid $g_1(t)=(2/\sqrt{3})\exp(-t/2)\sin(t\sqrt{3}/2)$ used as an example by Dubner and Abate [5] and by Crump. Most numerical methods can be expected to perform well on this transform of a continuous function. The other is $G_2(p)=1/p$ the transform of the heaviside unit function $g_2(t)=H(t)$ which is zero for negative t, unity for positive t and takes the value $\frac{1}{2}$ for t zero. The jump discontinuity at zero makes this a difficult example for all Fourier series inversion methods. In practice, if the location and magnitude of a discontinuity are known, it can often be treated by analytical methods.

To invert the transform $G_1(p)$, the parameter values of Crump were used, namely $T=7.5$, $N=2M+1=19$ and $\gamma=0.5+0.4\ \ln\,(10.0)\sim0.421$. Double precision-arithmetic was used if the single precision calculations were found to be affected by roundoff. The Crump estimate $\hat{f}_1$ was compared with our improved estimates $\hat{f}_2$ obtained with the complex epsilon-algorithm, $\hat{f}_3$ obtained with the quotient-difference algorithm and $\hat{f}_4$ obtained with accelerated evaluation of the continued fraction. The numerical differences between these and the true values of $g_1(t)$ for the four methods are shown in Table 1 for values of the time up to 15.0. Single precision results are given for $\hat{f}_3$ and $\hat{f}_4$; as to the accuracy shown, the errors were unchanged for these estimates by computing in double precision.

It can be seen that between about 5.0 and 10.0, the truncation error is negligible for all four estimates, the remaining error being due to discretization and not decreased by further increasing M. The convergence of the Fourier series may be affected by a Gibbs phenomenon near $t=0$ and $t=2T=15.0$, and near these points our methods are much more accurate than that of Crump. The single precision estimate $\hat{f}_3$ gives identical errors to the double precision estimate $\hat{f}_2$ showing that the continued fraction procedure is less susceptible to roundoff error as well as computationally much more

TABLE 2
Errors in estimates of $g_2(t)$, $\gamma = 1.0$, $T = 12.0$.

estimate: precision:	$\hat{f}_0(t, \gamma, T, 500)$ single	$\hat{f}_1(t, \gamma, T, 17)$ double	$\hat{f}_2(t, \gamma, T, 17)$ double	$\hat{f}_3(t, \gamma, T, 17)$ single	$\hat{f}_4(t, \gamma, T, 17)$ single
t					
0.00	-1.22×10^{-3}	-3.57×10^{-3}	-6.84×10^{-2}	-6.84×10^{-2}	8.25×10^{-2}
0.05	-2.27×10^{-2}	-3.86×10^{-1}	1.09×10^{-4}	1.09×10^{-4}	5.78×10^{-6}
0.10	-6.94×10^{-3}	2.20×10^{-3}	-7.43×10^{-4}	-7.43×10^{-4}	-4.17×10^{-5}
0.15	-1.82×10^{-5}	4.58×10^{-2}	3.04×10^{-4}	3.04×10^{-4}	1.07×10^{-5}
0.20	3.78×10^{-3}	1.16×10^{-1}	-5.99×10^{-6}	-5.99×10^{-6}	3.97×10^{-7}
0.30	5.47×10^{-3}	-2.73×10^{-3}	-1.60×10^{-6}	-1.60×10^{-6}	-1.01×10^{-7}
0.40	2.08×10^{-3}	5.08×10^{-3}	1.02×10^{-6}	1.02×10^{-6}	2.49×10^{-8}
0.50	-2.22×10^{-3}	9.98×10^{-3}	-2.44×10^{-7}	-2.44×10^{-7}	-2.98×10^{-9}
0.75	3.32×10^{-4}	-1.66×10^{-3}	3.20×10^{-9}	3.20×10^{-9}	1.41×10^{-10}
1.00	1.27×10^{-3}	1.10×10^{-4}	2.36×10^{-10}	2.36×10^{-10}	3.56×10^{-11}
1.50	3.58×10^{-3}	-1.97×10^{-5}	4.16×10^{-11}	4.16×10^{-11}	3.77×10^{-11}
2.00	3.22×10^{-3}	-4.00×10^{-6}	3.78×10^{-11}	3.78×10^{-11}	3.78×10^{-11}
4.00	-5.13×10^{-5}	-2.37×10^{-8}	3.78×10^{-11}	3.78×10^{-11}	3.78×10^{-11}
6.00	-6.40×10^{-2}	1.63×10^{-11}	3.78×10^{-11}	3.78×10^{-11}	3.78×10^{-11}

efficient. The accelerated continued fraction estimate $\hat{f}_4(t, \gamma, T, 9)$ yields a further increase in accuracy near zero and 15.0. Also, $N = 2M + 1 = 29$ terms were used to calculate the estimate $\hat{f}_4(t, \gamma, T, 14)$ and a further improvement was obtained.

