Noname manuscript No.
(will be inserted by the editor)

Fully Numerical Laplace Transform Methods

J.A.C. Weideman · Bengt Fornberg

the date of receipt and acceptance should be inserted later

Abstract The role of the Laplace transform in scientific computing has been predominantly that of a semi-numerical tool. That is, typically only the inverse transform is computed numerically, with all steps leading up to that executed by analytical manipulations or table look-up. Here we consider fully numerical methods, where both forward and inverse transforms are computed numerically. Because the computation of the inverse transform is one of the most well-studied problems in scientific computing, this paper focus mainly on the forward transform. Existing methods for computing the forward transform based on exponential sums are considered along with a new method based on the formulas of Weeks. Numerical examples include a nonlinear integral equation of convolution type, a fractional ordinary differential equation, and a partial differential equation with an inhomogeneous boundary condition.

Keywords Laplace transform, Weeks method, Padé approximation, exponential sums

J.A.C. Weideman
Department of Mathematical Sciences
Stellenbosch University
Stellenbosch 7600
South Africa
E-mail: weideman@sun.ac.za

Bengt Fornberg
Department of Applied Mathematics
University of Colorado
Boulder CO 80309
USA
E-mail: fornberg@colorado.edu

1 Introduction

Let $f(t)$ be defined on $[0,\infty)$. Its forward Laplace transform is defined by

$$F(z) = \int_0^\infty e^{-zt} f(t)\, dt, \tag{1}$$

and the Bromwich contour integral for the inverse is defined by

$$f(t) = \frac{1}{2\pi i} \int_{\sigma-i\infty}^{\sigma+i\infty} e^{zt} F(z)\, dz. \tag{2}$$

The assumption is that both integrals are well-defined. This means, among other things, that $\sigma > \sigma_0$, with σ_0 the convergence abscissa of the transform. That is, all singularities of $F(z)$ are located in the half-plane $\mathrm{Re}\, z < \sigma_0$.

When used to solve differential or integral equations these formulas are typically applied as follows: The forward transform of all terms in the equation is taken, thereby reducing the problem to an algebraic equation in the variable z. If this can be solved, an explicit expression for the inverse is obtained which then needs to be inverted. More often than not, however, this expression is unwieldy and defies analytical inversion, table look-up, and symbolic software. At this point numerical methods are used to compute the inverse for specified values of t. When used in this manner, the Laplace transform can be viewed as a semi-numerical tool, as only the last step involves numerical computation.

In this paper we consider the use of the Laplace transform as a fully numerical tool, i.e., with both the forward and inverse formulas computed numerically. The computation of the inverse transform has attracted an enormous body of literature. (The bibliography of [6] has no fewer than 267 entries.) For this reason we shall just skim the details of some of the most well-known numerical inversion strategies, namely methods based on Laguerre expansion, contour deformation, or rational approximation. Instead, the main focus here will be on the computation of the forward transform.

If (1) cannot be computed by analytical methods or table look-up, then one option for evaluating it is by a quadrature rule appropriate for the half-line. When used in combination with one of the inversion methods cited above, however, it becomes necessary to compute it for z in various regions of the complex plane. This is usually not a problem if $\mathrm{Re}\, z > \sigma_0$ but in several good inversion methods (particularly those based on contour deformation of (2)) the integral (1) will have to be computed for values $\mathrm{Re}\, z < \sigma_0$, i.e., outside the domain of convergence of the integral. This amounts to a form of analytic continuation, which can pose a challenge to numerical computations. Even inversion methods that stay within the domain of convergence can face a challenge if $\mathrm{Im}\, z$ gets large (the well-known Dubner & Abate and Weeks methods are both in this class [12,29]). The problem here is the oscillatory nature of the integrand of (1).

In this paper we consider two methods for the computation of the forward transform (1), one new and and one old (but improved here). The new method

proposed here is a reversal of the Weeks method for the inversion problem. The old method is the use of exponential sums, already proposed two decades ago [24]. In that paper the use of de Prony's method was suggested (among others). This is updated here with more recent and more stable methods for computing the coefficients in an exponential sum.

The outline of the paper is as follows: In section 2 we review some of the more popular inversion methods, as well as the two methods for the forward problem mentioned in the previous paragraph. Three model problems, ranging from integral equations and fractional differential equations to a partial differential equation are solved in section 3. The performance of the new Weeks method for the forward transform is assessed in this section. In section 4 this method is compared to the existing but newly updated method based on exponential sums. Our findings are summarized in section 5.

2 The Methods

Because of the vast number of methods that have been proposed for the inverse Laplace transform in particular, we limit ourselves here to just a small selection. The first group is methods based on Laguerre functions, which have become associated with the name of Weeks [29] (but who built on earlier work of Tricomi, Widder, and others). We reverse Weeks' inversion method here to compute the forward transform. The second set of methods (for the inverse transform only) is based on path deformation of the contour integral (2). These methods have become associated with the name of Talbot [26] (but Butcher [4] seems to have proposed the basic idea two decades earlier). The third set of methods are based on exponential sum expansions, maybe less familiar for the inversion problem compared to the above two methods, but more familiar for computing the forward transform.

2.1 Laguerre expansions

These methods are based on finite truncations of the following two series [15, 20, 30]. (We follow the notation of the latter reference.) The expansion for $f(t)$ is in terms of scaled Laguerre functions

$$f(t) = e^{\sigma t} \sum_{n=0}^{\infty} a_n e^{-bt} L_n(2bt), \tag{3}$$

where the L_n are the Laguerre polynomials, and σ and b are parameters ($\sigma > \sigma_0$, $b > 0$). The connection to $F(z)$ is

$$F(z) = \frac{(1-w)}{2b} \sum_{n=0}^{\infty} a_n w^n, \tag{4}$$

where the variables w and z are related by the conformal map

$$w = \frac{\sigma + b - z}{\sigma - b - z}. \tag{5}$$

The efficiency of these series for computational usage depends on the rate of decay of the expansion coefficients a_n as $n \to \infty$. It is a standard exercise to estimate this from power series like the one on the right of (4). Those results then have to be transplanted from the w-plane to the z-plane. Now (5) is a bilinear transformation that maps (generalized) circles in the z-plane to circles in the w-plane and vice versa; for a diagram we refer to [30]. Specifically, the line $\mathrm{Re}\, z = \sigma$ gets mapped to the unit circle $|w| = 1$ in the w-plane. The half-planes $\mathrm{Re}\, z > \sigma$ (resp. $\mathrm{Re}\, z < \sigma$) get mapped to the interior (resp. exterior) of $|w| = 1$. Because $\sigma > \sigma_0$, $F(z)$ is analytic in the half-plane $\mathrm{Re}\, z > \sigma$, and this means analyticity in the closed disk $|w| \leq 1$. The power series in (4) therefore converges with radius of convergence R, where $R > 1$. The larger R, the quicker the a_n decay to zero, which follows from the Cauchy estimates [5, p. 52]

$$|a_n| \leq \frac{M(r)}{r^n}, \quad n = 0, 1, 2, \ldots \tag{6}$$

Here $M(r)$ is the maximum of $|2bF(z)/(1 - w)|$ on the circle $|w| = r$, with $r < R$.

