AN ABSTRACT OF THE DISSERTATION OF

FuSen F. Lin for the degree of Doctor of Philosophy in Mathematics presented on September 3, 2003.

Title: Numerical Inversion of Laplace Transforms By the Trapezoidal-Type Methods

Redacted for privacy

Abstract approved: _____

Larry Chen

In this dissertation, we investigate three numerical methods for inverting the Laplace transform. These methods are all based on the trapezoidal-type approximations to the Bromwich integral. The first method is the direct integration method: It is a straightforward application of the trapezoidal rule to the Bromwich integral, followed by convergence acceleration techniques to sum efficiently the infinite series that arises. We identify and analyze three sources of error associated with this method, namely the discretization, truncation, and conditioning error. An integral representation of the discretization error is derived and the truncation and conditioning error are also estimated. The method contains a free parameter a (in fact, the position of Bromwich line) that can be adjusted to maximize the accuracy. We present both theoretical formulas and algorithmic techniques for selecting the optimal value of a. The second method we investigate owes to Talbot. It is likewise based on the trapezoidal-type approximation of the Bromwich integral,

but uses a deformed contour. We derive a formula for the discretization error associated with this method. Based on this, we propose an algorithm for the optimal selection of the free parameters contained in Talbot's method. The third method we believe to be new. It is based on Ooura and Mori's so-called double exponential formulas for integrals of Fourier-type that we have adapted to the Laplace inversion problem. Throughout the thesis, we test our theoretical formulas and practical algorithms on a wide range of transforms, many of which are taken from the engineering literature.

Numerical Inversion of Laplace Transforms By the Trapezoidal-Type Methods

by

FuSen F. Lin

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Completed September 3, 2003
Commencement June 2004

Doctor of Philosophy dissertation of FuSen F. Lin presented on September 3, 2003

APPROVED:

Redacted for privacy

Major Professor, representing Mathematics

Redacted for privacy

Chair of the Department of Mathematics

Redacted for privacy

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Redacted for privacy

FuSen F. Lin, Author

ACKNOWLEDGEMENTS

This work was supported by National Taiwan Ocean University for one year under the authorization of its former president, Dr. YanPing Shi. Unfortunately, he had no chance to see this dissertation since he passed away in 1998. The author sincerely presents this work in memory of Dr. YanPing Shi.

Next, I wish to thank many professors and friends who have provided suggestions for me. Particularly, I sincerely express my profoundest appreciation to Professor J.A.C. (Andre) Weideman, who advised me to complete this dissertation. Andre was my original major professor at Oregon State University; however, he moved back to South Africa at the end of 1998. Since then, he has worked at the University of Stellenbosch, South Africa. Even though he left OSU, he still continued to supervise me in finishing this dissertation. His conscientious and responsible actions towards me indicate that he is a successful educator.

I gratefully thank the Department of Applied Mathematics at University of Stellenbosch, which I twice visited, finishing many parts of this work. During my visits, the Department offered me computer facilities, accommodation, and much assistance in the working environment. Regarding the visit in year 2000, I am thankful to the former chair of Mathematics Department, John W. Lee, and the former dean of International Programs, Jack Van De Water, who offered me financial support.

Among others, I would like to express special thanks to Professors Bent Petersen, Adel Faridani, and Larry Chen. Bent Petersen gave me a reference on the Lambert W function and many discussions on complex integration and the Laplace transformation. Adel Faridani helped me to understand the FORTRAN software. Larry Chen kindly succeeded my major professor after Andre left. His many suggestions were most valuable.

In addition, I gratefully acknowledge Luisa D'Amore, who offered me the FORTRAN codes of their work. Last but not least, I was deeply indebted to my family, who have endured long terms of my absence. Especially, my wife, Maggie, who has helped with typing and remained supportive of me during the long process in completing this work.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

Numerical Inversion of Laplace Transforms By the Trapezoidal-Type Methods

1. INTRODUCTION AND BACKGROUND THEORY

In this chapter, we shall introduce numerical methods for inverting the Laplace transform. Also, elements of the fundamental theory of the Laplace transformation, some properties of the trapezoidal-type integration rules, and the convergence accelerators will be addressed. This background will be used throughout this dissertation.

1.1. Introduction

The technique of the Laplace transformation plays a significant role in mathematical applications in physics and engineering. It is one of the most powerful tools for solving differential equations and integral equations. When applying this technique, however, finding the inversion of a given transform is a critical problem. Unless the transform is given in a table, we encounter the evaluation of the Bromwich integral

$$f(t) = \frac{1}{2\pi i} \text{PV} \int_{a-i\infty}^{a+i\infty} e^{st} F(s)\, ds, \quad \text{for} \quad t \geq 0, \;\; a > \sigma_0, \tag{1.1}$$

in the complex plane, which is the inversion formula for the Laplace transform

$$F(s) = \int_0^{\infty} e^{-st} f(t)\, dt, \quad \text{for} \quad \text{Re}(s) > \sigma_0. \tag{1.2}$$

Here, σ_0 is the convergence abscissa of the Laplace transform, to be defined in Section 1.2, and PV denotes the Cauchy principal value. For brevity, we shall omit the notation PV hereafter. When all complex integration techniques fail to evaluate the Bromwich integral analytically, one has to rely on numerical methods.

For numerical work, the path of integration is typically parameterized by $s = a+iy$, which leads to

$$f(t) = \frac{e^{at}}{2\pi} \int_{-\infty}^{\infty} e^{ity} F(a+iy)\, dy. \tag{1.3}$$

The numerical approximation of this integral is challenging for a number of reasons: First, it is posed on the infinite line with the integrand highly oscillatory as $y \to \pm\infty$. In addition, the transform $F(a+iy)$ may decay slowly as $y \to \pm\infty$.

There exist many numerical methods for dealing with these difficulties. Davies and Martin gave an authoritative survey and did general testing on these methods in 1979 [1] . A more recent benchmark test focusing on three of the most efficient methods was presented by Duffy in 1993 [2]. These methods are the Weeks method [3], the Talbot method [4], and the so-called direct integration method [5, 6, 7], which is the trapezoidal rule plus certain enhancement techniques. In this dissertation, we focus on the direct integration method and Talbot's method, and also try to create a new method. All three methods are based on the variation of trapezoidal-rule approximation of the Bromwich integral (1.1). We shall introduce them in more details as follows.

The first method we shall consider is the direct integration method (also called the Fourier series method), which was first presented by Dubner and Abate in 1968 [5]. They derived the numerical inversion formula and analyzed the error by using a Fourier cosine series expansion. Later, Durbin combined the Fourier cosine and sine series to reduce the discretization error (i.e., the error introduced by replacing the integral with the sum) [6], thereby improving the method. Durbin's method may be summarized by the formula:

$$\widetilde{f}(t) = \frac{e^{at}}{T} \sum_{n=0}^{\infty} \left[\mathrm{Re}\left\{ F\left(a + \tfrac{n\pi i}{T}\right)\right\} \cos\left(\tfrac{n\pi t}{T}\right) - \mathrm{Im}\left\{ F\left(a + \tfrac{n\pi t}{T}\right)\right\} \sin\left(\tfrac{n\pi t}{T}\right)\right] \tag{1.4}$$

for $t \in [0,\, T/2]$, where $T > 0$ and $a \in \mathbb{R}$, $a > \sigma_0$. It has been noted that this technique is nothing more than the trapezoidal rule applied to the Bromwich integral (1.1) with step size $h = \pi/T$, more details are given in Section 2.1. This method is therefore called the direct integration method, named by Duffy [2].

Although this technique is easy to implement, it may converge slowly[1]. For these reasons, some enhancement techniques, such as convergence accelerators and correction terms, were proposed by Crump [8], and Honig and Hirdes [9], respectively. The idea of choosing $T = 2t$ was mentioned by Dubner and Abate [5] and implemented by Duffy [2]. With this choice, the sine and cosine factors disappear from (1.4), and one obtains

$$\widetilde{f}(t) = \left(\frac{e^{at}}{2t}\right)\left\{\frac{\mathrm{Re}\{F(a)\}}{2} + \sum_{n=1}^{\infty}(-1)^n\left[\mathrm{Re}\left\{F\left(a+\frac{n\pi i}{t}\right)\right\} + \mathrm{Im}\left\{F\left(a+\frac{(n-1/2)\pi i}{t}\right)\right\}\right]\right\}. \quad (1.5)$$

The parameters we need to focus on are the position of the Bromwich line, a, and the number of evaluation terms, N, say.

The series in (1.5) is suitable for acceleration of convergence. In Duffy's tests, Levin's u-transformation and the ε-algorithm were employed for sequence acceleration, and the author obtained high accuracy. However, the accuracy of this method depends significantly on the parameters a and N. The optimal selection of the parameter a is an important problem, and will be addressed in this thesis.

A recent implementation of this method, based on Durbin's formula (1.4), was presented in 1999 by D'Amore, Laccetti, and Murli. [10, 7]. Their approach involved estimating the discretization error theoretically and experimentally to find a suitable parameter a and then applying the q-d algorithm to accelerate convergence to reduce the truncation error (i.e., the error which occurs in truncating the infinite sum). Their algorithm does improve the efficiency of this method for some range of transforms. However, their estimate of the discretization error can still be improved, as our results will show in Chapter 2.

In Chapter 2, we shall explore both theoretical formulas as well as a numerical algorithm for selecting the optimal parameter a in order to achieve the highest accuracy.

[1] In Durbin's paper, several thousands of terms have been used to evaluate the test problems, and the author achieved only modest accuracy.

Also, we shall create an integral representation for the discretization error of the direct integration method that can be used for computing the approximations of the error. The truncation error (after convergence acceleration) can also be estimated both numerically and theoretically. Moreover, we take into account the roundoff error, which may play a significant role in the total error.

The second method we shall consider is Talbot's method, which is also based on the trapezoidal-type rules to approximate the integral (1.1). The path of integration we employ is along a special contour as prescribed by Talbot [4]. This new contour, obtained by deforming the Bromwich line, starts and ends in the left half-plane so that the series converges rapidly as $\mathrm{Re}(s) \to -\infty$, due to the exponential factor e^{st}. A FORTRAN implementation of this scheme was developed by Murli and Rizzardi [11]. Recently, a refined approach in approximating several values of $f(t)$, using the same sampling values of the Laplace transform $F(s)$, has been proposed by Rizzardi [12].

The accuracy of Talbot's method depends significantly on three geometric parameters that govern the shape of the deformed contour, as well as the number of evaluation points in the trapezoidal-type rules. Talbot has prescribed general formulas for selecting these parameters, based on geometric and empirical analyses. The parameters selected by Talbot's formulas may therefore not be truly optimal.

In Chapter 3, we propose, however, a new algorithm for optimal parameter selection in Talbot's method since we have found a new error formula that can be computed efficiently. This new approach improves the accuracy of Talbot's method.

Besides improving the direct integration method and Talbot's method, we shall also present a new method, also based on the trapezoidal-type rules. It is obtained by applying the double exponential formulas for Fourier-type integrals to the Bromwich integral (1.1) [13, 14]. This integral is essentially also of Fourier-type. This technique will be addressed in Chapter 4.

1.2. Some Properties of the Laplace Transform

In this section, we state some fundamental properties of the Laplace transform that will be referred to in this dissertation. The first theorem concerns the domain of convergence of the Laplace transform.

