An Accurate Numerical Inversion of Laplace Transforms Based on the Location of Their Poles

JINWOO LEE AND DONGWOO SHEEN*
Department of Mathematics, Seoul National University
Seoul 151-747, Korea
<jlee><sheen>@nasc.snu.ac.kr

(Received July 2003; revised and accepted August 2004)

Abstract—We introduce an efficient and easily implemented numerical method for the inversion of Laplace transforms, using the analytic continuation of integrands of Bromwich's integrals. After deforming the Bromwich's contours so that it consists of the union of small circles around singular points, we evaluate the Bromwich's integrals by quadrature rules. We prove that the error bound of our method has spectral accuracy of type $\epsilon^N + \epsilon^P$, $0 < \epsilon < 1$ and provide several numerical examples. © 2004 Elsevier Ltd. All rights reserved.

Keywords—Inverse Laplace transform, Numerical integration, Algorithms.

1. INTRODUCTION

Recall that the Laplace transform $\hat{f}(z)$,

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

of $f$ exists, for all $z$ with $\text{Re } z \geq \alpha$, provided $\hat{f}(z_0)$ exists where $\alpha = \text{Re } z_0$. The inversion of the Laplace transform, then, is obtained by the formula,

$$f(t) = \frac{1}{2\pi i} \lim_{M \to \infty} \int_{\Gamma_M} F(z) \, dz,$$

with $F(z) = e^{zt} \hat{f}(z)$, and $\Gamma_M$ is the straight line from $\alpha - Mi$ to $\alpha + Mi$ in the complex plane.

There are three types of commonly used numerical methods to invert Laplace transforms. The first and second methods evaluate the Bromwich's integrals along straight lines parallel to the imaginary axis by using the trapezoidal rule [1-4] and by using Laguerre polynomials [5-8], respectively. The third method applies the trapezoidal rule to the integration of Bromwich's
integral along a special contour devised by Talbot [9,10]. Duffy gave an excellent comparison of these three type of methods in [11], wherein the Talbot scheme [9,10] was shown to give more accurate results at less computational costs provided that the location of the singularities is known.

In this paper, we propose to use the information on the location of singular points more sharply, so that the computational cost can be reduced significantly. In fact, the error bound of our method which essentially is described by (1.5) below has spectral accuracy of type $e^N + \varepsilon P$, $0 < \varepsilon < 1$ (see Section 2). Thus, even one or two $f(z)$ values can recover $f(t)$ completely, which would be difficult to improve on by any other method. Furthermore, unlike Talbot's method [9,10], the method proposed in this paper is not seriously affected, even if there are infinite number of singular points whose imaginary parts go to infinity. Unfortunately, our scheme is restricted to the class of functions that do not have branch points.

For our purposes, we assume that $f(z)$ may be continued analytically to an analytic function, still denoted by $\tilde{f}(z)$, in the whole complex plane except singular points and it satisfies the condition (see [9,10]),

$$|\tilde{f}(z)| \to 0 \text{ uniformly in } \{z : \Re z \leq \alpha\} \text{ as } |z| \to \infty. \quad (1.2)$$

In that case, the contour $\Gamma_M$ in (1.1) will be replaced by a semicircle

$$\Gamma_M = \Gamma_{M,1} + \Gamma_{M,2}, \quad (1.3)$$

where

$$\Gamma_{M,1} = \{z : z = \alpha + iy, -M \leq y \leq M\} \quad \text{and}$$

$$\Gamma_{M,2} = \left\{z : z = \alpha + Me^{i\theta}, \frac{\pi}{2} \leq \theta \leq \frac{3\pi}{2}\right\},$$

with the diameter $M$ being sufficiently large to enclose most significant singularities. (Here, and in what follows, we will implicitly take the counterclockwise orientation.) Such a modification in the contour in (1.1) is justified since we have

$$\int_{\Gamma_{M,2}} e^{zt} \tilde{f}(z) \, dz \to 0, \quad \text{as } M \to \infty,$$

which can be proved by the condition (1.2) and the fact,

$$\int_{\pi/2}^{3\pi/2} e^{M \cos \theta} \, d\theta < \frac{\pi}{M} \quad (M > 0),$$

based on Jordan's inequality (see [12]).