Example: Consider the case $f(t) = \sin t$, $F(z) = 1/(z^2 + 1)$, and choose parameters $\sigma = b = \frac{1}{2}$. For these choices the expansion coefficients are known explicitly, namely [16, p. 42]

$$a_n = 2^{-(n+1)/2} \cos\Big(\frac{\pi}{4}(n + 1)\Big), \quad n = 0, 1, 2, \ldots \tag{7}$$

On the other hand, the exponential behaviour $a_n = O(2^{-n/2})$ could have been predicted from the estimates (6) and the fact that the poles of $F(z)$ are at $z = \pm i$. This corresponds to $w = 1 \pm i$, and the circle that passes through these two points is $|w| = R$, with $R = \sqrt{2}$.

In summary, the convergence of the series (3)–(4) therefore hinges on whether the convex hull of all singularities of $F(z)$ can be covered by a circle in the z-plane. If so, the a_n will decay to zero at the geometric rate determined by (6). The smaller the circle in the z-plane, the larger the circle in the w-plane, which means a more rapid geometric convergence rate. Optimal choices of σ and b will maximize the radius of the critical circle in the w-plane. When the convex hull of singularities of $F(z)$ cannot be covered by a circle, however, the rate of convergence switches from geometric to algebraic or worse, in which case these series are no longer attractive from a computational point of view. One example in this class is $F(z) = 1/\sqrt{z}$, which corresponds to the singular $f(t) = 1/\sqrt{\pi t}$.

The use of these expansions for computing the forward transform is the main novelty of this paper. We assume that $f(t)$ is a known function defined on $[0, \infty)$. The first step is to choose suitable parameters σ and b and then to

compute the a_n from (3). Probably the most straightforward way to do this is to use the orthogonality properties of the Laguerre functions to obtain

$$a_n = \int_0^\infty e^{-\frac{1}{2}t(1+\sigma/b)} f\big(t/(2b)\big) L_n(t)\, dt, \quad n = 0, 1, 2, \ldots \tag{8}$$

These integrals can be evaluated by quadrature, and the Gauss-Laguerre rule seems a natural candidate. (A code is included in the appendix.) For an alternative way of computing Laguerre coefficients, which may also be of use when $f(t)$ is known only through sampled values, see [27].

Once the a_n are available, the right-hand side of (4) can be evaluated by direct summation. One cannot expect accuracy in the entire z-plane, however, only in the region away from the singularities of $F(z)$ (corresponding to $|w| < R$). Relatedly, the pole at $z = \sigma - b$ of (5) should not be a concern for computations. Values of z near this point correspond to values of w outside the region of convergence.

To improve the region of accuracy of (4), one could sum the series not directly but with Padé summation. Here it means that if $2M + 1$ terms in the power series (4) is retained, it is converted to a type (M, M) rational Padé approximant of the series, which is then evaluated. This is a well-known technique for summing power series beyond their radius of convergence. Not only does this give better approximation near the singularities of $F(z)$, but it also improves accuracy elsewhere. The numerical results of section 3 will show the effectiveness of Padé summation in the current situation.

Going in the other direction, the use of (3)–(4) for the inversion problem is the well-established method of Weeks [15,20,29]. Here it is assumed that $F(z)$ on the left of (4) is available and can be evaluated on the unit circle in the w-plane. In this case the expansion coefficients a_n can be computed by a discrete Fourier transform (computed directly by Weeks, and by an FFT by later practitioners). This then defines the original function $f(t)$ via (3).

A final word about selecting the parameters σ and b. For the inversion problem, i.e., when $F(z)$ is available, algorithms for computing optimal values of these parameters were suggested in [30]. For the forward problem, however, $F(z)$ is not known and these algorithms do not apply. One could use trial-and-error in order to maximize the decay rate of the a_n (as computed by (8)) over a range of n, but in the examples below the singularity structure of the $F(z)$ is such that we could get good results with the generic choice $\sigma = b = 1$.

2.2 Contour deformation

It is unclear whether methods for the forward transform can benefit from the ideas of this section, so we discuss only the inversion problem here.

Butcher [4] was the first to suggest that judicious deformation of the contour in (2) could lead to efficient methods for Laplace transform inversion. Specifically, the contour should be of Hankel type, i.e., starting at infinity in the third quadrant and ending at infinity in the second quadrant while

avoiding all singularities of the transform; see Fig. 1. It is also assumed that $|F(z)| \to 0$ uniformly in $\mathrm{Re}\, z \le \sigma$ as $|z| \to \infty$ (which disqualifies transforms like $F(z) = e^{-z}$, $f(t) = \delta(t-1)$.)

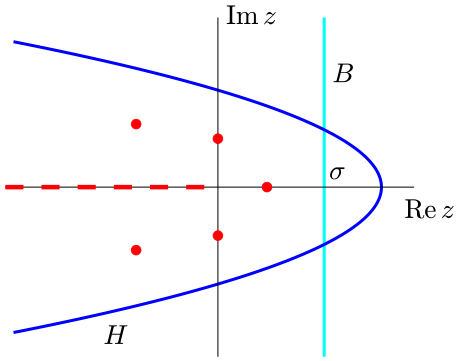

Fig. 1 Contour deformation for the computation of (2). The Bromwich line B is deformed into the Hankel contour H to force rapid decay of the integrand in the left half-plane. No crossing of singularities such as poles (dots) or branch cuts (dashes) is allowed.

If the Hankel contour is parameterized by $z = z(\phi)$, $-\infty < \phi < \infty$, then the integral (2) transforms into

$$f(t) = \frac{1}{2\pi i} \int_{-\infty}^{\infty} e^{z(\phi)t} F(z(\phi)) z'(\phi)\, d\phi. \tag{9}$$

The power of the deformation technique should now be apparent: the integrand of (9) is a rapidly decaying function and hence suitable for efficient approximation by the truncated trapezoidal or midpoint quadrature rules [28].

Butcher suggested a parabola as the simplest type of contour. Later, Talbot [26] came up with essentially the same idea, except for a different type of contour defined in terms of the cotangent function. For the results reported in this paper we shall use a parabola, taken in the form used in [32], namely

$$z = \mu\big(i\phi + 1\big)^2. \tag{10}$$

Here μ is a positive constant. The contour intersects the real axis at $z = \mu$ and the imaginary axis at $z = \pm 2\mu i$.

The integral (9) is approximated by a midpoint sum with step size h, and then truncated to $2J$ terms, symmetrically placed around $\phi = 0$. (We use the terms midpoint sum and trapezoidal sum interchangeably as these two rules are equally accurate in the current situation.) By using up-down symmetry it means a sum of length J has to be evaluated; see [32]. The question is, for a given J, what are good choices for μ and the step size h? By balancing the various errors in the trapezoidal/midpoint rule, optimal values for these

parameters can be found for the contour (10) in the case where all singularities of the transform $F(z)$ lie on the negative real axis [32]. These values are

$$h = \frac{3}{J}, \quad \mu = \frac{\pi J}{12\,t}, \tag{11}$$

and the associated rate of convergence is geometric, namely of the form $O(e^{-2\pi J/3})$.