A similar comparison was made for the transform $G_2(p)$ of the unit step function using the parameter values $\gamma = 1.0$ and $T = 12.0$ and various values of M. The first column of Table 2 shows the errors in the simple sum $\hat{f}_0(t, \gamma, T, 500)$ using $N = 2M + 1 = 1001$ terms in the sum. The second column shows the errors in the Crump estimate $\hat{f}_1(t, \gamma, T, 17)$ using only $N = 2M + 1 = 35$ terms clearly showing the great improvement resulting from accelerating the convergence. For the same value of M, the errors given by our improved acceleration methods are shown in the remaining columns of Table 2, and it can be seen that for this discontinuous function $g_2(t)$, the improvement over the Crump method is more dramatic. Right at the discontinuity, our methods are no better, but the improvement increases rapidly for increasing t. As in the first example, the doubly accelerated continued fraction estimate $\hat{f}_4$ gives the best results. Calculations (results not shown) were also done with $N = 2M + 1 = 41$ terms. However, owing to the inherent instability of the quotient-difference algorithm, inaccuracies in the single precision calculation of the coefficients d_n became apparent with $M = 20$. These did not greatly affect the final estimate $\hat{f}_3$ but affected the estimates $R_{2M}(z)$ of the remainder, causing $\hat{f}_4$ to be less of an improvement over $\hat{f}_3$ for this value of M.

5. Conclusions. The numerical results in the previous section show that for the accelerated Fourier series methods using the full complex series to calculate a trigonometric rational approximation with either the epsilon or the quotient-difference algorithm results in a significant improvement in accuracy over the method of Crump [3]. Using the $q - d$ algorithm to explicitly calculate the coefficients of the corresponding continued fraction is highly efficient when the inverse is required at many time-values. Evaluating the continued fraction is more stable than applying the epsilon-algorithm to the partial sums and can itself be accelerated to give a further improvement in accuracy.

Acknowledgments. The authors would like to thank Drs. B. Davies and B. L. Martin for a preprint of their survey article and Dr. Davies and Mr. J. E. Drummond for supplying other references.

REFERENCES

[1] G. A. BAKER, *Essentials of Padé Approximants*, Academic Press, New York, 1975.

[2] J. W. COOLEY, P. A. W. LEWIS AND P. D. WELCH, *The fast Fourier transform algorithm: Programming considerations in the calculation of sine, cosine and Laplace transforms*, J. Sound Vib., 12 (1970), pp. 315–337.

[3] K. S. CRUMP, *Numerical inversion of Laplace transforms using a Fourier series approximation*, J. Assoc. Comp. Mach., 23 (1976), pp. 89–96.

[4] B. DAVIES AND B. L. MARTIN, *Numerical inversion of Laplace transforms: A critical evaluation and review of methods*, J. Comp. Phys., 33 (1979), pp. 1–32.

[5] H. DUBNER AND J. ABATE, *Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform*, J. Assoc. Comp. Mach., 15 (1968), pp. 115–123.

[6] F. DURBIN, *Numerical inversion of Laplace transforms: An efficient improvement to Dubner and Abate's method*, Comput. J., 17 (1974), pp. 371–376.

[7] P. HENRICI, *Quotient-difference algorithms*, Mathematical Methods for Digital Computers, vol. 2, A. Ralston and H. S. Wilf, eds., John Wiley, New York, pp. 37–62, 1967.

[8] ———, *Applied and Computational Complex Analysis*, vol. 1, John Wiley, New York, 1974.

[9] ———, *Applied and Computational Complex Analysis*, vol. 2, John Wiley, New York, 1977.

[10] H. RUTISHAUSER, *Der Quotienten-Differenzen-Algorithmus*, Birkhauser Verlag, Basel, 1957.

[11] M. SILVERBERG, *Improving the efficiency of Laplace-transform inversion for network analysis*, Electronics Letts., 6 (1970), pp. 105–106.

[12] P. WYNN, *On a device for computing the $e_m(S_n)$ transformation*, MTAC, 10 (1956), pp. 91–96.

[13] ———, *Transformations to accelerate the convergence of Fourier series*, Blanch Anniversary Volume, B. Mond, ed., U.S. Air Force, Washington, DC, 1967, pp. 339–379.