Theorem 1..1. *The exact domain of convergence of Laplace transform (1.2) is the right half-plane:* $\mathrm{Re}(s) > \sigma_0$ *possibly including none, part, or all of the line,* $\mathrm{Re}(s) = \sigma_0$*; admitting the possibilities* $\sigma_0 = \pm\infty$.

Proof: For a proof, we refer to [15, p.17].

The number σ_0 is called *the abscissa of convergence* of the Laplace transform; the open half-plane $\mathrm{Re}(s) > \sigma_0$ is referred to as the *half-plane of convergence* of the Laplace transform; the vertical line $\mathrm{Re}(s) = \sigma_0$ is called the *line of convergence.* The analyticity of the Laplace transform is stated as follows.

Theorem 1..2. *A Laplace transform (1.2) is an analytic function in the interior of its half-plane of convergence,* $\mathrm{Re}(s) > \sigma_0$.

Proof: For a proof, we refer to [15, p.26].

We next state an important property that will be used in the proof of our main theorem in Chapter 2.

Theorem 1..3. *If* $F(s)$*, the Laplace transform of* $f(t)$ *(1.2), converges absolutely for* $\mathrm{Re}(s) \geq x_0 > \sigma_0$*, then*

$$F(s) \longrightarrow 0, \quad as \ |s| \longrightarrow \infty, \quad for \ \ \mathrm{Re}(s) \geq x_0. \tag{1.6}$$

In particular, one can select sufficiently large values X *and* Y *so that, for all* s *with* $\mathrm{Re}(s) \geq X$ *or for all* s *with* $|\mathrm{Im}(s)| \geq Y$, $|F(s)|$ *is arbitrarily small.*

Proof: For a proof, we refer to [15, p.146].

The functional behavior of an original function $f(t)$ as $t \to \infty$, is reflected in the behavior of the corresponding image function $F(s)$ near some finite point s_0. If s_0 is in the

interior of the half-plane of convergence of $F(s)$, where $F(s)$ is analytic, then the behavior of $F(s)$ is trivial in the sense that $F(s)$ is continuous at s_0, and $F(s) \to F(s_0)$ as $s \to s_0$ [15]. Therefore, our main focus is on singular points s_0 on the line of convergence. This will be stated in the next theorem.

Theorem 1..4. *Suppose that the real-valued function $f(t)$ has the asymptotic property*

$$f(t) \sim ct^m \quad as \quad t \longrightarrow \infty, \quad m > -1, \tag{1.7}$$

where c is a constant. Then its Laplace transform $F(s)$ exists for $Re(s) > 0$. It has, for $c \neq 0$, a singular point at $s = 0$ and can be asymptotically represented as

$$F(s) \sim c\frac{\Gamma(m+1)}{s^{m+1}} \quad as \ s \longrightarrow 0, \tag{1.8}$$

where s is in the angular region $|\arg(s)| < \pi/2$, and Γ denotes the classical Gamma function.

Proof: For a proof, we refer to [15, p.231].

For further properties of the Laplace transform and its inversion, we refer to the books by Doetsch [15] and Widder [16].

1.3. Convergence Accelerators

A convergence accelerator is a numerical method for transforming a series so that the new series may converge faster than the original [17, 18, 19].

1.3.1. Euler's Series Transformation

First, we introduce the classical Euler transformation of series, which is efficient in summing the class of slowly convergent alternating series [20, p. 244]. Since Euler's transformation is linear, it is easy to analyze the remainder of the transformed series, particularly for the class of series with the property that their associated sequences are completely monotonic (defined later).

Let $\{a_n\}_{n=1}^{\infty}$ be a sequence. We first define the notation of the calculus of finite differences as follows:

$$\begin{aligned} \Delta^0 a_k &= a_k, \\ \Delta^n a_k &= \Delta^{n-1} a_k - \Delta^{n-1} a_{k+1}, \quad \text{for all } n, k = 1, 2, \ldots. \end{aligned} \tag{1.9}$$

Now let

$$\sum_{n=1}^{\infty} (-1)^{n-1} a_n = a_1 - a_2 + a_3 - \cdots, \tag{1.10}$$

be an arbitrary convergent series with sum S. We shall transform this series into a new series by using a simple technique based on the work of Grosjean [21]. This method is the so-called Euler's transformation of series. We write the series (1.10) as

$$S = a_1 - a_2 + a_3 - a_4 + \cdots,$$

multiply by 2 on both sides, and then rearrange it as

$$\begin{aligned} 2S &= a_1 + (a_1 - a_2) - (a_2 - a_3) + (a_3 - a_4) - \cdots \\ &= a_1 + \{\Delta a_1 - \Delta a_2 + \Delta a_3 - \cdots\}. \end{aligned}$$

After dividing by 2, we have

$$S = \frac{a_1}{2} + \frac{1}{2} \sum_{k=1}^{\infty} (-1)^{k-1} \Delta a_k = \frac{a_1}{2} + R_1. \tag{1.11}$$

Again, we multiply by 2^2 on both sides of the equation (1.11) and then rearrange the portion of the remainder R_1 as

$$\begin{aligned} 2^2 S &= 2\,a_1 + \{\Delta a_1 + (\Delta a_1 - \Delta a_2) - (\Delta a_2 - \Delta a_3) + \cdots\} \\ &= 2a_1 + \Delta a_1 + \{\Delta^2 a_1 - \Delta^2 a_2 + \cdots\}. \end{aligned}$$

Dividing by 2^2 again on both sides of the equation above, we get

$$S = \frac{a_1}{2} + \frac{\Delta a_1}{2^2} + \frac{1}{2^2} \sum_{k=1}^{\infty} (-1)^{k-1} \Delta^2 a_k = \frac{a_1}{2} + \frac{\Delta a_1}{2^2} + R_2. \tag{1.12}$$

Proceeding in the same fashion, after n steps we obtain

$$2^n S = 2^{n-1}a_1 + 2^{n-2}\Delta a_1 + 2^{n-3}\Delta^2 a_1 + \cdots + \Delta^{n-1}a_1 + \sum_{k=1}^{\infty}(-1)^{k-1}\Delta^n a_k,$$

or, upon multiplication by 2^{-n},

$$\begin{aligned} S &= \frac{a_1}{2} + \frac{\Delta a_1}{2^2} + \cdots + \frac{\Delta^{n-1}a_1}{2^n} + \frac{1}{2^n}\sum_{k=1}^{\infty}(-1)^{k-1}\Delta^n a_k \\ &= \sum_{k=1}^{n}\frac{\Delta^{k-1}a_1}{2^k} + \frac{1}{2^n}\sum_{k=1}^{\infty}(-1)^{k-1}\Delta^n a_k = T_n + R_n, \end{aligned} \tag{1.13}$$

where

$$T_n := \sum_{k=1}^{n}\frac{\Delta^{k-1}a_1}{2^k} \quad \text{and} \quad R_n := \frac{1}{2^n}\sum_{k=1}^{\infty}(-1)^{k-1}\Delta^n a_k, \tag{1.14}$$

are the nth partial sum and the remainder of the new series, respectively. In the above process we have used the following properties of infinite series [22, p. 75]:

(i) A convergent series (with sum S) may be multiplied by a constant (c) and the resulting series remains convergent (to sum cS).

(ii) The sum of the resulting series obtained by inserting parentheses in a convergent series is not altered.

(iii) The sum of a convergent series is not altered when one replaces each term by an equivalent sum of a finite number of terms all having the same sign as the replaced one (here $2a_k$ has been replaced by $a_k + a_k$).

It can be shown that R_n approaches zero as n tends to infinity, provided the series (1.10) converges [20, p.245]. Thus we have the following theorem.

Theorem 1..5. *Given an arbitrary convergent series $\sum_{n=1}^{\infty}(-1)^{n-1}a_n$, we apply the Euler transformation to obtain a new series*

$$\sum_{n=1}^{\infty}\frac{\Delta^{n-1}a_1}{2^n} = \frac{a_1}{2} + \frac{\Delta a_1}{2^2} + \frac{\Delta^2 a_1}{2^3} + \cdots, \tag{1.15}$$

which also converges and has the same sum as the given series. That is,

$$\sum_{n=1}^{\infty}(-1)^{n-1}a_n = \sum_{n=1}^{\infty}\frac{\Delta^{n-1}a_1}{2^n}. \tag{1.16}$$

Remark 1.1 We observe that the given series (1.10) need not be alternating for Euler's transformation to be applicable; thus, the numbers a_n in (1.10) need not be positive [20, p.245]. This can be seen in Example 1..2.

From the standpoint of numerical analysis, we are interested in which class of series Euler's transformation can efficiently accelerate. If the series is slowly convergent, alternating, and its associated sequence is completely monotonic, then Euler's transformation works efficiently, as we shall show. We therefore introduce the class of completely monotonic sequences and give a theorem to estimate an upper bound on the remainder of the transformed series.

Definition 1. *[16, p.108] The sequence* $\{a_n\}_{n=1}^{\infty}$ *is* **completely monotonic** *if its elements and successive differences are all non-negative. That is, it satisfies the inequalities*

$$\Delta^k a_n \geq 0, \quad \forall\, k \geq 0,\ n \geq 1. \tag{1.17}$$

For example, the sequences $\{1/n\}_{n=1}^{\infty}$ and $\{e^{-n}\}_{n=1}^{\infty}$ are completely monotonic. We say that a sequence $\{a_n\}_{n=1}^{\infty}$ is *completely monotonic from some stage on*: If there exists a positive integer n_0 so that its subsequence $\{a_n\}_{n=n_0+1}^{\infty}$ is completely monotonic.

A convergent alternating series $\sum_{n=1}^{\infty}(-1)^{n-1}a_n$, whose associated sequence $\{a_n\}_{n=1}^{\infty}$ is completely monotonic, is called **a completely alternating series**. For the class of completely alternating series, we have the following theorem to estimate an upper bound on the remainder when using Euler's transformation.

Theorem 1..6. *[21] If the series* $\sum_{n=1}^{\infty}(-1)^{n-1}a_n$ *is completely alternating, then the remainder of the transformed series, after applying Euler's transformation, is bounded by*

$$0 < R_n \leq \frac{\Delta^n a_1}{2^n}. \tag{1.18}$$

Proof. Since the sequence $\{a_n\}_{n=1}^{\infty}$ is completely monotonic, by the definitions (1.9) and (1.17), we obtain the inequalities

$$\Delta^n a_k \geq \Delta^{n+1} a_k \geq 0, \quad \Delta^n a_k \geq \Delta^n a_{k+1} \geq 0, \quad \forall\ n \geq 0,\ k \geq 1. \tag{1.19}$$

Thus, the remaining series R_n in (1.14) is also completely alternating and convergent. According to the theorem on convergent alternating series, the sum of the remaining series R_n is bounded by the first neglected term in absolute values (see Knopp [20, p.250]). Hence the inequality (1.18) holds. □

Remark 1.2: This theorem gives an estimate for the error when Euler's transformation is applied to a completely alternating series. However, even for alternating series that are not completely alternating, it is still a good estimate for the error since the transformed series typically converges so rapidly that the remainder can be estimated by the first neglected term (see Example 1..3).