Next, by using the Cauchy-Goursat theorem [12] with the deformed contour represented by (1.3) for sufficiently large $M$, $f(t)$ can be approximated by the finite sum of contour integrals in which the contours are small circles around the poles of $F(z)$. For example,

$$f(t) \approx \frac{1}{2\pi i} \int_{\Gamma_M} F(z) \, dz = \frac{1}{2\pi i} \sum_{k=1}^{l} \int_{C_k} F(z) \, dz, \quad \text{for some } l, \quad (1.4)$$

where the $C_k$ are positively-oriented circles around singular points $z_k$ of $F(z)$ and are sufficiently small that any two of them do not have points in common. $l$ is the number of poles enclosed by $\Gamma_M$. In particular, we choose $C_k$ centered at the isolated singular point $z_k$. Using the change
of variables \( z = \phi_k(\theta) = z_k + \varepsilon e^{i\theta} \), for sufficiently small \( \varepsilon > 0 \), and applying the trapezoidal rule to the integrals in the summation, we approximate (1.1) as follows,

\[
Q(t) = \frac{1}{2\pi i} \sum_{k=1}^{l} \left( \frac{2\pi}{N_k} \sum_{j=0}^{N_k-1} F(\phi_k(\theta_{k,j})) \phi'_k(\theta_{k,j}) \right)
\]

\[
= \sum_{k=1}^{l} \left( \frac{1}{N_k} \sum_{j=0}^{N_k-1} F(\phi_k(\theta_{k,j})) \varepsilon e^{i\theta_{k,j}} \right)
\]

(1.5)

with quadrature points \( \theta_{k,j} = (2\pi/N_k)j \in [0, 2\pi] \), since \( F(\phi_k(0))\phi'_k(0) = F(\phi_k(2\pi))\phi'_k(2\pi) \).

Here, \( N_k \) is the number of quadrature points to employ.

In Section 2, we will analyze the errors of this quadrature scheme. We will prove that the errors are sufficiently small, if a \( p \)-mesh-point quadrature rule is employed, for each singular point, where the order of poles is less than or equal to \( p \). Also, we discuss the propagation of round-off errors and provide the criteria for parameter selection. In Section 3, we compare our scheme with several other numerical schemes.

2. ERROR ESTIMATES

2.1. The basic error estimate

In this section, we analyze the approximation errors of the scheme defined in (1.5). If a singular point \( z_k \) of \( F(z) \) in (1.4) is isolated and is a pole of order \( \leq p_k \), then, \( F(z) \) has the Laurent series representation in a deleted neighborhood \( V_k \) of \( z_k \) in the form

\[
F(z) = \sum_{m=-p_k}^{\infty} c_{k,m}(z - z_k)^m, \quad \text{for} \, z \in V_k.
\]

(2.1)

Furthermore, Cauchy’s residue theorem implies that

\[
\frac{1}{2\pi i} \int_{C_k} F(z) \, dz = \text{Res}_{z=z_k} F(z) = c_{k,-1},
\]

where \( C_k \) is a circle centered at \( z_k \) of radius \( \varepsilon \). Our first result for computing the residue \( c_{k,-1} \) of \( F \) using the method (1.4) for a single \( k \) is as follows.

**Theorem 2.1.** Assume that \( z_k \) is an isolated singularity of \( F(z) \) and is a pole of order \( \leq p_k \). Let the number \( N_k \) of mesh points be larger than or equal to \( p_k \). Then,

\[
\left| \frac{1}{N_k} \sum_{j=0}^{N_k-1} F(\phi_k(\theta_{k,j})) \varepsilon e^{i\theta_{k,j}} - c_{k,-1} \right| = O\left( \varepsilon^{N_k} \right),
\]

where \( \phi_k(\theta_{k,j}) = z_k + \varepsilon e^{i\theta_{k,j}} \) and \( \theta_{k,j} = (2\pi/N_k)j, \, j = 0, 1, \ldots, N_k - 1 \).