Other types of contours, including Talbot's, can give slightly faster rates but the optimal parameters are not explicitly computable as in (11) [9]. Even when the singularities are not restricted to the real axis the choice (11) can give good results even if not perfectly optimal. Below, we shall just use (11) without regard to the particular singularity structure of $F(z)$, which is often unknown anyway. This is especially true in the situation of this paper where $F(z)$ is available only in numerical form.

Depending on the singularity structure of $F(z)$, the convergence rate of the Weeks method can be faster or slower than the rate achieved by contour deformation. However, for a transform such as $F(z) = 1/\sqrt{z}$, with branch cut defined along the entire negative real axis, the convergence rate of Weeks will no longer be geometric as explained in section 2.1. By contrast, the contour deformation method will invert this with the geometric rate cited below (11).

It should be noted that the optimal contour (10)–(11) depends on the value of t. This means that a new contour is used for every new t, and transform evaluations cannot be re-used. In the Weeks case the Laguerre expansion (3) can be evaluated for new values of t without additional cost. This should be taken into account if the transform $F(z)$ is expensive to compute. (One example is the problem of section 3.3 below, where a large linear system has to be solved in order to compute $F(z)$ at each midpoint node.)

In terms of simplicity and generality the method described in this section is currently among the top Laplace inversion techniques, and therefore we shall use it for most of our computations in section 3. It is not always guaranteed to be effective, however, particularly when the transform to be inverted has singularities that extend far from the real axis. This will not be the case in our examples, but if the situation occurs different inversion techniques should be considered.

2.3 Exponential sums

Laplace transform methods in many areas of engineering (particularly control theory [1]) are based on exponential sums of the form

$$f(t) = \sum_{n=1}^{M} \alpha_n e^{\lambda_n t} \tag{12}$$

with

$$F(z) = \sum_{n=1}^{M} \frac{\alpha_n}{z - \lambda_n}. \tag{13}$$

For the forward transform, the coefficients (α_n, λ_n) can be computed from (12) by any of the methods descended from the original (unstable) de Prony's method (such as the matrix pencil method, ESPRIT, ESPIRA, MUSIC, etc. [2,3,7,22,23]). These methods differ in details, but they all sample the function $f(t)$ on some interval $[0, T]$ specified by the user. The significant frequencies λ_n are then obtained by an SVD or similar, upon which the corresponding amplitudes α_n are found by solving (12) as an overdetermined system. In some variations the number of terms, M, on the right of (12) is prescribed, and in others a target accuracy. In all cases M is much smaller than the number of samples of $f(t)$.

Once the (α_n, λ_n) are available, the forward transform (13) can then be evaluated anywhere in the complex plane where the inversion algorithm requires it, excluding at the poles. This procedure was suggested in [24] for dealing with differential equations with inhomogeneous terms.

Inversion methods based on (12)–(13) compute the coefficients (α_n, λ_n) from (13) by using, for example, Padé approximation [19, sect. 16.4] or best approximation [18]. The original function can then be approximated via (12). We shall not use this inversion technique in our comparisons of section 4, however. Instead, in order to use the same inversion method for all forward transform methods, we shall use the contour deformation technique of section 2.2.

3 Three Illustrative Examples

Since we consider the Weeks method for computing the forward transform as a novelty of the present paper, we start with a demonstration of its capabilities. Comparisons with the exponential sum method will follow in the next section.

3.1 A nonlinear integral equation

This is a text book example, taken from [25, p. 126]:

$$u(t) - \int_0^t u(s)u(t-s)\,ds = f(t), \quad 0 \le t < \infty. \tag{14}$$

Let $U(z)$ and $F(z)$ denote the Laplace transforms of $u(t)$ and $f(t)$, respectively. By taking Laplace transforms of the equation and using the convolution theorem and quadratic formula one finds that

$$U(z) = \frac{1}{2}\left(1 \pm \sqrt{1 - 4F(z)}\right). \tag{15}$$

With $f(t) = \frac{1}{2}\sin 2t$, i.e., $F(z) = 1/(z^2+4)$, and by taking the minus sign one obtains

$$U(z) = \frac{1}{2}\Big(\frac{\sqrt{z^2+4} - z}{\sqrt{z^2+4}}\Big). \tag{16}$$

Inversion by table look-up produces a solution in terms of Bessel functions, namely $u(t) = J_1(2t)$. (The other branch of the square root gives a singular solution.) The forcing function $f(t)$ and response $u(t)$ are shown in Fig. 2.

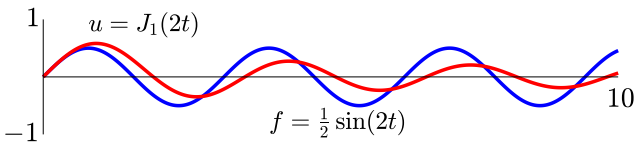

Fig. 2 Solution to the integral equation (14) on the interval $[0, 10]$. The forcing function f is shown in blue and the solution u in red.

In order to demonstrate the accuracy of the Weeks approach, the exact forward transform formula $F(z) = 1/(z^2 + 4)$ is disregarded and instead $F(z)$ is approximated by the method of section 2.1. The first step is to compute the Laguerre coefficients in (3), which is done here by approximating the integral (8) by Gauss-Laguerre quadrature. If N is the number of terms retained in (3), then we use $2N$-point quadrature to allow for oversampling. Parameters were taken as $\sigma = b = 1$, for which the exact coefficients can be deduced from the example below (6). (We have used this information as an accuracy check only, the coefficients that were used in the results below were computed numerically.) Once the coefficients are available, the transform $F(z)$ can be computed for any z by summation of (4), which then gives the value of the transform $U(z)$ via (15). At this point one can proceed with the inverse transform. The new expansion coefficients, say a'_n, can be computed by an FFT from (4) (with $U(z)$ on the left). The final solution $u(t)$ can then be computed by inserting the coefficients a'_n into (3).

In Fig. 3 the errors, defined by

$$\text{Error} = \max_{0 \le t \le T} |u(t) - u^{LT}(t)|, \tag{17}$$

are shown as a function N, the number of terms in the expansions (3)–(4). Here $u(t)$ and $u^{LT}(t)$ are the exact and numerical solutions, respectively. Evidently the convergence rate is geometric, more specifically of the form $O(2^{-N/2})$, which is directly related to the decay rate of the expansion coefficients; cf. (7). The implied constant in this convergence rate grows exponentially with T, however, so this method is probably only practical on relatively short t intervals.

In the computation of Fig. 3, direct summation was used in (4). Padé summation offered no improvement, because the errors were dominated by the inverse transform rather than the forward.