The summation of the transformed series by Euler's transformation involves difference calculations, which require more arithmetic operations than summing the original series. This means that when we evaluate the finite series

$$\sum_{k=1}^{n} \frac{\Delta^{k-1} a_1}{2^k} = \frac{a_1}{2} + \frac{\Delta a_1}{2^2} + \frac{\Delta^2 a_1}{2^3} + \cdots + \frac{\Delta^{n-1} a_1}{2^n} \tag{1.20}$$

to approximate the original series (1.10), we actually use the same number of terms, $a_1, a_2, \cdots, a_n$, as summing n terms of the original series

$$\sum_{k=1}^{n} (-1)^{k-1} a_k = a_1 - a_2 + \cdots + (-1)^{n-1} a_n. \tag{1.21}$$

However, for summing (1.20), the computer has to do at least $n(n-1)/2$ subtractions to calculate the $\Delta^k a_1$ for $k = 1, 2, \cdots, n-1$, (by the recursive algorithm) plus n divisions and $n-1$ additions.

In the difference calculations, the subtraction operations may cause cancellation so as to increase the relative errors (this cancellation is also known as *loss of significant digits*). However, for the class of alternating convergent series, our numerical experiments

show that the cancellation in Euler's transformation does not have a severe influence on the accuracy of the approximation of (1.20) (see Example 1..1 and 1..2. The reason is that the most significant cancellation happens in the subtraction of the two closest numbers during the calculation of $\Delta^{n-1}a_1$, but the last term of (1.20) is very small, compared to the first term, since the number $\Delta^{n-1}a_1$ is divided by 2^n. Therefore, the significant digits of the rear terms, $(\Delta^{k-1}a_1)/2^k$ (for $k \geq 2$), are less important than those of the first term, so that the influence of cancellation is negligible.

Next, we investigate the roundoff error introduced by summing a finite number of terms of the transformed series (1.20), compared to the roundoff of summing the same terms of the original series (1.21). There are two reasons that, *for the class of alternating convergent series, summing a finite number of terms of the transformed series does not increase the roundoff errors compared to summing the same terms of the original series.* First, the summation of the series (1.20) involves difference calculus which causes cancellation at each subtraction. However, in floating-point arithmetic when cancellation takes place, the roundoff is zero [23, p. 49] or [24, p. 9]. Second, every term $\Delta^k a_1$ in the series (1.20) is divided by the exact floating-point number 2^k so that the roundoff becomes negligible. That there is no increase in roundoff error, can be seen from the numerical results of the following example.

Example 1..1. *Consider the well-known alternating harmonic series*

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n} \tag{1.22}$$

with exact sum $\log(2)$.

It can easily be shown that the associated sequence of this series, $\{1/n\}_{n=1}^{\infty}$, is completely monotonic. This implies that the Euler transformation is efficient for this series. Now, applying the Euler transformation to the series (1.22), we obtain a new series

$$\sum_{n=1}^{\infty} \frac{1}{n\,2^n}, \tag{1.23}$$

which has the same sum, but it converges more rapidly than the original series (1.22). This can be seen in Figure 1.1. In this figure, we plotted the relative error curves versus the number of terms N for the direct sum of (1.22) and the sum of the transformed series (1.23). As one may note, when applying Euler's transformation to sum the alternating series (1.22), we used only 48 terms and the computed error is about 1.1×10^{-16}, which is the same as the roundoff unit $\mathbf{u}$ (in IEEE standard double precision arithmetic $\mathbf{u} = 2^{-53} \approx 1.1 \times 10^{-16}$). It is evident that no roundoff is accumulated.

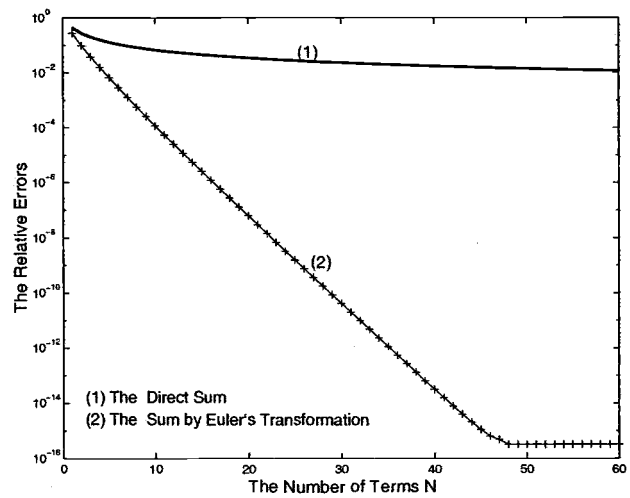

Figure 1.1: Application of Euler's transformation to the series (1.22).

Most authors [21], [25, p.201] suggest that several terms of the series should be summed directly with Euler's transformation applied to the remaining terms. In Chapter 2 we shall follow this strategy to sum the series in the numerical inversion formula of the Laplace transform.

1.3.2. Epsilon Algorithm and Examples

Although Euler's transformation is applicable to this inverse problem, in some cases the ε-algorithm is more efficient (see Example 1..2). We therefore briefly introduce this algorithm without going into detail. The ε-algorithm is a simple algorithm used to compute the Shanks transformation (also called the Schmidt transformation) [26, p.120]. It

may be summarized as follows. Let $n = 2q+1$, $q \in \mathbb{N}$, $s_m = \sum_{k=1}^{m} a_k$, and define

$$\begin{aligned} \varepsilon_{p+1}^{(m)} &= \varepsilon_{p-1}^{(m+1)} + \frac{1}{\varepsilon_p^{(m+1)} - \varepsilon_p^{(m)}}, \quad m, p = 1, 2, \ldots, \\ \varepsilon_0^{(m)} &= 0, \quad \varepsilon_1^{(m)} = s_m. \end{aligned} \tag{1.24}$$

Then the sequence, $\varepsilon_1^{(1)}, \varepsilon_3^{(1)}, \varepsilon_5^{(1)}, \ldots, \varepsilon_{2q+1}^{(1)} = \varepsilon_n^{(1)}$, converges to $s = \sum_{k=1}^{\infty} a_k$, provided the series is convergent. For more details we refer to Wimp's book [26, p.138] and Brezinski's book [17, p.10]. We next give some examples illustrating the efficiency of the two convergence accelerators.

Example 1..2. *Consider the irregular oscillating series*

$$\sum_{n=1}^{\infty} \frac{\sin n}{n} \tag{1.25}$$

with exact sum $s = (\pi - 1)/2$.

We first apply Euler's transformation to this series by rewriting it in alternating form

$$\sum_{n=1}^{\infty} (-1)^{n-1} a_n, \quad \text{with} \quad a_n = \frac{(-1)^{n-1} \sin n}{n}. \tag{1.26}$$

This is not a completely alternating series since a_n is not completely monotonic. Therefore, the transformed series by Euler's transformation does not converge particularly rapidly, but still converges faster than the original series (see Figure 1.2). We also apply the ε-algorithm to this series. One can see that it works more efficiently than the Euler transformation does by comparing the number of evaluation terms N (see Figure 1.2 and Figure 1.3). To achieve an accuracy of about 10^{-15}, the ε-algorithm employed only 64 terms. By contrast, the Euler transformation required at least 198 terms.

Example 1..3. *Consider the convergent alternating series*

$$\sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n^2 + 4}, \tag{1.27}$$

with exact sum $1/8 + (\pi/4)\mathrm{csch}(2\pi)$.

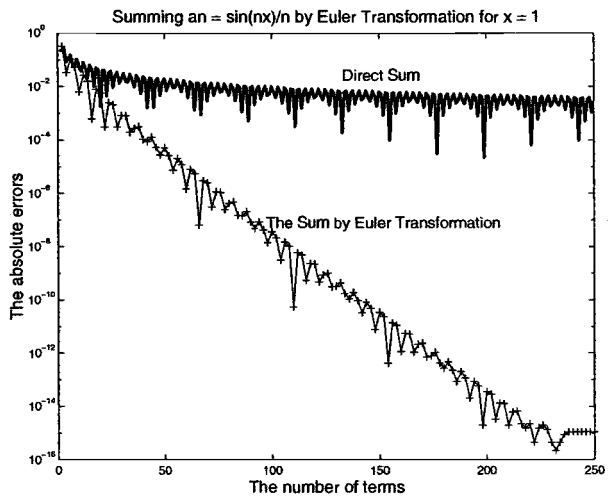

Figure 1.2: Application of the Euler transformation to the series (1..3).

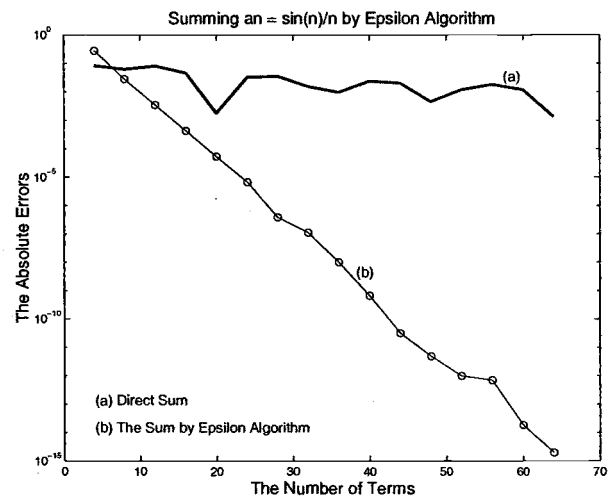

Figure 1.3: Application of the ε-algorithm to the series (1..3).

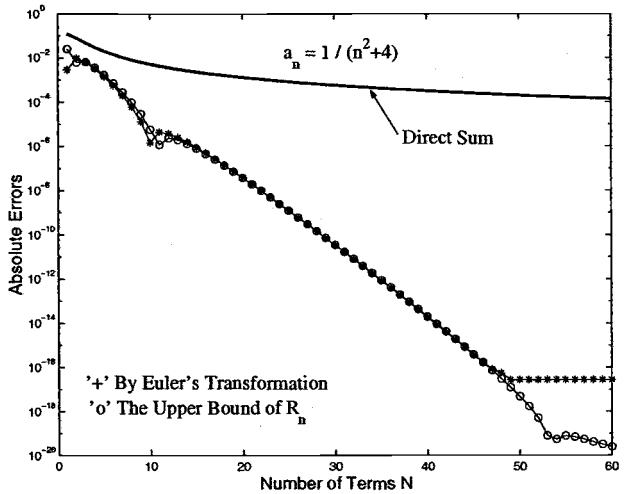

Figure 1.4: Application of Euler's transformation to the series (1.27).

It can be shown that its associated sequence $\{1/(n^2+4)\}_{n=1}^{\infty}$ is not completely monotonic but tends monotonically to zero. In this case, Euler's transformation and its error bound (1.18) are also applicable. In Figure 1.4, we have plotted the error curves of summing the transformed series and the original series versus the number of terms N. For comparison, the error estimate by formula (1.18) is also displayed. One can see that the estimated error is a good approximation to the true error, despite the fact that the error estimate (1.18) is, strictly speaking, not applicable since the series is not completely alternating. In Chapter 2 we shall use (1.18) as a practical error estimate, without establishing that the underlying series are completely monotonic.

1.4. The Trapezoidal-Type Rules

In this dissertation, we shall apply the trapezoidal or midpoint rule throughout Chapter 2 to 4 to approximate the Bromwich integral (1.1). In general, the trapezoidal sum converges slowly unless enhancement techniques are applied. However, even without enhancement, these rules can give good approximations in certain special cases [27]. We shall discuss these cases now.

A general *trapezoidal rule* is defined by [28, p.209]

$$T_h(g) := h\left[\tfrac{1}{2}g(x_0) + \sum_{j=1}^{n-1} g(x_j) + \tfrac{1}{2}g(x_n)\right], \tag{1.28}$$

where $h := (\beta - \alpha)/n$ and $x_j = \alpha + jh \ \ (j = 0, 1, ..., n)$, and approximates the definite integral:

$$\int_\alpha^\beta g(x)\,dx. \tag{1.29}$$

Here we assume that the integrand $g(x)$ is integrable on the interval $[\alpha, \beta]$.