**Proof.** By (2.1),

\[
\frac{1}{N_k} \sum_{j=0}^{N_k-1} F(\phi_k(\theta_{k,j})) \varepsilon e^{i\theta_{k,j}} = \frac{1}{N_k} \sum_{j=0}^{N_k-1} \sum_{m=-p_k}^{\infty} c_{k,m} (\varepsilon e^{i\theta_{k,j}})^m \varepsilon e^{i\theta_{k,j}}
\]

\[
= \frac{1}{N_k} \sum_{m=-p_k}^{\infty} c_{k,m} \varepsilon^{m+1} \left( \sum_{j=0}^{N_k-1} (\varepsilon e^{i\theta_{k,j}})^j \right).
\]

(2.2)
Here, the $\sum_j$ term in the last equation becomes $N_k$, if $e^{i\theta_k,m+1} = 1$; otherwise, it is equal to
\[
\frac{1 - (e^{i\theta_k,m+1})^{N_k}}{1 - e^{i\theta_k,m+1}},
\]
which is $0$. Moreover, since $e^{i\theta_k,m+1} = 1$ holds, when $m + 1$ is a multiple of $N_k$, the remaining terms of (2.2) are just
\[
\cdots + c_{k,-2N_k-1}e^{-2N_k} + c_{k,-N_k-1}e^{-N_k} + c_{k,-1} = c_{k,N_k-1}e^{N_k} + c_{k,2N_k-1}e^{2N_k} + \cdots.
\]
Since we have $N_k \geq p_k$ and $c_{k,m} = 0$, if $m < -p_k$, those terms which are to the left of $c_{k,-1}$ vanish. This completes the proof.

We remark that Theorem 2.1 implies that, in order to obtain a highly accurate inverse, we do not need to take a large $N_k$ which requires more computational cost; it is enough to have $N_k = p_k$ and take a smaller $\varepsilon$ which is independent of computational expense (c.f., Section 2.2). We also note that the method does not require the order of pole $p_k$ exactly; we can choose any $N_k$, such that $N_k \geq p_k$.

In order to consider truncation error, we take $M$ in (1.3) or (1.4) sufficiently large so that the difference between the first and second terms of (1.4) is less than $\delta > 0$. If we also take $N$ to be the smallest one among all $N_k$, then, we have the following theorem.

**Theorem 2.2.** Assume that $M$ given in (1.3) or (1.4) is large enough so that
\[
\left| f(t) - \frac{1}{2\pi i} \int_{\Gamma_M} F(z) \, dz \right| \leq \delta.
\]
Let $z_k$, $k = 1, \ldots, l$, be the poles of $F(z)$ enclosed by $\Gamma_M$, $N = \min\{N_k, k = 1, \ldots, l\}$ with $N_k \geq p_k$ where $p_k$ is the order of pole at $z_k$, and $Q(t)$ be the approximation of $f(t)$ defined by (1.5). Then,
\[
|f(t) - Q(t)| \leq \delta + O(\varepsilon^N), \tag{2.3}
\]
where $\varepsilon$ is the radius of circles around $z_k$'s as in Theorem 2.1.

In case the number of poles of $F(z)$ is finite, one can choose $M$ large enough, so that $\Gamma_M$ includes all poles with $\delta = 0$ in Theorem 2.2.

### 2.2. The propagation of roundoff errors

Since the method (1.5) involves the numerical integration around singularities, the effect of round-off errors should be considered in more detail. In this section, we investigate in the round-off error terms of the order of the poles, the number of quadrature points, the radius of circles around the poles, and the machine precision. We will also provide some criteria for the choice of $N$ and $\varepsilon$.