The choice $\sigma = b = 1$ was a generic one. By varying one or both these parameters faster convergence can be achieved. For example, by keeping $\sigma = 1$ but allowing b to vary, one could use the methods of [15,30] to show that $b = \sqrt{5}$ is optimal. (The calculations assumes knowledge of the fact that the singularities of the transform are branch points located at $z = \pm 2i$.) With this

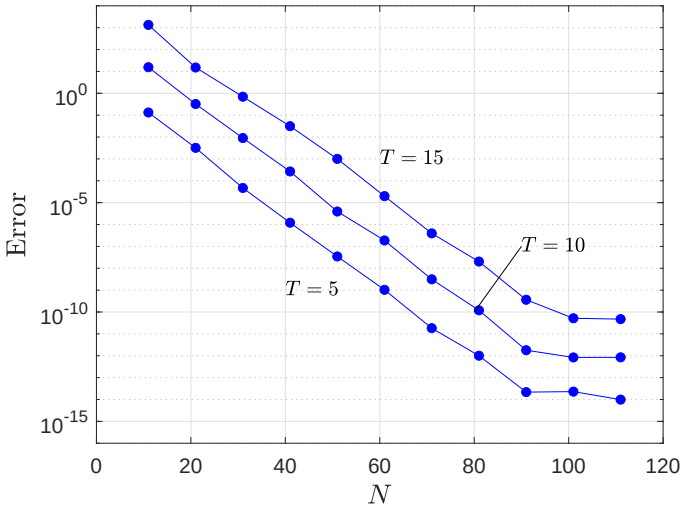

Fig. 3 Numerical results for the integral equation (14), with $f(t) = \frac{1}{2}\sin 2t$. Shown are sup-norm errors (17) on intervals of various lengths as a function of the number of terms in the series (3)–(4).

parameter choice the convergence rate improves from $O(2^{-N/2}) \approx O(0.71^N)$ to $O(\varphi^{-N}) \approx O(0.62^N)$, where φ is the golden ratio.

Although the results of Fig. 3 are encouraging, it should be emphasized that this example is particularly well suited to using Weeks inversion. The reason becomes clear if one looks at the transform (16), the approximate version of which that gets inverted. This transform has singularities at $z = \pm 2i$, and they can be connected by a short branch-cut. This is the favourable situation in which Weeks inversion converges geometrically, as described below (4). The examples of the next two sections involve transforms that are more challenging to Weeks inversion and therefore require a different method.

3.2 A fractional differential equation

The Bagley-Torvik equation is a well-known model problem in the field of fractional differential equations [8,21]. It represents a driven harmonic oscillator with fractional damping

$$u''(t) + u^{(\nu)}(t) + u(t) = f(t), \tag{18}$$

where $u^{(\nu)}$ represents a fractional derivative in the sense of Caputo. With initial conditions $u(0) = u'(0) = 0$, Laplace transformation produces the transform

$$U(z) = \frac{F(z)}{z^2 + z^\nu + 1}. \tag{19}$$

From a numerical perspective the difference between the ordinary and fractional differential equation lies in the following fact. In the ordinary case the

denominator of the transform defines pole singularities, while in the fractional case there is a branch point (ν integer vs noninteger).

With $f(t) = J_0(2t)$, i.e., $F(z) = 1/\sqrt{z^2+4}$, numerical inversion of the transform (19) produced the solution graphs shown in Fig. 4. (The exact solution can, in fact, be expressed as an infinite sum involving Mittag-Leffler functions [8, p. 184].)

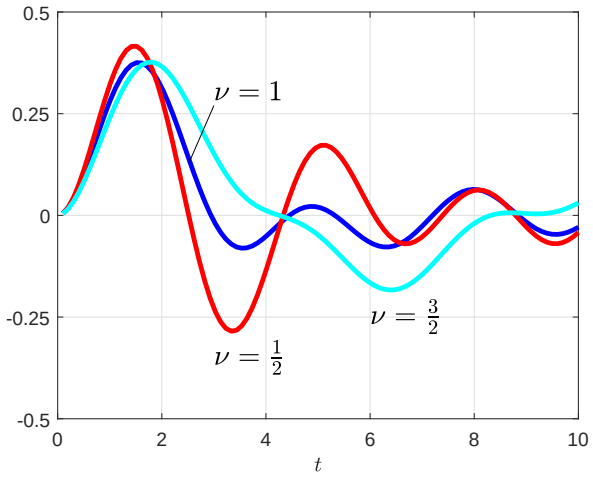

Fig. 4 Solutions to the Bagley-Torvik equation (18), with $f(t) = J_0(2t)$, $\nu = \frac{1}{2}, 1, \frac{3}{2}$, and initial conditions $u(0) = u'(0) = 0$.

For the numerical computations below, we have not used the exact transform $F(z) = 1/\sqrt{z^2+4}$ like we did in Fig. 4 but instead computed it by the forward Weeks method described in section 2.1. The results are shown in Fig. 5.

The error curve A corresponds to using the Weeks method for both the forward and the inverse transform. Unlike the error curves in Fig. 3, which were computed by the same method, the convergence rate is no longer geometric but has deteriorated into a slow algebraic rate. The reason for this is clear from the transform (19). When ν is noninteger (here $\nu = \frac{1}{2}$), the transform has a branch point at $z = 0$, with a branch cut along the entire negative real axis (in our implementation of the transform). This causes the Weeks inversion method to converge slowly. Switching from direct summation to Padé summation in (4) does not help because the dominant contribution to the overall error comes from the inverse.

By contrast, the error curves B and C show geometric convergence. For both of these the inverse was computed not with the Weeks method but with contour deformation, the convergence of which is not adversely affected by a branch cut on the entire negative real axis. (To be specific, we used the method based on the contour (10), with parameters (11), and 60 terms in the midpoint sum.)

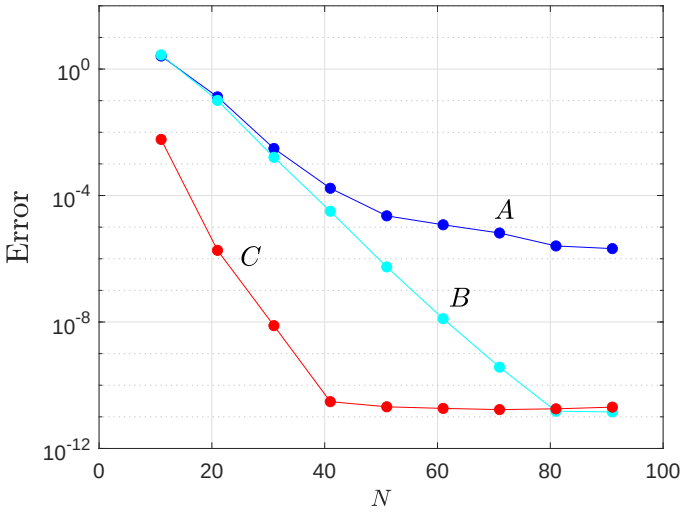

Fig. 5 Numerical results for the Bagley-Torvik equation (18), with $f(t) = J_0(2t)$, $u(0) = u'(0) = 0$, and $\nu = \frac{1}{2}$. Shown are sup-norm errors defined by (17), with $T = 10$, as a function of the number of terms in the series (3)–(4). In all cases the forward transform was computed by the Weeks method with parameters $\sigma = b = 1$. Both direct summation and Padé summation was used in (4). The inverse was computed by (A) the Weeks inversion method, (B) contour deformation (direct summation), (C) contour deformation (Padé summation). Further details on (A)–(C) are given in the text.

The difference between error curves B and C is caused by us having used direct summation in (4) for B, and Padé summation in C. Evidently the dominant contribution to the overall error here comes from the forward transform, not the inverse. The superiority of Padé summation is clear.