A quadrature rule with the same order of accuracy as the trapezoidal rule is the *midpoint rule*:

$$M_h(g) := h\sum_{j=0}^{n-1} g(x_j), \quad x_j = \alpha + \left(j + \tfrac{1}{2}\right)h, \tag{1.30}$$

which corresponds to a Riemann sum (also called the rectangular rule). Since this rule has the same order of accuracy as the trapezoidal rule, we refer to both of them as trapezoidal-type rules. We present the theory below for the trapezoidal rule, with the understanding that all of them can also be applied to the rectangular rule.

A useful formula for estimating the error for the trapezoidal rule is the Euler-Maclaurin formula [28, p.209].

Theorem 1..7. *Let* $g : [\alpha, \beta] \longrightarrow \mathbb{R}$ *have* mth *order continuous derivatives, for* $m \geq 2$. *Then*

$$\begin{aligned}\int_\alpha^\beta g(x)\,dx \;&=\; T_h(g) - \sum_{k=1}^{\left[\frac{m}{2}\right]} \frac{b_{2k}h^{2k}}{(2k)!}\left[g^{(2k-1)}(\beta) - g^{(2k-1)}(\alpha)\right] \\ &\quad + (-1)^m h^m \int_\alpha^\beta \widetilde{B}_m\left(\frac{x-\alpha}{h}\right) g^{(m)}(x)\,dx,\end{aligned} \tag{1.31}$$

where the b_{2k} *are the Bernoulli numbers,* $\widetilde{B}_m$ *is the periodic extension of the Bernoulli polynomials* B_m, *and* $[\frac{m}{2}]$ *denotes the largest integer smaller than or equal to* $\frac{m}{2}$.

Proof: For a proof, we refer to [28, p. 209].

One can see that the trapezoidal rule has an accuracy of order h^2 if the function g has the property that $g'(\beta) - g'(\alpha) \neq 0$, which is typically the case. However, when $g'(\beta) - g'(\alpha) = 0$ and, in particular, when $g(x)$ is smooth and periodic with period $\beta - \alpha$ (for instance 2π) a higher accuracy can be achieved. This can be seen from the following corollary.

For the integral (1.29) whose integrand $g(x)$ is $(\beta - \alpha)$-periodic and continuous, the trapezoidal rule coincides with the rectangular rule:

$$\int_\alpha^\beta g(x)\,dx \approx h \sum_{j=1}^{n} g(\alpha + jh), \tag{1.32}$$

and its error is defined by

$$|E_n(g)| := \int_\alpha^\beta g(x)\,dx - h \sum_{j=1}^{n} g(\alpha + jh). \tag{1.33}$$

Corollary 1..7.1. *Let $g : \mathbb{R} \longrightarrow \mathbb{R}$ be $(\beta - \alpha)$-periodic and $(2m+1)$-times continuously differentiable for $m \in \mathbb{N}$. Then, for the error produced by the rectangular rule, we have*

$$|E_n(g)| \leq \frac{C}{n^{2m+1}} \int_\alpha^\beta |g^{(2m+1)}(x)|\,dx,$$

where

$$C = 2 \sum_{k=1}^{\infty} \frac{1}{k^{2m+1}}.$$

Proof: For a proof, we refer to [28, p. 210].

This corollary tells us that for a $(\beta - \alpha)$-periodic function with continuous $(2m+1)th$ order derivatives, the trapezoidal rule has accuracy of $O(1/n^{2m+1})$, in contrast to the typical accuracy $O(1/n^2)$ for the non-periodic case.

For a function that is periodic and analytic, a stronger error estimate may be proved by means of contour integration [28, p. 211].

Theorem 1..8. *Let $g : \mathbb{R} \longrightarrow \mathbb{R}$ be a real-analytic and 2π-periodic function. Then there exists a strip $\mathcal{D} = \mathbb{R} \times (-d, d)$ in the complex-plane with $d > 0$ such that g can be extended to a complex-analytic and 2π-periodic bounded function $g : \mathcal{D} \longrightarrow \mathbb{C}$. Furthermore, the*

error for the rectangular rule is represented by the contour integral:

$$E_n = \mathrm{Re} \int_{i\gamma}^{i\gamma+2\pi} \left(1 - i \cot \frac{nz}{2}\right) g(z)\, dz, \tag{1.34}$$

where $0 < \gamma < d$. *Moreover, it is bounded by*

$$|E_n| \leq \frac{4\pi M}{e^{nd} - 1}, \tag{1.35}$$

where M *is a bound of the function* g *on* $\mathcal{D}$.

Proof: For a proof, we refer to [28, p. 211].

This theorem shows that for a periodic and analytic function, the error of the trapezoi-dal-type rules are of exponential order. The corresponding formula for the midpoint rule is as follows.

Corollary 1..8.1. *By the same hypothesis as Theorem 1..8, the error for the midpoint rule may be expressed by*

$$E_n = \mathrm{Re} \int_{i\gamma}^{i\gamma+2\pi} \left(1 + i \tan \frac{nz}{2}\right) g(z)\, dz. \tag{1.36}$$

For further discussion of the convergence rate of the trapezoidal-type rules over periodic functions, we refer to [27]. We now give examples to demonstrate how the trapezoi-dal type rules work in these special cases.

Example 1..4. *Consider the integral*

$$\int_0^{2\pi} \frac{dx}{2 + \cos x}, \tag{1.37}$$

which has the exact value $2\pi/\sqrt{3}$.

Its integrand is a 2π-periodic and analytic function. The extended analytic region is $\mathcal{D} = \mathbb{R} \times (-d, d)$ with $d = \log(2 + \sqrt{3}) - \epsilon$, and the integrand is bounded by

$$M = \max_{z \in \mathcal{D}} \left\{ \left| \frac{1}{2 + \cos z} \right| \right\}.$$

The trapezoidal rule gives exponential convergence as shown in Figure 1.5. In this figure, we computed the absolute error (with base 10 logarithm) of the trapezoidal approximation

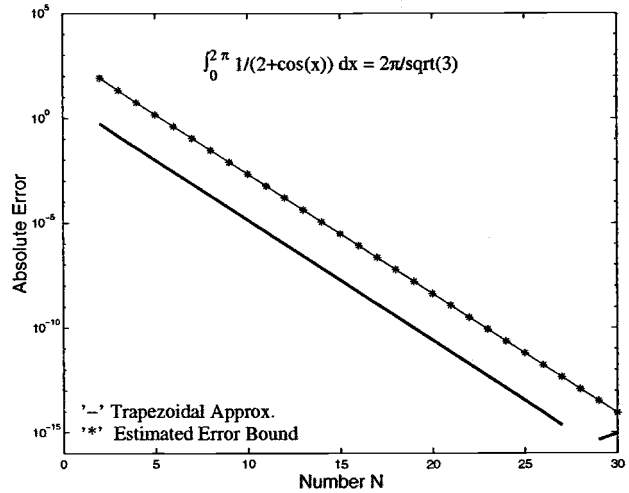

Figure 1.5: The errors in the trapezoidal approximation to the integral (1.37), compared with the estimated error bound (1.35) with $M \approx 83$ and $d \approx 1.31$.

versus the number of evaluation points N and the estimated error by the upper bound (1.35) with $M \approx 83$ and $d \approx 1.31$. The results are reasonable and consistent with the conclusion of Theorem 1..8 (also see [27]).

The trapezoidal-type rules are also suitable for improper integrals over $(-\infty, \infty)$, provided the integrand decays rapidly. In this case, the formulas (1.28) and (1.30) can be generalized to two-sides unbounded interval so that the trapezoidal and midpoint rule become

$$T_h(g) := h \sum_{j=-\infty}^{\infty} g(x_j), \quad x_j = jh, \tag{1.38}$$

$$M_h(g) := h \sum_{j=-\infty}^{\infty} g(x_j), \quad x_j = \left(j + \tfrac{1}{2}\right) h. \tag{1.39}$$

In practice, the infinite sums have to be truncated. The application of these formulas is efficient only for sufficiently fast decaying functions. We now present the following example to demonstrate the efficiency of the trapezoidal-type rules.

Example 1..5. *Consider the improper integral*

$$\int_{-\infty}^{\infty} \mathrm{e}^{-x^2} dx \tag{1.40}$$

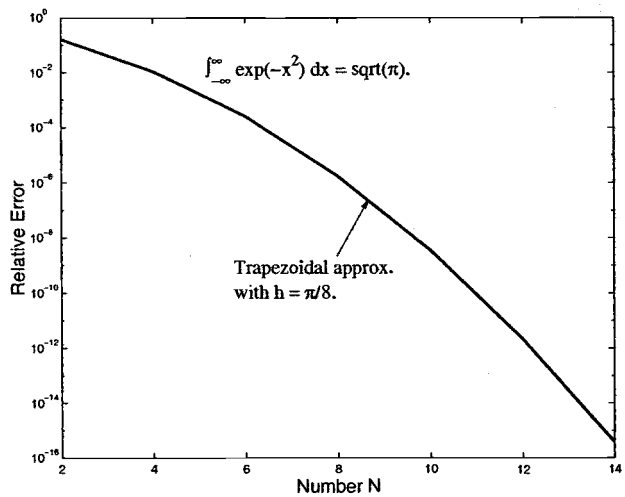

Figure 1.6: The error of the trapezoidal approximation to the integral (1.40) with $h = \pi/8$.

with the value $\sqrt{\pi}$.

Its integrand is a super-exponentially decaying function as $x \to \pm\infty$. Applying the trapezoidal rule to this integral, one finds that it can achieve high accuracy using few terms. In Figure 1.6, we have plotted the relative error computed with step size $h = \pi/8$. We used only 14 terms and obtained an accuracy of nearly machine precision ($\approx 2.2 \times 10^{-16}$). Therefore the trapezoidal-type rules are good quadrature formulas for this kind of problem, as has been investigated by Goodwin [29].

When the integrand decays slowly, the trapezoidal-type rules can be enhanced by some techniques, such as convergence acceleration (see Section 1.3) or the double exponential transformation (which will be introduced in Chapter 4) [13, 14]. The next example demonstrates the trapezoidal rule enhanced by a convergence accelerator, namely the ε-algorithm.

Example 1..6. *Consider the improper integral*

$$\int_0^\infty \frac{\sin x}{x} dx \tag{1.41}$$

with exact value $\pi/2$.

Since the integrand is a slowly decaying oscillatory function, the trapezoidal ap-

proximation converges very slowly. However, if we apply the ε-algorithm to the resulting series, then the method turns out to be efficient. We have plotted the relative errors with respect to the number of terms N in Figure 1.7.

Another technique for dealing with the integral (1.41) is a variable transformation method, which will be discussed in Chapter 4, Example 4..2 .