Suppose $F(z)$ is of the form as in (2.1),
\[
F(z) = \sum_{m=-p_k}^{\infty} c_{k,m} (z - z_k)^m, \quad \text{for } z \in V_k.
\]
In some cases, for instance if $|z - z_k|$ is very small, the propagation of roundoff errors in the quantity $z - z_k$ may be very erroneous and harmful [13]. To cope with the roundoff error, let $\tilde{z} = z + \chi$ with $|\chi| \leq K \cdot \varepsilon$, where $\varepsilon$ denotes the numerical precision with $K$ being a small positive integer. As a first-order approximation, we have
\[
F(z) - F(\tilde{z}) \approx \frac{dF(z)}{dz} (z - \tilde{z}) = - \sum_{m=-p_k}^{\infty} mc_{k,m} (z - z_k)^{m-1} \chi.
\]
In method (1.5), recalling that \( z = z_k + \varepsilon e^{i\theta} \), we have

\[
|F(z) - F(\tilde{z})| \leq \sum_{m=-p_k}^{\infty} |m_{ck,m}| \varepsilon^{m-1} K \cdot \text{eps} = O\left( \frac{\text{eps}}{\varepsilon^{p_k+1}} \right),
\]

where we observe that \( \varepsilon \) should not be too small compared to the numerical precision \( \text{eps} \).

Keeping this observation in mind, if we take into account of round-off errors, \( Q(t) \) in (1.5) can be approximated by

\[
Q_{\text{eps}}(t) = \sum_{k=1}^{l} \left( \frac{1}{N_k} \sum_{j=0}^{N_k-1} F(\phi_k(\theta_{k,j}) + x_{k,j}) e^{i\theta_{k,j}} \right),
\]

with \( |x_{k,j}| \leq K \cdot \text{eps} \). Then, assuming the supremum \( P := \sup_k \{ p_k \} \) exists and letting \( \beta = \max_{1 \leq k \leq l} \max_{0 \leq \theta \leq 2\pi} \text{Re}(\phi_k(\theta)) \), we have by (2.4),

\[
|Q(t) - Q_{\text{eps}}(t)| \leq O\left( \frac{\text{eps}}{\varepsilon^P} \right).
\]

This gives an asymptotic order of magnitude of error which may not be improved. The total error in (2.3), then, is replaced by

\[
|f(t) - Q_{\text{eps}}(t)| \leq \delta + O\left( \varepsilon^N + \frac{\text{eps}}{\varepsilon^P} \right). \tag{2.5}
\]

We summarise the above result in the following theorem.

**Theorem 2.3.** Suppose the same assumption as in Theorem 2.2. Moreover, assume the supremum \( P := \sup_k \{ p_k \} \) exists and suppose that \( |z_k - z_k'| \leq K \cdot \text{eps} \), for \( K > 0 \) and machine precision \( \text{eps} \). Suppose \( Q_{\text{eps}}(t) \) is the approximation of \( f(t) \) defined by (1.5) with the roundoff error considered. Then, the following estimate holds

\[
|f(t) - Q_{\text{eps}}(t)| \leq \delta + C \left( \varepsilon^N + \frac{\text{eps}}{\varepsilon^P} \right),
\]

for some positive constant \( C \).

Motivated by (2.5), we describe a method of optimal choice of \( \varepsilon \) and \( N \) assuming that \( N = N_k \), \( k = 1, \ldots, l \). For a fixed \( N \), we should have

\[
\varepsilon^N + \frac{\text{eps}}{\varepsilon^P} \geq 2 \varepsilon^{N/(N+P)}, \quad \text{for all } \varepsilon > 0,
\]

with the equality holds for \( \varepsilon = \varepsilon^{1/(N+P)} \). Thus, the total error in (2.5) is minimized asymptotically if we choose \( \varepsilon = \varepsilon^{1/(N+P)} \). Moreover the error bound tends to \( \delta + C \cdot \text{eps} \) as \( N \) tends to \( \infty \).

In order to examine the above argument numerically, we begin with the following example. Let \( f(t) = t^{p-1}e^t \) whose Laplace transform is \( 1/(z - 1)^p \). Choose \( N = p, p = 1, 2 \), and apply the method (1.5) with double precision calculation, for which \( \text{eps} \approx 10^{-16} \). The results are illustrated in Table 1, where one can see that the errors are minimized with \( \varepsilon = \varepsilon^{1/(N+P)} \), for both \( p = 1 \) and \( p = 2 \). Next, turn to see if the error tends to \( \delta + C \cdot \text{eps} \) as \( N \) increases. For this, we fix \( \varepsilon = \varepsilon^{1/(N+P)} \) and \( p = 1 \) and increase \( N \) and the computational results are given in Table 2. Clearly, the total error (2.5) decreases to the machine precision \( \text{eps} \) as \( N \) increases.