3.3 A partial differential equation

Consider the heat equation

$$u_t = u_{xx}, \qquad 0 < x < 1, \quad t > 0 \tag{20}$$

with initial and boundary conditions

$$u(x,0) = 0, \quad u(0,t) = f(t), \quad u(1,t) = 0. \tag{21}$$

A standard way of dealing with such an inhomogeneous boundary condition is to define a new function

$$v = u + f(t)(x-1). \tag{22}$$

This leads to the new problem

$$v_t = v_{xx} + f'(t)(x-1), \qquad 0 < x < 1, \quad t > 0,$$

subject to

$$v(x,0) = f(0)(x-1), \quad v(0,t) = 0, \quad v(1,t) = 0.$$

We shall use Laplace transforms to deal with the time variable. Discretization of the space variable is by the Chebyhev pseudospectral method, which leads to the system of ordinary differential equations

$$\boldsymbol{v}_t = D_2\boldsymbol{v} + f'(t)(\boldsymbol{x} - \mathbf{1}), \tag{23}$$

with initial condition

$$\boldsymbol{v}(0) = f(0)(\boldsymbol{x} - \mathbf{1}). \tag{24}$$

Here $\boldsymbol{x}$ is the vector of nodes, the Chebyshev points of the second kind scaled to $[0,1]$. The $\boldsymbol{v}$ is the vector of function values corresponding to these nodes, and $\mathbf{1}$ is a vector of ones, of the same size as $\boldsymbol{x}$. D_2 is the second derivative Chebyshev matrix with the homogeneous boundary conditions on v enforced [31].

The semi-discrete system (23) can be solved by any standard ODE solver (the so-called method-of-lines). Indeed, we shall use this as reference solution to test the Laplace based solution method. For this purpose we used MATLAB's stiff-solver `ode15s`, with error tolerances set to 10^{-12}.

Taking the Laplace transform of (23) and making use of the fact that the Laplace transform of $f'(t)$ is $zF(z) - f(0)$, one gets

$$(zI - D_2)\boldsymbol{V}(z) = zF(z)(\boldsymbol{x} - \mathbf{1}). \tag{25}$$

If $F(z)$ can be computed at the values of z required by the inversion scheme, this linear system can then be solved for the corresponding value of $\boldsymbol{V}(z)$. In this way $\boldsymbol{v}$ can be computed and hence the solution to (20)–(21) can be recovered via (22).

Consider as test case a function with a known transform, namely

$$f(t) = \frac{1}{1+t}, \quad F(z) = e^z E_1(z), \tag{26}$$

where E_1 is the exponential integral defined by [10, eq. (6.2.1)]. This transform has a branch point at the origin, and is defined with branch cut along the entire negative real axis. This means that the Weeks method for inversion is bound to converge slowly and we have not considered it. Instead, the contour deformation method based on (9)–(11) should be tailor made since the transform is analytic everywhere off the branch cut. Because of this, we could get away with as few as 12 nodes in the midpoint sum when $t = 1$. The errors for this example are shown in Fig. 6.

We ignored the availability of the explicit formula for $F(z)$ and instead computed it by the Weeks method (4). The coefficients, a_n, were computed from (8), with parameters $\sigma = b = 1$. The errors are shown as a function of the number of terms, N, in the series (4). We have used both direct summation and Padé summation to evaluate this series, with the Padé method yielding distinctly superior results.

The error norm in Fig. 6 is defined as follows. If $\boldsymbol{v}^{SD}(t)$ is the solution to the semi-discrete system (23) and $\boldsymbol{v}^{LT}(t)$ the solution obtained by inverting $\boldsymbol{V}(z)$ defined by (25), then

$$\text{Error} = \|\boldsymbol{v}^{SD}(t) - \boldsymbol{v}^{LT}(t)\|_\infty. \tag{27}$$

The solution $\boldsymbol{v}^{SD}(t)$ was computed by `ode15s` as mentioned above. For the space discretization 100 Chebyshev points were used, which is more than sufficient to guarantee that spatial errors are subdominant to temporal errors.

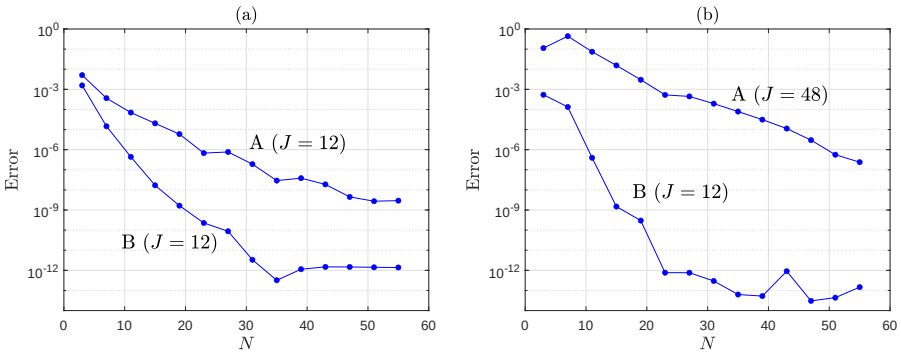

Fig. 6 Numerical results for the heat equation (20)–(21), with $f(t) = 1/(t+1)$. In (a) $t = 1$ and in (b) $t = 10$. Shown is the error defined by (27) as a function of N, the number of terms used in the series (4). Direct and Padé summation of this series corresponds, respectively, to curves A and B. J is the number of nodes in the midpoint sum for the inversion; cf. (11).

Regarding the convergence rate observed in Fig. 6, it seems to be subgeometric in (a). (In (b) it is harder to say). By contrast, the straight line convergence curves in Figs. 3 and 5 implied geometric convergence. In the latter two test cases the forcing functions $f(t)$ were entire functions (sine and Bessel), whereas the function of Fig. 6 has a singularity at $t = -1$. This causes the decay of the coefficients a_n to turn from exponential decay (cf. (7), for example) to root-exponential $O(e^{-c\sqrt{n}})$. (An empirical fit yielded $c \approx 2.9$ for this example. The exact value of c can probably be obtained from [13, example 1], but we have not pursued it.) The convergence curves in Fig. 6(a) reflect this subgeometric trend. Estimates of the rate of decay of Laguerre coefficients in various function classes are derived in [13,33].

A question that may be raised is why use Laplace transforms if an ODE solver is all that is required? The attractive feature of the Laplace transform is that if the solution is required only at a particular value, say $t = T$, the method can compute it directly. By contrast, the ODE solver steps through the entire interval $[0, T]$. For example, to compute the reference solutions of Fig. 6 to the accuracy desired here, the solver `ode15s` executed over 400 LU decompositions in each case. By contrast, the Laplace method required only 12 in Fig. 6(a), one for each node in the midpoint sum (with further improvements possible if a Hessenberg decomposition is considered rather than an

LU [17].) (In Fig. 6(b) we had to increase the number of nodes to 48 when direct summation is used, as will be explained below.) Moreover, the linear solves (25) can be done in parallel, which would be a major advantage when going from the one dimensional heat equation to two dimensions or higher. To be fair, the Laplace transform technique is only applicable to linear ODEs with constant coefficients, while methods based on ODE solvers impose no such restrictions.