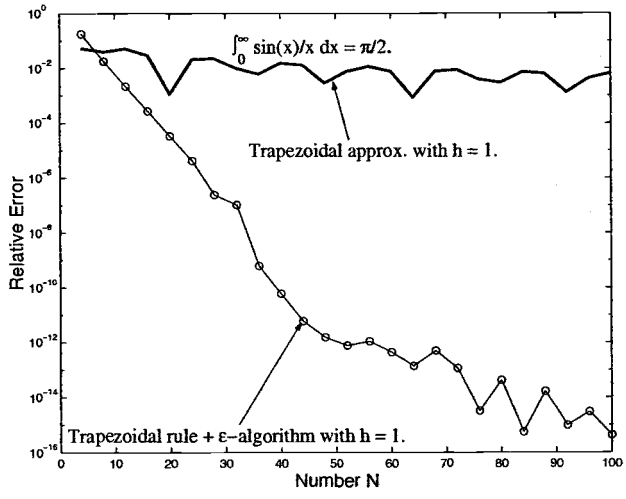

Figure 1.7: The errors in the trapezoidal approximation and in the trapezoidal rule enhanced by the ε-algorithm, when applied to the integral (1.41) with $h = 1$.

2. THE DIRECT INTEGRATION METHOD

In this chapter, we shall focus on the direct integration method. This method is obtained by numerically integrating the Bromwich integral (1.1) using the trapezoidal rule. To accelerate the convergence of the resulting series, sequence acceleration is used. This method contains a free parameter, a, which can be tuned for optimal accuracy. We address the question of how to find the optimal parameter a both from a theoretical as well as a computational standpoint.

The chapter is organized as follows. In the first section we derive the method. In the second section, we analyze three sources of error associated with this method, and present our main theorem for error analysis. Some asymptotic formulas for finding the optimal parameter a will be introduced in the third section. An automatic numerical algorithm for selecting the optimal parameter a, and the number of transform evaluations, will be presented in the fourth section. In the final section, an implementation of the numerical algorithm for both theoretical and practical problems will be demonstrated, and a comparison with the latest work will be discussed.

2.1. Derivation

The direct integration method has been derived from a Fourier series approach by Dubner and Abate [5], and Durbin [6]. However, we derive the same method from the trapezoidal rule, which we believe to be a more natural approach.

We shall assume throughout this chapter that the convergence abscissa, σ_0, of the given transform $F(s)$, is equal to zero. This means that the transform $F(s)$ has its rightmost singularities on the imaginary axis and has no singularities in the right half-plane. We refer to any transforms that have this property as being of class $\mathcal{A}$. In the case, in which some singularities exist to the right of the origin, a suitable translation of the

imaginary axis can be performed by[2] using

$$F(s+d) \quad \bullet\!\!-\!\!\circ \quad \mathrm{e}^{-dt} f(t), \tag{2.1}$$

where $d \neq 0$, so that those singularities are shifted to the left half-plane and the rightmost singularities are on the imaginary axis[3].

We remark that, if $F(s)$ is in class $\mathcal{A}$, then the corresponding original function $f(t)$ satisfies

$$|f(t)| \leq ct^m, \quad t > 0 \tag{2.2}$$

for some constant $c > 0$ and non-negative integer m [5]. It is also assumed that both the transform $F(s)$ and its corresponding inverse function $f(t)$ exist for $\mathrm{Re}(s) > \sigma_0 \geq 0$ and $t > 0$, respectively. That is, for a given transform $F(s)$, which is analytic in the right half-plane of the imaginary axis, the corresponding Bromwich integral (1.1) converges to its inversion $f(t)$ for $t > 0$ in the Cauchy principal value sense, along any vertical line $\mathrm{Re}(s) = a > \sigma_0 \geq 0$.

We start by applying the trapezoidal rule with step size h to the Bromwich integral (1.1). This yields

$$\widetilde{f}(t) = \frac{\mathrm{e}^{at}}{2\pi} \left\{ h \sum_{j=-\infty}^{\infty} e^{ijht} F(a+ijh) \right\}. \tag{2.3}$$

The above series converges for all $t > 0$ by our assumption that the Bromwich integral (1.1) converges in the Cauchy principal value sense. Therefore, if we select the step size $h = \pi/2t$, fixed for each value of t, and separate the real and imaginary parts, we obtain

$$\begin{aligned} \widetilde{f}(t) &= \frac{\mathrm{e}^{at}}{4t} \sum_{j=-\infty}^{\infty} \mathrm{e}^{ij\pi/2} F\left(a + \tfrac{ij\pi}{2t}\right) \\ &= \frac{\mathrm{e}^{at}}{4t} \sum_{j=-\infty}^{\infty} \left[\cos\left(\tfrac{j\pi}{2}\right) + i \sin\left(\tfrac{j\pi}{2}\right)\right] \left[\mathrm{Re}\left\{F\left(a + \tfrac{ij\pi}{2t}\right)\right\} + i\,\mathrm{Im}\left\{F\left(a + \tfrac{ij\pi}{2t}\right)\right\}\right]. \end{aligned}$$

[2]Here we have used Doetsch's **symbol of correspondence** •—o [15, p. 20].

[3]For a transform with no singularities, its inversion involves a distribution. We mention here that throughout this dissertation we do not consider the distribution

By the property that $F(\overline{z}) = \overline{F(z)}$, one obtains

$$\widetilde{f}(t) = \frac{\mathrm{e}^{at}}{2t}\left\{\frac{1}{2}\mathrm{Re}\{F(a)\} + \sum_{n=1}^{\infty}(-1)^n\left[\mathrm{Re}\left\{F\left(a+\frac{n\pi i}{t}\right)\right\} + \mathrm{Im}\left\{F\left(a+\frac{(n-1/2)\pi i}{t}\right)\right\}\right]\right\}, \quad (2.4)$$

which we have already introduced in (1.5). This was first implemented by Duffy [2].

This formula, in spite of the two series that typically converge slowly, has advantages compared to that of Durbin (1.4): The oscillatory factors, sine and cosine, disappear by the choice of $h = \pi/2t$, so that there is no need to evaluate the trigonometric functions. Additionally, the two series typically exhibit an alternating pattern, which allows the application of Euler's transformation for convergence acceleration (see Theorem 1..5 in Chapter 1).

Most convergence accelerators can be applied to this problem. Two of the most efficient convergence accelerators (see [19]), the ε-algorithm and Levin's u-transformation, were employed in Duffy's tests [2]. However, they do not have a convenient error estimate– like most nonlinear convergence accelerators. A well-known exception is the classical Euler transformation of series, which admits an error estimate, since it is linear. This error bound has been presented in Theorem 1..6 of Section 1.3. In addition, in our research, we have found that the Euler transformation also works efficiently for this problem, even though it does not work better than the ε-algorithm and the Levin u-transformation. We shall therefore use Euler's transformation for theoretical analysis and the ε-algorithm in practical algorithms.

After applying Euler's transformation to the two series in (2.4), the inversion formula becomes

$$\widetilde{f}(t) = \frac{e^{at}}{2t}\left\{\frac{\mathrm{Re}\{F(a)\}}{2} - \sum_{k=1}^{\infty}\frac{\Delta^{k-1}}{2^k}\mathrm{Re}\left\{F\left(a+\frac{\pi i}{t}\right)\right\} - \sum_{k=1}^{\infty}\frac{\Delta^{k-1}}{2^k}\mathrm{Im}\left\{F\left(a+\frac{(1/2)\pi i}{t}\right)\right\}\right\}. \quad (2.5)$$

In a practical implementation, these series are, of course, truncated at a certain term, say N.

2.2. Error Analysis

In this section we shall analyze three sources of error in the direct integration method. First, applying numerical quadratures, such as the trapezoidal rule, to an integral generates a *discretization error*. Second, when one truncates an infinite series such as (2.5), a *truncation error* will be produced. Third, in floating-point evaluations of the transform $F(s)$, roundoff errors are introduced and accumulated statistically in the summation of the series. In the formula (2.5), the roundoff errors will be amplified by the exponential factor e^{at} for large values of at; we call this error the *conditioning error*. We therefore model the total error by the formula:

$$\text{Total Error}\,(E) =$$

$$\text{Discretization Error}\,(D) + \text{Truncation Error}\,(T) + \text{Conditioning Error}\,(C).$$

In order to visualize the three kinds of error, we demonstrate the typical error

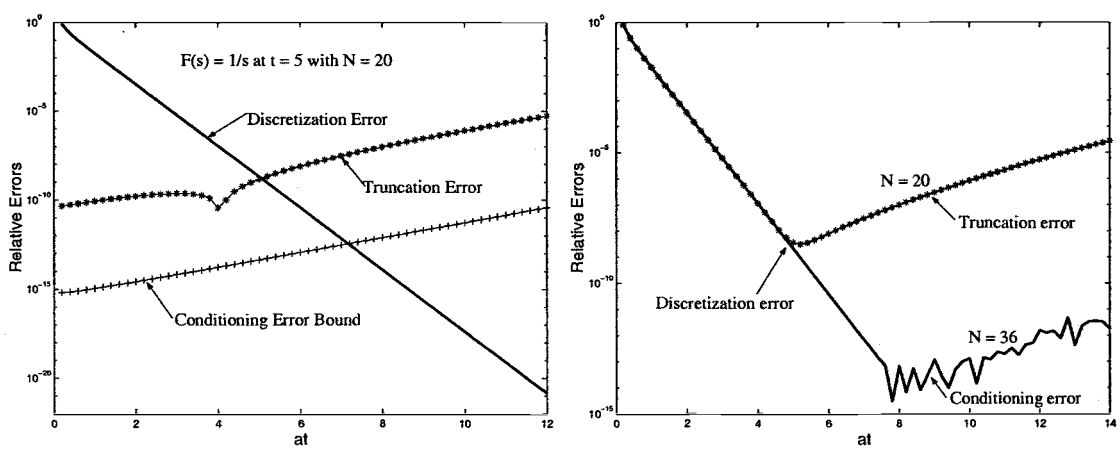

Figure 2.1: Three sources of error in the direct integration method.

models of method (2.5) in Figure 2.1. These pictures are computed by inverting the toy

transform $F(s) = 1/s$ at $t = 5$. This transform is so simple that we have the exact discretization error ($= 1/(e^{4at} - 1)$, see Formula (2.17)), and the actual truncation error by summing the two series exactly in terms of the ψ function. Since it is difficult to get the conditioning error independently, we computed the error bound of conditioning error by using formula (2.28).

In Figure 2.1, we plotted the relative error in base 10 logarithm versus the value at, owing the fact that the optimal value of at tends to a fixed number for a given transform and all values of t ($t > 0$). For this same reason, most of the figures in this chapter will be plotted in similar fashion.

In Figure 2.1 (left), three curves computed separately represent the discretization error, the truncation error, and an upper bound of the conditioning error, respectively. One can see that the discretization error is decreasing exponentially, and the conditioning error and truncation error are increasing as parameter a increases. It is clear that there exists an intersection point of the discretization error and the truncation error (or the conditioning error). At that point, this method produces a minimum error for computing $f(t)$.

In Figure 2.1 (right), the actual computed total errors are displayed for two different values of N. One can see that the total error is dominated either by the discretization error and conditioning error for a sufficiently large N, or by the discretization error and truncation error for a moderate N. In other words, if N is sufficiently large, then the truncation error is negligible. We shall justify this statement later in this section. Inspecting the two error curves further, we notice that the minimum error happens at some value of at, which is the optimal at, denoted as $a_{\text{opt}}t$. It is observed that the discretization error dominates when $at \leq a_{\text{opt}}t$, and either the truncation error or the conditioning error dominates when $at \geq a_{\text{opt}}t$. The optimal value a_{opt} is what we try to find or estimate in this chapter. Next, we analyze the three sources of error, one by one, beginning with the discretization error.