<table>
<thead>
<tr>
<th>( p )</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10^{-2}</td>
</tr>
<tr>
<td></td>
<td>8.3 (-3)</td>
</tr>
<tr>
<td>2</td>
<td>3.4 (-6)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>( N )</th>
<th>( \varepsilon )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Error</td>
<td>1.8 (-8)</td>
</tr>
</tbody>
</table>
3. NUMERICAL EXAMPLES

In this section, we give several numerical examples in order to illustrate and compare our proposed method (1.5) with Talbot’s, Durbin’s, and Tzou’s methods. All computations were carried out in double precision on a PC based on a Pentium III processor.

Let $E$ denote the root-mean-square deviation error between analytical and numerical solutions, defined by

$$
E = \sqrt{\frac{1}{100} \sum_{j=1}^{100} \left[ f(t_j) - Q_{\epsilon_{\text{eps}}}(t_j) \right]^2 / 100},
$$

(3.1)

where $t_j = j - 1/2$, $j = 1, \ldots, 100$. The error $E$ depends on the radius and the number of mesh points near the singular points in (1.5), and we denote by $E(\epsilon)$ the error for fixed $N$ in each row in error tables. Also $\rho_\epsilon$ denotes the reduction order $\log_{10}(E(\epsilon)/E(\epsilon/10))$ for various values of parameters mentioned above. In the following error tables, $a(b)$ denotes the floating number $a \cdot 10^b$. For example, $1.3(-2)$ means $1.3 \times 10^{-2}$.

EXAMPLE 1. As our first example, we consider the function $\hat{f}(z) = 1/(z + 1)$, which is the Laplace transform of $f(t) = e^{-t}$. Table 3 shows errors between the computed and exact solutions defined in (3.1) and its dependence on $\epsilon$ and $N$.

<table>
<thead>
<tr>
<th>$N$</th>
<th>$E(1.0(-1))$</th>
<th>$E(1.0(-2))$</th>
<th>$E(1.0(-3))$</th>
<th>$E(1.0(-4))$</th>
<th>$E(1.0(-5))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.5 (-3)</td>
<td>5.1 (-4)</td>
<td>5.1 (-5)</td>
<td>5.1 (-6)</td>
<td>5.1 (-7)</td>
</tr>
<tr>
<td>2</td>
<td>4.3 (-4)</td>
<td>4.3 (-5)</td>
<td>4.3 (-6)</td>
<td>4.3 (-7)</td>
<td>4.3 (-8)</td>
</tr>
<tr>
<td>3</td>
<td>4.0 (-5)</td>
<td>4.0 (-6)</td>
<td>4.0 (-7)</td>
<td>4.0 (-8)</td>
<td>4.0 (-9)</td>
</tr>
<tr>
<td>4</td>
<td>3.7 (-6)</td>
<td>3.7 (-7)</td>
<td>3.7 (-8)</td>
<td>3.7 (-9)</td>
<td>3.7 (-10)</td>
</tr>
</tbody>
</table>

The numerical results in Table 3 confirm the convergence behavior predicted by Theorem 2.1 rather than that predicted by Theorem 2.3, in that the apparent rate of convergence $\rho_\epsilon$ with variable $\epsilon$ is almost exactly equal to $N$. We also observe that even a single mesh point gives satisfactory results as predicted by the remark of Theorem 2.1. It would be difficult to improve this result by any other method. For $N = 4$, the values $E(1.0(-4)) = 3.6(-15)$ and $\rho_{1.0(-3)} = 1.01$ appear to be the worst, but this is due to limitations of double precision calculation.

Notice that the bound in Theorem 2.3 is actually given in the form

$$
|f(t) - Q_{\epsilon_{\text{eps}}}(t)| \leq \delta + C_1 \epsilon^N + C_2 \epsilon_{\text{eps}}^{1/P},
$$

where $C_1$ and $C_2$ depend on the variable $t$. We infer that in this example the terms $C_1(t_j)\epsilon^N$ dominate $C_2(t_j)\epsilon_{\text{eps}}^{1/P}$ for most of $t_j$.