The aim of the final figure of this section, Fig. 7, is to shed some light on the results shown in Fig. 6. The figure shows contour plots of the error in the computed transform $F(z)$ in a square in the complex plane (this time making use of the exact formula $F(z) = e^z E_1(z)$ to be able to compute errors). Superimposed on that plot are the actual contour and quadrature nodes used in the inversion formula in the case $t = 1$. Padé summation yields an improvement over direct summation that is quantitatively in agreement with the improvement in accuracy observed in Fig. 6. Also, by increasing the value of t from 1 to 10, the contour of integrations moves closer to the origin; cf. (11). In Fig. 7(a) the contour then passes close to the high error area (yellow), and the method diverges. Therefore, the value of J had to be increased from $J = 12$ to $J = 48$ to obtain the results of Fig. 6(b). No such increase was necessary with Padé summation.

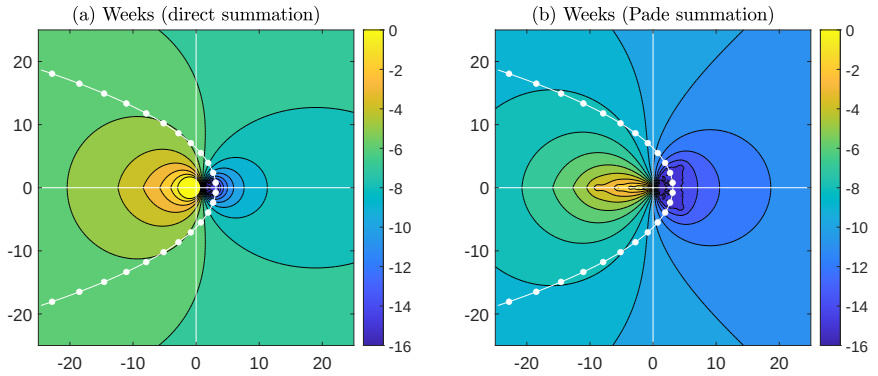

Fig. 7 Contour plot of absolute errors (log base-10) in the computation of the transform $F(z)$ defined by (26). The series (4) with $N = 23$ terms was summed directly in (a) and with Padé summation in (b). In each figure the white curve is the contour (10) and the white dots the corresponding midpoint nodes, for the case $J = 12$. In both (a) and (b) the contours pass through the high accuracy areas just to the right of the origin (dark blue), which corresponds to terms that make the biggest contribution to the integral. This feature contributes to the success of these methods. Looking also at the errors along the length of the contour into the left half-plane, one sees that the contour traverses regions where the errors in (b) are about 4 to 5 orders of magnitude smaller than in (a). This accounts for the difference in accuracy observed at $N = 23$ in Fig. 6.

We now turn to comparisons of the Weeks method and the method of exponential sums for computing $F(z)$ in the complex plane.

4 Errors in the complex plane

Ideally, one would like to compare the Weeks and exponential sum methods on model problems such as those solved in section 3. A fair comparison is no simple task, however. First, there is a multitude of methods available for computing the parameters in the exponential sum; recall the list in section 2.3. We have implemented here just one of these, namely the algorithm described in [2, 3]. Second, the exponential sum methods come from a different background than the Weeks method, namely, signal processing vs approximation theory. In practice the difference is that in the exponential sum methods the user specifies a sampling interval and an accuracy tolerance (or the number of terms in the sum), while in the Weeks method scaling parameters (b and σ) are specified. Third, the combination with an inversion method complicates things further. A good inversion method for Weeks may not be optimal for exponential sums and vice versa. Fourth, the computational complexities of the two methods are quite different. The Weeks method requires the computation of the quadratures (8) and, if Padé summation is used, the equivalent of solving a linear system. The various methods for computing exponential sums typically require an SVD or similar plus a least-squares solver, as explained below (13).

We shall therefore remove the complications of computational cost and inversion techniques completely from the comparisons, and simply look at errors in the forward transform in the display format of Fig. 7. To make the comparisons somewhat meaningful, we decided on the following rules:

(a) All functions $f(t)$ are sampled at precisely 102 values. For the Weeks method this means the series (3) has 51 terms, in order to allow a factor of two oversampling in the quadratures of (8). The Padé summation of (4) is therefore based on a type $(25, 25)$ approximant.

(b) For the parameter choices we used the generic $\sigma = b = 1$ in the Weeks method. For the exponential sum method $f(t)$ was sampled at equidistant points on $[0, 1]$ and the tolerance parameter was set to 10^{-12} in the algorithm of [2,3].

As test problems we considered the following, in decreasing order of regularity of $f(t)$. In the first, $f(t)$ is an entire function, in the second it is analytic everywhere except for a singularity on the negative t-axis, and in the third $f(t)$ has a singularity at the origin.

Test 1: $f(t) = J_0(t)$, $F(z) = 1/\sqrt{z^2+1}$ (branch cut between $z = \pm i$)
Test 2: $f(t) = 1/(t+1)$, $F(z) = e^z E_1(z)$ (branch cut on negative real axis)
Test 3: $f(t) = \sin(\sqrt{t})$, $F(z) = \frac{1}{2}\sqrt{\pi} e^{-1/(4z)}/z^{\frac{3}{2}}$ (essential singularity at the origin and branch cut on negative real axis)

The results of Figs. 8–10 should merely give an indication of the accuracy that can be achieved by these methods with rather generic parameter choices. Although the results do not appear to be super sensitive to changes in these parameters, smaller errors can no doubt be achieved with more careful tuning. For this reason one should resist drawing too many conclusions regarding the superiority of one method over the other just based on these plots.

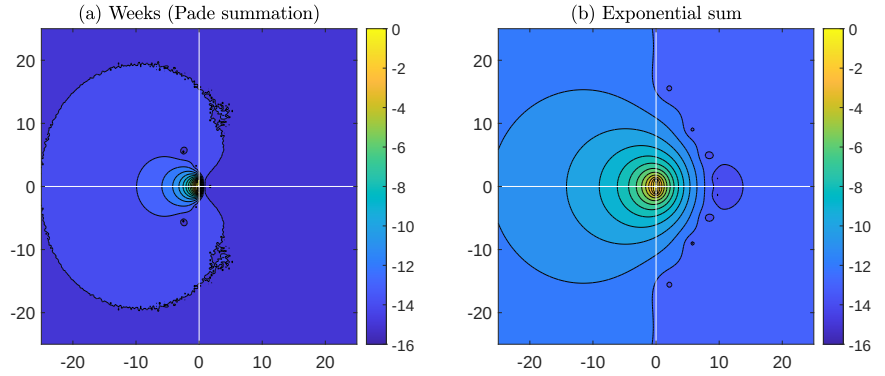

Fig. 8 Contour plot of absolute errors (log base-10) in the computation of the transform $F(z)$ defined by Test 1. In (a) the Weeks method (with Padé summation) was used and in (b) an exponential sum. Parameters are defined in the text.