2.2.1. The Discretization Error

In the previous section, we derived the approximate formula (2.4) to Bromwich's integral (1.1) using the trapezoidal rule. The error between the two formulas is the so-called discretization error, $D = \widetilde{f}(t) - f(t)$. Our idea for estimating this error is to express the series (2.4) as a contour integral. This idea comes from the equivalent contour integral representations for the error in the approximations of the trapezoidal or rectangular rule to the integral involving periodic integrands (see Theorem 1..8). This is the subject of the following theorem, which we consider to be the main analytical contribution of this dissertation. A similar theorem was proved by Murli and Patruno [30].

Theorem 2..1. *Suppose that the image function $F(s)$ of Laplace transform is analytic in the right half-plane $Re(s) > 0$ and its corresponding Bromwich integral (1.1) exists and converges to the inverse function $f(t)$ for all $t > 0$. Let the approximation to the Bromwich integral by the trapezoidal rule with step size $h = \pi/2t$ be*

$$\widetilde{f}(t) = \left(\frac{e^{at}}{2t}\right)\left\{\frac{\mathrm{Re}\{F(a)\}}{2} + \sum_{n=1}^{\infty}(-1)^n\left[\mathrm{Re}\left\{F\left(a+\frac{n\pi i}{t}\right)\right\} + \mathrm{Im}\left\{F\left(a+\frac{(n-1/2)\pi i}{t}\right)\right\}\right]\right\}. \quad (2.6)$$

Then the discretization error, $D = \widetilde{f}(t) - f(t)$, of this approximation can be represented by the contour integral

$$D(a,t) = \frac{1}{2\pi i}\int_{b-i\infty}^{b+i\infty}\frac{e^{ts}F(s)}{e^{-4t(s-a)}-1}\,ds, \quad (2.7)$$

along the vertical line $Re(s) = b$ for an arbitrary b with $0 < b < a$.

Proof. Recall from Section 2.1 that the formula (2.6) actually comes from the formula (2.3) with step size $h = \pi/2t$. We therefore rewrite (2.3) as

$$\widetilde{f}(t) = \lim_{K\longrightarrow\infty}\frac{h}{2\pi}\sum_{j=-K}^{K} e^{t(a+ijh)}F(a+ijh), \quad (2.8)$$

where K is a positive integer. We shall show that this formula corresponds to the sum of residues of the function

$$G(s) = \frac{e^{ts}F(s)}{e^{2\pi(s-a)/h}-1}.$$

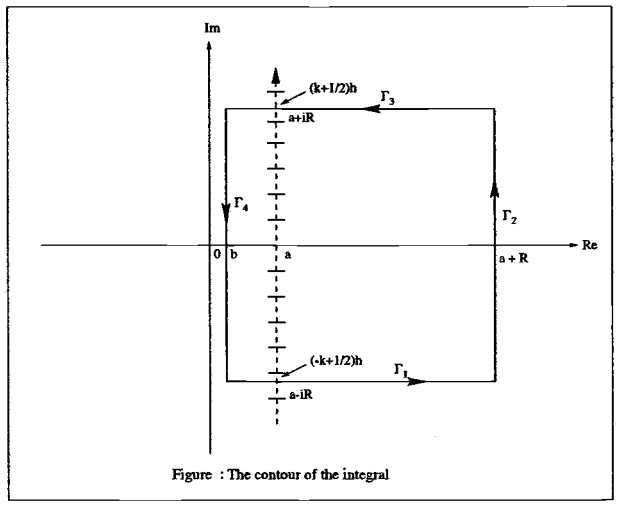

Figure 2.2: The contour used in the proof of the Theorem 2..1.

Suppose that we integrate $G(s)$ along the closed square contour Γ, composed of two horizontal line sections, Γ_1 and Γ_3, below and above the real axis, respectively, and two vertical line sections, Γ_2 and Γ_4, at $\mathrm{Re}(s) = a + R$ and $\mathrm{Re}(s) = b$ respectively (see Figure 2.2). Here $0 < b < a$, and R is chosen as $(K+1/2)h$ for positive integer K such that the square contour Γ encloses $(2K+1)$ singularities of the function $1/(\exp(2\pi(s-a)/h)-1)$ inside the contour. The integration proceeds along the closed contour Γ in the positive sense. Thus, according to the residue theorem, we have

$$\begin{aligned}
\frac{1}{2\pi i}\int_{\Gamma} G(s)\,ds &= \frac{1}{2\pi i}\int_{\Gamma} \frac{\mathrm{e}^{ts}F(s)}{\mathrm{e}^{2\pi(s-a)/h}-1}\,ds \\
&= \sum_{j=-K}^{K} \mathrm{Res}\left(\frac{\mathrm{e}^{ts}F(s)}{\mathrm{e}^{2\pi(s-a)/h}-1},\ s_j = a+ijh\right) \\
&= \frac{h}{2\pi}\sum_{j=-K}^{K} e^{t(a+ijh)}F(a+ijh). \qquad (2.9)
\end{aligned}$$

By comparing (2.9) with (2.8) we obtain

$$\begin{aligned}
\widetilde{f}(t) &= \lim_{R\longrightarrow\infty}\frac{1}{2\pi i}\int_{\Gamma}\frac{\mathrm{e}^{ts}F(s)}{\mathrm{e}^{2\pi(s-a)/h}-1}\,ds \\
&= \lim_{R\longrightarrow\infty}\frac{1}{2\pi i}\left(\int_{\Gamma_1}+\int_{\Gamma_2}+\int_{\Gamma_3}+\int_{\Gamma_4}\right)\frac{\mathrm{e}^{ts}F(s)}{\mathrm{e}^{2\pi(s-a)/h}-1}\,ds. \qquad (2.10)
\end{aligned}$$

Recall that $F(s)$ is analytic in the right half-plane $\mathrm{Re}(s) > 0$. We therefore have, by Cauchy's theorem, the equivalent integrals [15]

$$f(t) = \lim_{R\longrightarrow\infty} \frac{1}{2\pi i}\int_{a-iR}^{a+iR} e^{ts}F(s)\,ds = -\lim_{R\longrightarrow\infty} \frac{1}{2\pi i}\int_{\Gamma_4} \mathrm{e}^{ts}F(s)\,ds. \qquad (2.11)$$

Thus, combining (2.10) and (2.11), the discretization error can be expressed as

$$\begin{aligned} D(a,t) &= \widetilde{f}(t) - f(t) \\ &= \lim_{R\longrightarrow\infty} \frac{1}{2\pi i}\left\{\left(\int_{\Gamma_1}+\int_{\Gamma_2}+\int_{\Gamma_3}\right)\frac{\mathrm{e}^{ts}F(s)}{\mathrm{e}^{2\pi(s-a)/h}-1}\,ds + \int_{\Gamma_4}\frac{\mathrm{e}^{ts}F(s)}{1-\mathrm{e}^{-2\pi(s-a)/h}}\,ds\right\} \\ &= I_1 + I_2 + I_3 + I_4. \end{aligned} \qquad (2.12)$$

We now claim that

$$I_j = \int_{\Gamma_j} G(s)\,ds \longrightarrow 0 \quad \text{as} \quad R\longrightarrow\infty,$$

for each $j = 1, 2,$ and 3.

(1) *The integrals I_1 and I_3 approach zero as $R \longrightarrow \infty$:*

The line segment Γ_1 is parameterized as $s = x - iR$, $b \le x \le a + R$, where $R = (K+1/2)h = \pi(K+1/2)/(2t)$. On this line segment, the integrand may be estimated as

$$\begin{aligned} |G(s)| &= \left|\frac{\mathrm{e}^{t(x-iR)}F(x-iR)}{\mathrm{e}^{4t[(x-iR)-a]}-1}\right| = \frac{|\mathrm{e}^{t(x-iR)}||F(x-iR)|}{|\mathrm{e}^{4t(x-a)-i2\pi(K+1/2)}-1|} \\ &= \frac{\mathrm{e}^{tx}|F(x-iR)|}{|\mathrm{e}^{4t(x-a)}\mathrm{e}^{-i\pi}-1|} \le \frac{\mathrm{e}^{tx}M_R}{\mathrm{e}^{4t(x-a)}+1}, \end{aligned} \qquad (2.13)$$

where M_R is an upper bound on $|F(s)|$ for all s on the line segment Γ_1. Since $F(x-iR)$ satisfies the following property of the Laplace transform

$$|F(s)| = |F(x+iR)| \le M_R \longrightarrow 0 \quad \text{as} \quad R\longrightarrow\infty \qquad (2.14)$$

for every vertical line, $\mathrm{Re}(s) = x > \sigma_0$ (see Theorem 1..3), by (2.13) and (2.14), we have that

$$\begin{aligned} |I_1| &= \frac{1}{2\pi}\left|\int_{\Gamma_1} G(s)\,ds\right| \le \frac{M_R}{2\pi}\int_b^{a+R}\frac{\mathrm{e}^{tx}}{\mathrm{e}^{4t(x-a)}+1}\,dx \le \frac{\mathrm{e}^{4at}M_R}{2\pi}\int_b^{a+R}\mathrm{e}^{-3tx}\,dx \\ &= \frac{\mathrm{e}^{at}M_R}{6\pi t}\left(\mathrm{e}^{3(a-b)t}-\mathrm{e}^{-3tR}\right) \longrightarrow 0, \quad \text{as} \quad R\longrightarrow\infty. \end{aligned}$$

Similarly, I_3 vanishes as $R \longrightarrow \infty$.

(2) *The integral I_2 vanishes as $R \longrightarrow \infty$:*

Along the vertical line segment Γ_2, $s = (a+R) + iy$, $-R \leq y \leq R$, so that

$$I_2 = \frac{1}{2\pi i} \int_{\Gamma_2} \frac{e^{ts} F(s)}{e^{2\pi(s-a)/h} - 1}\, ds = \frac{1}{2\pi} \int_{-R}^{R} \frac{e^{t(a+R+iy)} F(a+R+iy)}{e^{4t(R+iy)} - 1}\, dy.$$

Because $|z-w| \geq ||z| - |w||$ for all complex z and w, it follows that $|e^{4t(R+iy)} - 1| \geq |e^{4tR} - 1|$. We thus obtain the inequality

$$\begin{aligned} |I_2| &\leq \frac{1}{2\pi} \int_{-R}^{R} \left| \frac{e^{t(a+R+iy)} F(a+R+iy)}{e^{4t(R+iy)} - 1} \right| dy \leq \frac{e^{t(a+R)} M_R}{2\pi(e^{4tR} - 1)} \int_{-R}^{R} dy \\ &= \frac{e^{t(a+R)} M_R R}{\pi(e^{4tR} - 1)} \longrightarrow 0, \quad \text{as} \quad R \longrightarrow \infty. \end{aligned}$$

Putting these results back into (2.12), we conclude that $D(a,t)$ approaches I_4 when $R \longrightarrow \infty$. That is,

$$D(a,t) = \lim_{R \longrightarrow \infty} \frac{1}{2\pi i} \int_{\Gamma_4} \frac{e^{ts} F(s)}{1 - e^{-2\pi(s-a)/h}}\, ds = \frac{1}{2\pi i} \int_{b-i\infty}^{b+i\infty} \frac{e^{ts} F(s)}{e^{-4t(s-a)} - 1}\, ds,$$

which completes the proof. □

We may use this contour integral in our numerical algorithm to estimate the discretization error. For practical use, the following equivalent representation of the discretization error (in terms of the original function $f(t)$) is also sometimes useful.