EXAMPLE 2. Next, we apply several methods for each of the test functions listed in Table 4. Some functions in Table 4 were used by Davies and Martin [14]. We tested modified Durbin’s method [15], Tzou’s method [15,16], Talbot’s method [9], and our proposed method (1.5). The parameters used for Durbin’s and Tzou’s methods are the same as those in [15], while those for Talbot’s method are provided by the subroutine TAPAR [17]. Numerical results are summarized in Table 5. For Durbin’s and Tzou’s methods 1000 terms were used in its evaluation [15], for Talbot’s method 10–20 mesh points, and only 2–3 mesh points are used for method (1.5). Consequently, the evaluation times $T$ (in 0.1 millisecond) are smallest for method (1.5).

From this table, we see that our method is very efficient when $\hat{f}(z)$ has a finite number of poles. In the case of having an infinite number of poles will be discussed in subsequent examples.

In the test example $f_8(t)$, the function tends to infinity as $t$ increases, and thus, we used the relative errors which are defined similarly to (3.1) replacing $f(t_j) - Q_{\epsilon_{\text{eps}}}(t_j)$ by $(f(t_j) - Q_{\epsilon_{\text{eps}}}(t_j))/f(t_j)$. 

<table>
<thead>
<tr>
<th>$N$</th>
<th>$E(1.0(-1))$</th>
<th>$E(1.0(-2))$</th>
<th>$E(1.0(-3))$</th>
<th>$E(1.0(-4))$</th>
<th>$E(1.0(-5))$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.5 (-3)</td>
<td>5.1 (-4)</td>
<td>5.1 (-5)</td>
<td>5.1 (-6)</td>
<td>5.1 (-7)</td>
</tr>
<tr>
<td>2</td>
<td>4.3 (-4)</td>
<td>4.3 (-5)</td>
<td>4.3 (-6)</td>
<td>4.3 (-7)</td>
<td>4.3 (-8)</td>
</tr>
<tr>
<td>3</td>
<td>4.0 (-5)</td>
<td>4.0 (-6)</td>
<td>4.0 (-7)</td>
<td>4.0 (-8)</td>
<td>4.0 (-9)</td>
</tr>
<tr>
<td>4</td>
<td>3.7 (-6)</td>
<td>3.7 (-7)</td>
<td>3.7 (-8)</td>
<td>3.7 (-9)</td>
<td>3.7 (-10)</td>
</tr>
</tbody>
</table>
Table 4. Functions used in testing various methods.

<table>
<thead>
<tr>
<th>Function</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f_1(z) = (z + 1/2)^{-1} )</td>
<td>( f_1(t) = e^{-t/2} )</td>
</tr>
<tr>
<td>( f_2(z) = ((z + 0.2)^2 + 1)^{-1} )</td>
<td>( f_2(t) = e^{-0.2t} \sin(t) )</td>
</tr>
<tr>
<td>( f_3(z) = z^{-1} )</td>
<td>( f_3(t) = 1 )</td>
</tr>
<tr>
<td>( f_4(z) = z^{-2} )</td>
<td>( f_4(t) = t )</td>
</tr>
<tr>
<td>( f_5(z) = (z + 1)^{-2} )</td>
<td>( f_5(t) = te^{-t} )</td>
</tr>
<tr>
<td>( f_6(z) = (z^2 + 1)^{-1} )</td>
<td>( f_6(t) = \sin(t) )</td>
</tr>
<tr>
<td>( f_7(z) = (z^2 - 1)(z^2 + 1)^{-2} )</td>
<td>( f_7(t) = t \cos(t) )</td>
</tr>
<tr>
<td>( f_8(z) = z^2(z^3 + 8)^{-1} )</td>
<td>( f_8(t) = (e^{-2t} + 2e^t \cos(\sqrt{3}t))/3 )</td>
</tr>
</tbody>
</table>

Table 5. Summary of the comparison of apparent errors \( E \), and the evaluation times \( \tau \) (in 0.1 millisecond) of several methods. \( \epsilon \) is the parameter we used. Also, we took \( N = 2 \) except test 5, of which \( N = 3 \).