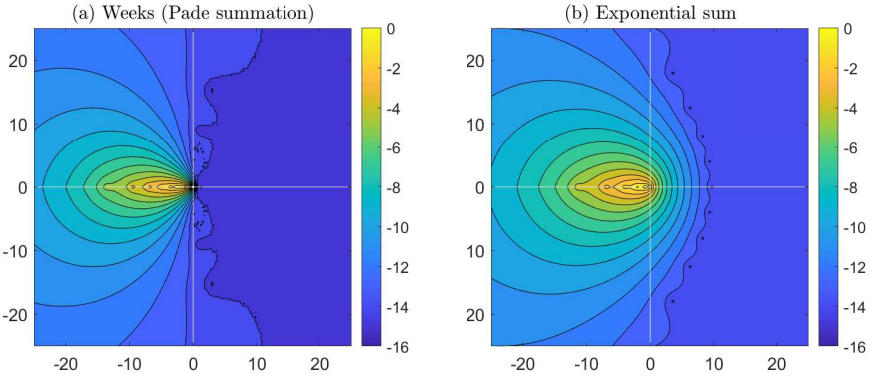

Fig. 9 Same as Fig. 8 but for Test 2.

Moreover, as mentioned above the difference in computational cost is not reflected in these results. To compute the coefficients, the Weeks method is cheaper than the method of exponential sums, as outlined in the first paragraph of this section. Once the coefficients of the exponential sum are available, however, the approximate formula for $F(z)$ defined by the right side of (13) is cheaper to evaluate than (4). The reason for this is that even if the number of samples of $f(t)$ on the left of (12) is large, the effective number of terms on the right can be relatively small. (For example, the results of Figs. 8–10 all sampled the function at 102 values, but the exponential sums generated were respectively of lengths 5, 6, and 13.) This advantage is not a deciding factor, however, as the computational cost of computing the forward transform dominates that of the inverse (when the latter is based on efficient methods such those discussed in section 2.2).

As remarked in the caption of Fig. 7, the Weeks-Padé method seems excellent for approximating near and to the right of the singularities of the

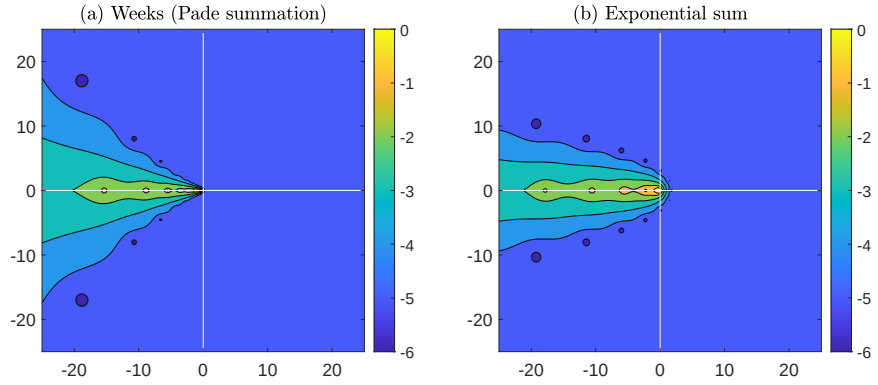

Fig. 10 Same as Fig. 8 but for Test 3. (Note that the colour scale in this plot is different from that in Figs. 8–9.)

transform. If the inversion scheme samples the transform in those areas high accuracy can result. Fig. 11 shows the results when the forward transform as computed in Fig. 8 is inverted by the method described in section 2.2. With the parameter values chosen as in (11) with $J = 20$, the contours are indeed located in the good regions for the Weeks-Padé method. For example, for $t = 1$ the contour crosses the real axis at $x = 20\pi/12 \approx 5.2$ and the imaginary axis at $y = \pm 10.4$. Consequently the contour passes through regions where the accuracy is near machine precision, and this is reflected in the results of Fig. 11.

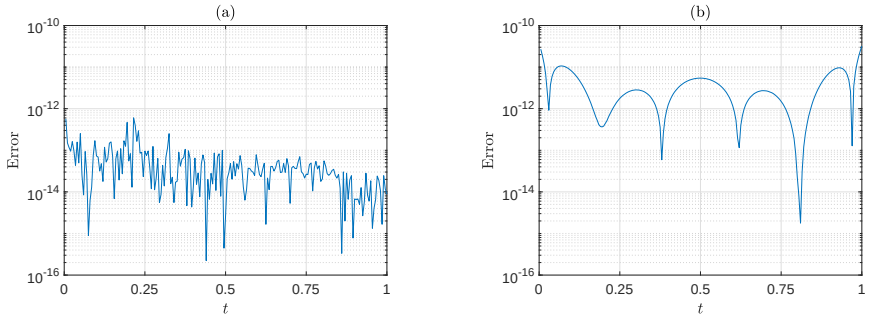

Fig. 11 Absolute errors when the forward transform of $f(t) = J_0(t)$ is computed and the result then inverted to return to $f(t)$. The inversion is by the contour deformation method of section 2.2. In (a) the Weeks-Padé method was used to compute the forward transform and in (b) exponential sums. Because the inversion method samples the transforms in regions where the Weeks-Padé errors are smaller (cf. Fig. 8), higher accuracy of about two orders of magnitude is achieved.

5 Conclusions

This paper focused on the computation in the complex plane of the forward Laplace transform formula (1). The novel method proposed here is based on the Weeks method, a well-known method for computing the inverse but reversed here for computing the forward transform. Previously suggested methods for the forward problem based on exponential sums are also considered, here with attention to techniques developed relatively recently for computing the coefficients in such expansions.

For reasons outlined in the first paragraph of section 4 a head-to-head, quantitative comparison of the Weeks and exponential sum methods is no simple task. On a qualitative level, however, a few comparisons can be made. Regarding the theory, there seems to be very few convergence results for exponential sums of the form (12). By contrast, convergence in the Weeks method depends on the decay of the expansion coefficients in (3), for which there exist plentiful results in the literature; see [13,33] for example. Such information should give an a priori indication of the possible success or failure of the method, something that does not seem to exist for exponential sums. Regarding practical issues, the coefficients in the Weeks method is cheaper to compute. The main work comes from the computation of (8) by quadrature. Assuming the nodes and weights of the rule have been pre-computed, this is equivalent to a matrix vector multiplication (see the code in the appendix). If Padé summation is used for accuracy enhancement, the equivalent of solving a linear system (of half size) has to be added to that cost. By contrast, the methods based on exponential sums typically compute an SVD followed by the solution of an overdetermined linear system of Vandermonde type. Regarding numerical stability, methods based on exponential sums are notoriously unstable, although the method implemented here [2,3] seems to have overcome that issue largely. Experiments such as those of sections 3 and 4 have not revealed any similar instability issues in the Weeks method.

Because we wished to illustrate the concept of the Weeks method for a fully numerical Laplace transform approach, the examples presented here were nontrivial but of academic type. In all cases but one, the functions $f(t)$ were analytic in a wide region about $[0, \infty)$, which meant rapidly decaying expansion coefficients a_n and hence fast convergence. When faced with more challenging problems, such as Test 3 in section 4 or more complicated cases, it is unclear whether this method (or the method of exponential sums, for that matter) will be up to the challenge. The development of new methods that can handle such difficult cases seems like a fruitful area for future research.

Acknowledgements Annie Cuyt, Volker Mehrmann, Gerlind Plonka, Daniel Potts, and Rina-Marí Weideman generously shared advice and software related to exponential sum approximations.