Corollary 2..1.1. *The discretization error in the direct integration method may be equivalently represented as:*

$$D(a,t) = \sum_{n=1}^{\infty} \left(e^{-4at}\right)^n f[(4n+1)t], \tag{2.15}$$

where $f(t)$ is assumed to be bounded by a polynomial function as (2.2).

Proof. We may rewrite (2.7) as

$$D(a,t) = \frac{1}{2\pi i} \int_{b-i\infty}^{b+i\infty} \frac{e^{ts} e^{4t(s-a)} F(s)}{1 - e^{4t(s-a)}}\, ds.$$

The factor $1/(1 - e^{4t(s-a)})$ may be expanded in a geometric series, since

$$|e^{4t(s-a)}| = e^{-4t(a-b)} < 1 \quad \text{for} \quad a > b.$$

We therefore have

$$\begin{aligned} D(a,t) &= \frac{1}{2\pi i}\int_{b-i\infty}^{b+i\infty} \mathrm{e}^{5ts}\,\mathrm{e}^{-4at}\left\{\sum_{n=0}^{\infty}\mathrm{e}^{4t(s-a)n}\right\}F(s)\,ds \\ &= \frac{1}{2\pi i}\int_{b-i\infty}^{b+i\infty}\left\{\sum_{n=0}^{\infty}\mathrm{e}^{4ts(n+5)}\,\mathrm{e}^{-4at(n+1)}\right\}\mathrm{e}^{ts}\,F(s)\,ds \\ &= \sum_{n=1}^{\infty}\left(\mathrm{e}^{-4at}\right)^n\left\{\frac{1}{2\pi i}\int_{b-i\infty}^{b+i\infty}\mathrm{e}^{(4n+1)ts}\,F(s)\,ds\right\}. \end{aligned} \tag{2.16}$$

Since F(s) is analytic in the right half-plane $\mathrm{Re}(s) > 0$ and has the property that $|F(b+iy)|$ approaches zero as $|y|$ tends to infinity, the series inside the integral converges uniformly along the vertical line $\mathrm{Re}(s) = b\ (> 0)$. This implies that term-wise integration in (2.16) is permissible. Observe that the integral inside the braces of (2.16) is, for each n, the Bromwich integral of the inverse transform $f[(4n+1)t]$. Hence

$$D(a,t) = \sum_{n=1}^{\infty}\left(\mathrm{e}^{-4at}\right)^n f[(4n+1)t],$$

which completes the proof. □

This expression for the discretization error is equivalent to the error formula derived from the Fourier series approach used by Dubner and Abate [5] and Durbin [6].

It is clear that the representation (2.15) of the discretization error involves the original function $f(t)$ which we typically do not know. If one wants to estimate the discretization error numerically, one should rely on the contour-integral formula (2.7). However, formula (2.15) is still useful in some cases. For instance, if we assume that the original function $f(t)$ is bounded by a positive constant, c say, for all $t \geq 0$, then the discretization error is bounded by

$$\begin{aligned} D(a,t) &= \left|\sum_{n=1}^{\infty}\left(\mathrm{e}^{-4at}\right)^n f[(4n+1)t]\right| \\ &\leq c\sum_{n=1}^{\infty}\left(\mathrm{e}^{-4at}\right)^n \leq \frac{c}{e^{4at}-1}. \end{aligned} \tag{2.17}$$

This formula is similar to the error bound of integrating a periodic and analytic function by the trapezoidal rule given as (1.35). In considering formulas (2.15) and (2.17), we observe

that $F(s)$ should be in class $\mathcal{A}$ (so that $f(t)$ has the property (2.2)), the discretization error decreases as $O(\mathrm{e}^{-4at})$ when a approaches infinity for fixed t.

2.2.2. The Truncation Error

In the approximate inversion formula (2.4), there are two infinite series that need to be evaluated. In a floating-point implementation, one has to truncate at some term, say N. Therefore, the truncation error is

$$T(a,t) = \frac{\mathrm{e}^{at}}{2t}\sum_{k=N+1}^{\infty}(-1)^k\left[\mathrm{Re}\left\{F(a+\frac{k\pi i}{t})\right\}+\mathrm{Im}\left\{F\left(a+\frac{(k-1/2)\pi i}{t}\right)\right\}\right]. \tag{2.18}$$

This error may be unacceptably large, since both series typically converge slowly. A useful idea is to apply sequence accelerators to improve the convergence. In this section we shall apply the Euler transformation of series, introduced in Subsection 1.3.1, as our main convergence accelerator to investigate the truncation error. We recall from Subsection 1.3.1 the strategy of applying Euler's transformation, namely to sum several terms directly and then to apply Euler's transformation to the remaining terms. We follow this approach in summing the two series in formula (2.4). Denote the two sequences in (2.18) by

$$p_k := \mathrm{Re}\left\{F\left(a+\frac{k\pi i}{t}\right)\right\}, \qquad q_k := \mathrm{Im}\left\{F\left(a+\frac{(k-1/2)\pi i}{t}\right)\right\}, \tag{2.19}$$

and let P and Q be the sums of the two series $\sum_{k=1}^{\infty}(-1)^k p_k$ and $\sum_{k=1}^{\infty}(-1)^k q_k$, respectively. We assume that there exists a positive integer N_1 so that both subsequences $\{p_k\}_{k=N_1+1}^{\infty}$ and $\{q_k\}_{k=N_1+1}^{\infty}$ are completely monotonic and thus the two subseries

$$\sum_{k=N_1+1}^{\infty}(-1)^k p_k, \qquad \sum_{k=N_1+1}^{\infty}(-1)^k q_k$$

are completely alternating. We can therefore sum N_1 terms directly and then apply Euler's transformation to the remainders for N_2 terms; so, we use N $(= N_1 + N_2)$ terms in total.

Therefore, by Formula (1.16), the two series can be expressed as

$$\begin{aligned}\sum_{k=1}^{\infty}(-1)^k p_k &= P_{N_1} + \sum_{k=N_1+1}^{\infty}(-1)^k p_k \\ &= P_{N_1} + (-1)^{N_1+1}\sum_{n=1}^{\infty}(-1)^{n-1} p_{N_1+n} \\ &= P_{N_1} + (-1)^{N_1+1}\sum_{n=1}^{\infty}\frac{\Delta^{n-1}p_{N_1+1}}{2^n} \end{aligned} \tag{2.20}$$

and

$$\sum_{k=1}^{\infty}(-1)^k q_k = Q_{N_1} + (-1)^{N_1+1}\sum_{n=1}^{\infty}\frac{\Delta^{n-1}q_{N_1+1}}{2^n}, \tag{2.21}$$

where P_{N_1}, Q_{N_1} are the N_1-th partial sums. We now truncate the transformed series (2.20) and (2.21) at N_2 and thus obtain the truncation error, as defined by

$$\begin{aligned} T(a,t,N) &= \left(\frac{e^{at}}{2t}\right)\left|\sum_{n=N_2+1}^{\infty}\left(\frac{\Delta^{n-1}p_{N_1+1}}{2^n} + \frac{\Delta^{n-1}q_{N_1+1}}{2^n}\right)\right| \\ &= \left(\frac{e^{at}}{2t}\right)\left|\sum_{n=N_2+1}^{\infty}\frac{\Delta^{n-1}}{2^n}\mathrm{Re}\left\{F\left(a+\tfrac{(N_1+1)\pi i}{t}\right)\right\}\right. \\ &\qquad \left. + \sum_{n=N_2+1}^{\infty}\frac{\Delta^{n-1}}{2^n}\mathrm{Im}\left\{F\left(a+\tfrac{(N_1+1/2)\pi i}{t}\right)\right\}\right|. \end{aligned} \tag{2.22}$$

By our assumption that both subsequences $\{p_k\}_{k=N_1+1}^{\infty}$ and $\{q_k\}_{k=N_1+1}^{\infty}$ are completely monotonic, we can apply Theorem 1..6 to obtain an error bound of the remainders of the transformed series. Consequently, by (1.18) we have

$$\sum_{n=N_2+1}^{\infty}\frac{\Delta^{n-1}p_{N_1+1}}{2^n} \le \frac{\Delta^{N_2}p_{N_1+1}}{2^{N_2}}, \quad \sum_{n=N_2+1}^{\infty}\frac{\Delta^{n-1}q_{N_1+1}}{2^n} \le \frac{\Delta^{N_2}q_{N_1+1}}{2^{N_2}}. \tag{2.23}$$

Therefore, the truncation error of the direct integration method can be estimated as

$$T(a,t,N) \le \left(\frac{e^{at}}{2t}\right)\left(\frac{\Delta^{N_2}p_{N_1+1}}{2^{N_2}} + \frac{\Delta^{N_2}q_{N_1+1}}{2^{N_2}}\right). \tag{2.24}$$

In this case, the differences $\Delta^{N_2}p_{N_1+1}$ and $\Delta^{N_2}q_{N_1+1}$ are bounded and thus the truncation error decreases as $O(2^{-\rho N})$ for fixed t, a, and $\rho := N_2/N$, when N increases (i.e., N_2

increases too). On the other hand, for fixed t and N, the truncation error increases as $O(\mathrm{e}^{at})$ when a increases.

Notice that the assumption of the complete monotonicity from some stage (N_1) on of the two sequences $\{p_k\}_{k=1}^{\infty}$ and $\{q_k\}_{k=1}^{\infty}$ arising from the Laplace transform $F(s)$ is not true (it is typically similar to Example 1..3). However, according to our experiments, for most transforms $F(s)$ in class $\mathcal{A}$, the two sequences tend monotonically to zero (see Example 1..3). Following the discussions of Remark 1.2 and Example 1..3, the error bound of Theorem 1..6 is still applicable to be a practical error estimate.

2.2.3. The Conditioning Error

In exact arithmetic, the direct integration method would be implemented as

$$\widehat{f}(t) = \left(\frac{\mathrm{e}^{at}}{2t}\right)\left\{\frac{F(a)}{2} + P_{N_1} + Q_{N_1} + (-1)^{N_1+1}\sum_{k=1}^{N_2}\left[\frac{\Delta^{k-1}p_{N_1+1}}{2^k} + \frac{\Delta^{k-1}q_{N_1+1}}{2^k}\right]\right\}. \tag{2.25}$$

In a floating-point implementation, however, roundoff errors are introduced. The accumulation of these roundoff errors, amplified by the factor e^{at}, is what we refer to as the conditioning error. This is what we shall estimate in this subsection.

We first observe that the main roundoff errors come from summation (including addition and subtraction operations) of the four finite series in formula (2.25). Assuming that, in floating-point arithmetic, the roundoff of each floating-point number is less than or equal to the unit roundoff $\mathbf{u}$, the roundoff errors of summing a finite sequence $\{a_k\}_{k=1}^{N}$ can be estimated.

According to Wilkinson [23, p.17] or Higham [24, p.90], when summing an arbitrary finite sequence $\{a_k\}_{k=1}^{N}$ by a general algorithm, the upper bound of the roundoff error is given by

$$|E_N| \leq \mathbf{u}\cdot(N-1)\sum_{k=1}^{N}|a_k| + O(\mathbf{u}^2), \tag{2.26}$$

where E_N denotes the exact roundoff error of summing a finite sequence $\{a_k\}_{k=1}^{N}$.