<table>
<thead>
<tr>
<th>Test</th>
<th>Durbin</th>
<th>Tzou</th>
<th>Talbot</th>
<th>Method (1.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( E )</td>
<td>( \tau )</td>
<td>( E )</td>
<td>( \tau )</td>
</tr>
<tr>
<td>1</td>
<td>9.7 ((-4))</td>
<td>1695</td>
<td>1.8 ((-4))</td>
<td>1851</td>
</tr>
<tr>
<td>2</td>
<td>2.9 ((-4))</td>
<td>1764</td>
<td>3.2 ((-4))</td>
<td>1218</td>
</tr>
<tr>
<td>3</td>
<td>9.8 ((-4))</td>
<td>1639</td>
<td>1.1 ((-4))</td>
<td>1175</td>
</tr>
<tr>
<td>4</td>
<td>2.1 ((-2))</td>
<td>1712</td>
<td>1.4 ((-2))</td>
<td>1203</td>
</tr>
<tr>
<td>5</td>
<td>2.9 ((-4))</td>
<td>1724</td>
<td>3.2 ((-4))</td>
<td>1234</td>
</tr>
<tr>
<td>6</td>
<td>2.9 ((-4))</td>
<td>1727</td>
<td>3.3 ((-4))</td>
<td>1212</td>
</tr>
<tr>
<td>7</td>
<td>1.5 ((-2))</td>
<td>1801</td>
<td>1.0 ((-2))</td>
<td>1244</td>
</tr>
<tr>
<td>8</td>
<td>9.9 ((-1))</td>
<td>1811</td>
<td>9.8 ((-1))</td>
<td>1270</td>
</tr>
</tbody>
</table>

Remark 3.1. We emphasize that the error values in Table 3 and Table 5 are not the errors at a single time, but the errors defined by (3.1), which measure errors over \( t = 1, \ldots, 100 \).

Example 3. We now consider the case of \( f(z) = 1/(z(1 + e^{-\pi z})), \) which has an infinite number of poles whose imaginary parts increase without bound. Notice that Talbot's method is not applicable directly to this kind of problems. Its inversion is the well-known square wave function with discontinuities at \( t = n\pi, n = 1, 2, \ldots \), namely, \( f(t) = \sum_{n=0}^{\infty} (-1)^n U(t - n\pi) \), where \( U(t - a) \) is the unit step function at \( t = a \).

Many popular methods fail to give a satisfactory result for this function [11], since they must resolve an infinite number of singularities and these poles lie along the imaginary axis, leading to sinusoidal solutions. In Table 6, we summarize the errors between the computed and exact solutions for typical time values using Durbin’s and Tzou’s methods and the proposed method (1.5). For the numerical computation, 1000 terms are used for Durin’s and Tzou’s methods, and 1000 poles are included in \( \Gamma_M \) for method (1.5). From the errors in Table 6, we conclude that our method (1.5) gives similar results as Durbin’s and Tzou’s methods. In this case, our method loses its advantage since the number of included poles are at least several hundreds.

Example 4. Finally, we give an application to solving partial differential equations. In this case, although the number of poles to be considered are infinite, only a few poles are dominant so that our method is relatively efficient.

Table 6. Errors for Example 3. The parameters \( \epsilon = 1.0(-3) \) and \( N = 2 \) are used.

<table>
<thead>
<tr>
<th>( t )</th>
<th>( f(t) )</th>
<th>Durbin</th>
<th>Tzou</th>
<th>Method (1.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \pi/2 )</td>
<td>1 ((-3))</td>
<td>1.6 ((-3))</td>
<td>2.6 ((-5))</td>
<td>1.6 ((-4))</td>
</tr>
<tr>
<td>( \pi )</td>
<td>1 ((-2))</td>
<td>3.3 ((-4))</td>
<td>5.4 ((-4))</td>
<td>1.2 ((-6))</td>
</tr>
<tr>
<td>( 3\pi/2 )</td>
<td>0 ((-3))</td>
<td>1.3 ((-3))</td>
<td>1.1 ((-4))</td>
<td>1.6 ((-4))</td>
</tr>
</tbody>
</table>
Consider the one-dimensional heat equation,