References

1. P. Benner, V. Mehrmann, V. Sima, S. Van Huffel, and A. Varga. SLICOT—a subroutine library in systems and control theory. In *Applied and computational control, signals, and circuits, Vol. 1*, volume 1 of *Appl. Comput. Control Signals Circuits*, pages 499–539. Birkhäuser Boston, Boston, MA, 1999.
2. G. Beylkin and L. Monzón. On approximation of functions by exponential sums. *Appl. Comput. Harmon. Anal.*, 19(1):17–48, 2005.
3. G. Beylkin and L. Monzón. Approximation by exponential sums revisited. *Appl. Comput. Harmon. Anal.*, 28(2):131–149, 2010.
4. J. C. Butcher. On the numerical inversion of Laplace and Mellin transforms. Conference on Data Processing and Automatic Computing Machines. Salisbury, Australia, 1957.
5. G. F. Carrier, M. Krook, and C. E. Pearson. *Functions of a complex variable: Theory and technique.* McGraw-Hill Book Co., New York-Toronto, Ont.-London, 1966.
6. A. M. Cohen. *Numerical methods for Laplace transform inversion*, volume 5 of *Numerical Methods and Algorithms.* Springer, New York, 2007.
7. N. Derevianko, G. Plonka G., and M. Petz. From ESPRIT to ESPIRA: Estimation of signal parameters by iterative rational approximation, 2022.
8. K. Diethelm. *The analysis of fractional differential equations*, volume 2004 of *Lecture Notes in Mathematics.* Springer-Verlag, Berlin, 2010.
9. B. Dingfelder and J. A. C. Weideman. An improved Talbot method for numerical Laplace transform inversion. *Numer. Algorithms*, 68(1):167–183, 2015.
10. *NIST Digital Library of Mathematical Functions.* http://dlmf.nist.gov/, Release 1.0.17 of 2017-12-22. F. W. J. Olver, A. B. Olde Daalhuis, D. W. Lozier, B. I. Schneider, R. F. Boisvert, C. W. Clark, B. R. Miller and B. V. Saunders, eds.
11. T. A Driscoll, N. Hale, and L. N. Trefethen. *Chebfun Guide.* Pafnuty Publications, 2014.
12. H. Dubner and J. Abate. Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *J. Assoc. Comput. Mach.*, 15:115–123, 1968.
13. D. Elliott and P. D. Tuan. Asymptotic estimates of Fourier coefficients. *SIAM J. Math. Anal.*, 5:1–10, 1974.
14. A. Gil, J. Segura, and N. M. Temme. Efficient computation of Laguerre polynomials. *Comput. Phys. Commun.*, 210:124–131, 2017.
15. G. Giunta, G. Laccetti, and M. R. Rizzardi. More on the Weeks method for the numerical inversion of the Laplace transform. *Numer. Math.*, 54(2):193–200, 1988.
16. D. Gottlieb and S. A. Orszag. *Numerical analysis of spectral methods: theory and applications.* CBMS-NSF Regional Conference Series in Applied Mathematics, No. 26. Society for Industrial and Applied Mathematics, Philadelphia, Pa., 1977.
17. N. Hale and J. A. C. Weideman. Contour integral solution of elliptic PDEs in cylindrical domains. *SIAM J. Sci. Comput.*, 37(6):A2630–A2655, 2015.
18. I. M. Longman. Best rational function approximation for Laplace transform inversion. *SIAM J. Math. Anal.*, 5:574–580, 1974.
19. Y. L. Luke. *The special functions and their approximations. Vol. II.* Mathematics in Science and Engineering, Vol. 53. Academic Press, New York-London, 1969.
20. J. N. Lyness and G. Giunta. A modification of the Weeks method for numerical inversion of the Laplace transform. *Math. Comp.*, 47(175):313–322, 1986.
21. I. Podlubny. *Fractional differential equations*, volume 198 of *Mathematics in Science and Engineering.* Academic Press, Inc., San Diego, CA, 1999.
22. D. Potts and M. Tasche. Parameter estimation for nonincreasing exponential sums by Prony-like methods. *Linear Algebra Appl.*, 439(4):1024–1039, 2013.
23. D. Potts, M. Tasche, and T. Volkmer. Efficient spectral estimation by MUSIC and ESPRIT with application to sparse FFT. *Frontiers in Applied Mathematics and Statistics*, 2, 2016.
24. D. Sheen, I. H. Sloan, and V. Thomée. A parallel method for time discretization of parabolic equations based on Laplace transformation and quadrature. *IMA J. Numer. Anal.*, 23(2):269–299, 2003.
25. M. R. Spiegel. *Theory and problems of Laplace transforms.* Schaum Publishing Co., New York, 1965.

26. A. Talbot. The accurate numerical inversion of Laplace transforms. *J. Inst. Math. Appl.*, 23(1):97–120, 1979.
27. A. V. Terekhov. Generating the Laguerre expansion coefficients by solving a one-dimensional transport equation. *Numer. Algorithms*, 89(1):303–322, 2022.
28. L. N. Trefethen and J. A. C. Weideman. The exponentially convergent trapezoidal rule. *SIAM Rev.*, 56(3):385–458, 2014.
29. W. T. Weeks. Numerical inversion of Laplace transforms using Laguerre functions. *J. Assoc. Comput. Mach.*, 13:419–429, 1966.
30. J. A. C. Weideman. Algorithms for parameter selection in the Weeks method for inverting the Laplace transform. *SIAM J. Sci. Comput.*, 21(1):111–128, 1999.
31. J. A. C. Weideman and S. C. Reddy. A MATLAB differentiation matrix suite. *ACM Trans. Math. Software*, 26(4):465–519, 2000.
32. J. A. C. Weideman and L. N. Trefethen. Parabolic and hyperbolic contours for computing the Bromwich integral. *Math. Comp.*, 76(259):1341–1356, 2007.
33. S. Xiang. Asymptotics on Laguerre or Hermite polynomial expansions and their applications in Gauss quadrature. *J. Math. Anal. Appl.*, 393(2):434–444, 2012.

A Appendix: MATLAB code for computing Laguerre coefficients

The following code computes the expansion coefficients (8). It uses Gauss-Laguerre quadrature, the computation of which is done here by a function from Chebfun [11]. The Laguerre polynomials are computed by the well-known three-term recursion [10, sect. 18.9], which is stable in the forward direction. For large n the improved methods of [14] should be considered.

```
function an = LaguerreCoefficients(f,sigma,b,N,M)
%  Function to compute the first N Laguerre coefficients
%  of the function f.   M-point Gauss-Laguerre quadrature is used,
%  where M > N (oversampling).   M = 2*N should be sufficient in
%  many cases.

% First, compute Gauss-Laguerre nodes and weights.  Here Chebfun is used,
% but any equivalent function can be substituted.

[x,w] = lagpts(M);

% Compute the Laguerre polynomials by 3-term recursion.

L0 = ones(size(x));  L1 = 1-x;
L = [L0 L1];

for k = 1:N-2
    L2 = 1/(k+1)*(2*k+1-x).*L1-k/(k+1)*L0;
    L = [L L2];
    L0 = L1; L1 = L2;
end

% Define the integrand

ff = @(x) exp(x*(b-sigma)/(2*b)).*f(x/(2*b));

% Compute the integrals

an = L'*(ff(x).*w');

end
```

Data availability statement: Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.