\[ u_t = u_{xx}, \quad x \in [0, \pi], \quad (3.2a) \]
\[ u(0, t) = u(\pi, t) = 0, \quad \text{for } t > 0, \quad (3.2b) \]
\[ u(x, 0) = u_0(x) = \begin{cases} x, & 0 \leq x \leq \frac{\pi}{2}, \\ \pi - x, & \frac{\pi}{2} < x \leq \pi. \end{cases} \quad (3.2c) \]

Taking the Laplace transformation of the above equation with respect to time, we have

\[ z\hat{u} - \hat{u}_{xx} = u_0, \quad (3.3) \]
\[ \hat{u}(0, z) = \hat{u}(\pi, z) = 0, \]

where \( z \) is the variable of the Laplace transform. Then, the solution is given by the Laplace inversion formula (see [18–20]),

\[ u(x, t) = \frac{1}{2\pi i} \int e^{zt} \hat{u}(x, z) \, dz, \quad \text{for } t > 0, \quad (3.4) \]

where \( \hat{u}(x, z) = (zI - \partial_x^2)^{-1}u_0(x) \), and \( I \) and \( \partial_x \) denote the identity operator and the partial differential operator with respect to \( x \). In practice, (3.3) for fixed \( z \) must be solved by using space discretization methods such as finite element, finite difference method, etc.

The solution of (3.2) is given by

\[ u(x, t) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} e^{-(2n-1)^2t} \sin(2n-1)x}{(2n-1)^2}, \quad (3.5) \]

and thus, the solution of (3.3) is given in the form

\[ \hat{u}(x, z) = \sum_{n=1}^{\infty} \frac{(-1)^{n+1} \sin(2n-1)x z + (2n-1)^2}{(2n-1)^2}, \quad (3.6) \]

We apply Durbin’s and Tzou’s methods, and the method (1.5) to (3.4) with (3.6) taken only the first 100 terms. Resulting \( L^2(0, \pi) \)-errors between the computed and exact solutions (3.5) are summarized in Table 7. We take the first 100 terms in the summation (3.5) as the reference solution. In method (1.5), we have \( \varepsilon = 10^{-6} \), \( N = 1 \), and \( \Gamma_M \) contained only one pole \( z = -1 \). Since solution (3.5) converges exponentially fast, the effect of excluded poles is quite negligible. (It is less than \( 1.2 \times 10^{-40} \).)

Table 7. Summary of \( L^2(0, \pi) \)-errors, and the evaluation times \( \tau \) (in 0.1 millisecond) of several methods.

<table>
<thead>
<tr>
<th>Time</th>
<th>Durbin</th>
<th>Talbot</th>
<th>Method (1.5)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( E )</td>
<td>( \tau )</td>
<td>( E )</td>
</tr>
<tr>
<td>10</td>
<td>4.8 (-3)</td>
<td>23527</td>
<td>1.8 (-4)</td>
</tr>
<tr>
<td>15</td>
<td>6.8 (-3)</td>
<td>23522</td>
<td>2.3 (-4)</td>
</tr>
<tr>
<td>20</td>
<td>4.0 (-4)</td>
<td>23521</td>
<td>2.9 (-4)</td>
</tr>
</tbody>
</table>

In this application, the amount of the calculation is vast since numerical inversion of Laplace transforms should be performed, for each fixed space variable. In Durbin’s and Tzou’s methods, 1000 \( \hat{u}(x, z) \) values are used, for each \( x \). On the other hand, in the method (1.5) only one \( \hat{u}(x, z) \) value is used for each \( x \), resulting in great efficiency as shown in Table 7.

**REMARK 3.2.** For complicated functions the poles are not given analytically, the location of poles must be found numerically. See [21] and the references therein. After finding the poles...
numerically, we can apply (1.5). A theoretical investigation of the errors that will occur in the approximation of poles while using our method is our next goal.

REFERENCES