This upper bound is a worst-case estimate and, in practice, cancellation of roundoff errors during summation usually happens so that the total roundoff error does not accumulate to be as large as in (2.26). A practical estimate, which takes this cancellation effect into account, was suggested by Wikinson [23, p. 26], and is given by

$$|E_N| \leq \mathbf{u} \cdot \sqrt{N} \sum_{k=1}^{N} |a_k| + O(\mathbf{u}^2). \tag{2.27}$$

The direct integration method (2.25) involves the summation of two transformed series. Recall from Section 1.3 that summing the transformed series by Euler's transformation does not increase the roundoff errors compared with summing the same terms of the original sequence. We can conclude that the roundoff error of summing the four series in formula (2.25) is roughly equivalent to that of summing the two finite series $\sum_{n=1}^{N}(-1)^n p_n$ and $\sum_{n=1}^{N}(-1)^n q_n$. Therefore, by formula (2.27) with the $O(\mathbf{u}^2)$ term neglected, the conditioning error of this method can be estimated by

$$C(a,t,N) \;\leq\; \mathbf{u}\left(\frac{e^{at}}{2t}\right)\left\{\frac{|\mathrm{Re}\{F(a)\}|}{2} + \sqrt{N}\sum_{n=1}^{N}(|p_n| + |q_n|)\right\}. \tag{2.28}$$

Note that in (2.28), if both $\sum_{n=1}^{N}|p_n|$ and $\sum_{n=1}^{N}|q_n|$ are bounded by $O(N)$, then the conditioning error roughly increases as $O(N^{3/2}\cdot\mathbf{u})$ for fixed t and a when N increases. On the other hand, in the preceding subsection we have shown that for fixed t and a the truncation error given by (2.24) decreases as $O(2^{-\rho N})$ for some fixed ρ, $0 < \rho < 1$, when N increases.

Therefore, we conclude that, when N is sufficiently large, the truncation error must be less than the conditioning error. Moreover, the truncation error is also less than the discretization error for fixed t and a when N is sufficiently large, since the discretization error is $O(e^{-4at})$, independent of N. In this situation the total error is dominated by the discretization error and the conditioning error (see Figure 2.1-left). We have accordingly presented what we believe to be convincing arguments that, provided N is sufficiently large, the truncation error after convergence acceleration is negligible, compared to the discretization error and conditioning error.

2.3. Formulas for the Optimal Parameter a

Following the conclusion of the preceding section, we shall estimate further the bounds of the discretization error and the conditioning error and present theoretical formulas for the optimal parameter a in this section. Throughout this section, we assume that the number of terms, N, is sufficiently large so that the truncation error may be neglected, as compared to either the discretization error or the conditioning error. We first estimate the discretization error and the conditioning error by an asymptotic approach.

2.3.1. Estimating the Errors

We have assumed throughout this chapter that the given transform $F(s)$ is in class $\mathcal{A}$, as defined in Section 2.1, i.e., there exist some constant $c > 0$ and a non-negative integer m such that

$$|f(t)| \leq c\,t^m, \quad t > 0. \tag{2.29}$$

For the case $m = 0$, we have seen that $D(a,t) \leq c/(\mathrm{e}^{4at} - 1)$ in (2.17). For $m > 0$, by (2.15) in Section 2.2, for any fixed t, the discretization error is bounded by

$$\begin{aligned} D(a,\,t) &\leq c \sum_{n=1}^{\infty} \left(\mathrm{e}^{-4at}\right)^n [(4n+1)t]^m \\ &\leq c(5t)^m \sum_{n=1}^{\infty} \mathrm{e}^{-4atn} n^m, \quad m = 1, 2, \cdots . \end{aligned} \tag{2.30}$$

The last series of the above inequality is asymptotically approximated by

$$\sum_{n=1}^{\infty} \mathrm{e}^{-4atn} n^m \sim \begin{cases} \frac{m!}{(4at)^{m+1}}, & \text{for a fixed } a \text{ and as } m \longrightarrow \infty, \\ \mathrm{e}^{-4at}, & \text{for a fixed } m \text{ and as } a \longrightarrow \infty. \end{cases} \tag{2.31}$$

A justification of this estimate is provided in Appendix A. In practice, for a given transform $F(s)$, m is fixed and a varies in $(0, \infty)$. Our aim is to determine the optimal value of a. Therefore, from the standpoint of numerical analysis, a reasonable estimate for the series

in (2.30) is

$$\sum_{n=1}^{\infty} \mathrm{e}^{-4atn} n^m \approx \begin{cases} \frac{m!}{(4at)^{m+1}}, & 0 < 4at < m, \\ \mathrm{e}^{-4at}, & m \le 4at. \end{cases} \tag{2.32}$$

Putting (2.32) back into (2.30), we obtain

$$D(a,t) \le \begin{cases} c(5t)^m \frac{m!}{(4at)^{m+1}}, & 0 < 4at < m, \\ c(5t)^m \mathrm{e}^{-4at}, & m \le 4at, \end{cases} \tag{2.33}$$

which is an estimate for the discretization error.

Next, we want to show that the conditioning error, as a function of a, increases as $O(\mathbf{u} \cdot \mathrm{e}^{at}/a^{m+1}t)$ for fixed t and N; on the other hand, as a function of N, it increases at most by $O(N^{3/2} \cdot \mathbf{u})$ for fixed t and a.

Lemma 2..1.1. *Let a given transform $F(s)$ be in class $\mathcal{A}$, then both sequences in formula (2.4) are bounded by $cm!/a^{m+1}$ for some positive constant c and for all positive real number a.*

Proof: It follows from the definition of Laplace transform that for each positive integer n,

$$\begin{aligned} \left| F\left(a + \frac{n\pi i}{t}\right) \right| &= \left| \int_0^\infty \mathrm{e}^{-(a+n\pi i/t)\tau} f(\tau)\, d\tau \right| \le \int_0^\infty \mathrm{e}^{-a\tau} |f(\tau)|\, d\tau \\ &\le c \int_0^\infty \mathrm{e}^{-a\tau} \tau^m\, d\tau = \frac{c\, m!}{a^{m+1}}, \end{aligned}$$

where we have used (2.29). Since $|\mathrm{Re}(w)| \le |w|$ and $|\mathrm{Im}(w)| \le |w|$, we obtain

$$\left| \mathrm{Re}\left\{ F\left(a + \frac{n\pi i}{t}\right) \right\} \right| \le \left| F\left(a + \frac{n\pi i}{t}\right) \right| \le \frac{c\, m!}{a^{m+1}}.$$

Similarly,

$$\left| \mathrm{Im}\left\{ F\left(a + \frac{(n-1/2)\pi i}{t}\right) \right\} \right| \le \left| F\left(a + \frac{(n-1/2)\pi i}{t}\right) \right| \le \frac{c\, m!}{a^{m+1}}. \quad \square$$

Hence, the two sums, $\sum_{n=1}^{N} |p_n|$ and $\sum_{n=1}^{N} |q_n|$ in formula (2.28), are both bounded by $c\, m!\, N/a^{m+1}$, and the conditioning error is estimated by

$$\begin{aligned} C(a,t,N) &\le \mathbf{u} \cdot \left(\frac{\mathrm{e}^{at}}{2t}\right) \left\{ \frac{|F(a)|}{2} + 2\sqrt{N}\, \frac{c\, m!\, N}{a^{m+1}} \right\} \\ &\sim \mathbf{u} \cdot c \left(\frac{m!\, \mathrm{e}^{at}}{a^{m+1}\, t}\right) N^{3/2}, \quad N \longrightarrow \infty. \end{aligned} \tag{2.34}$$

2.3.2. Finding the Optimal Parameter a

We now present formulas for selecting the optimal parameter a in the direct integration method. The optimal parameter a is defined as the value that minimizes the total error as given in the following definition.

Definition 2. *For given t and N, the parameter a of the direct integration method is* optimal *if the magnitude of the total error,*

$$E(a,t,N) = D(a,t) + T(a,t,N) + C(a,t,N),$$

is minimal.

Here we have used $D(a,t)$, $T(a,t,N)$, and $C(a,t,N)$ to represent the three sources of error, namely, the discretization error, truncation error, and conditioning error, respectively. These errors have been represented or estimated by (2.7), (2.24), and (2.28) respectively. To find the optimal parameter a is to find the value that minimizes the function $E(a,t,N)$. Of these three errors, the truncation error is the most difficult to estimate theoretically. However, as we observed in Section 2.2, the truncation error becomes negligible for a sufficiently large N, compared to the other two errors. Under this condition, the total error becomes, approximately,

$$E(a,t,N) \approx D(a,t) + C(a,t,N),$$

or for fixed t and N,

$$E(\alpha) \approx D(\alpha) + C(\alpha). \tag{2.35}$$

Here we have omitted N and t and used α, where $\alpha := at$.

Now, using the estimated error of the discretization error (2.33) and conditioning error (2.34), we can estimate the total error for small α, by

$$E(\alpha) = c(5t)^m m! \left[\frac{1}{(4\alpha)^{m+1}} + \frac{\beta}{m!} \frac{\mathrm{e}^{\alpha}}{\alpha^{m+1}} \right], \qquad 0 < 4\alpha < m, \tag{2.36}$$

and for large α by

$$E(\alpha) = c(5t)^m \left[\mathrm{e}^{-4\alpha} + \beta \frac{\mathrm{e}^{\alpha}}{\alpha^{m+1}} \right], \qquad m \leq 4\alpha. \tag{2.37}$$

Here, the constant β is given by

$$\beta = \mathbf{u}\, m!\, N^{3/2}/5^m. \tag{2.38}$$

The optimal value a (or α) is now obtained by taking the derivative of the function inside the brackets with respect to α, and setting it equal to zero (note that for fixed t, the value t is assumed to be constant). This leads, in the case of small α (i.e., (2.36)) to

$$-\frac{(m+1)}{4^{m+1}} + \frac{\beta}{m!}\left[\alpha - (m+1)\right] \mathrm{e}^{\alpha} = 0.$$

Since it was assumed that $0 < 4\alpha < m$, both terms on the left are negative, as a result, there is no real solution α. It therefore follows that the optimal point must satisfy $\alpha \geq m/4$, and differentiation of (2.37) with respect to α gives

$$\beta\left[\alpha - (m+1)\right] \mathrm{e}^{5\alpha} = 4\alpha^{m+2}. \tag{2.39}$$

This equation has a single real root $\alpha = \alpha_{\mathrm{opt}}$ in the interval $(m+1, \infty)$, which defines the optimal value of a through $a_{\mathrm{opt}} = \alpha_{\mathrm{opt}}/t$. Moreover, α_{opt} increases as m increases.

This may be verified as follows: Rewrite (2.39) as

$$\mathrm{e}^{5\alpha} = \frac{4\alpha^{m+2}}{\beta\left[\alpha - (m+1)\right]} \tag{2.40}$$

and notice that for $\alpha < m+1$ the right-hand side is negative so that there is no real solution in this domain. Thus

$$\alpha_{\mathrm{opt}} > m+1. \tag{2.41}$$

Next, we consider the real solution $\alpha = \alpha_{\mathrm{opt}}$ as the intersection of the graphs of two functions on the left and right-hand sides of (2.40). Define the right-hand side function as

$$r(\alpha) = \frac{4\alpha^{m+2}}{\beta\left[\alpha - (m+1)\right]}. \tag{2.42}$$

Its derivative is then

$$r'(\alpha) = \frac{4\alpha^{m+1}(m+1)[\alpha - (m+2)]}{\beta\left[\alpha - (m+1)\right]^2}, \tag{2.43}$$

which implies that $r(\alpha)$ has a minimum at $\alpha = m + 2$.

We now claim that α_{opt} must satisfy, for some m_*, the inequalities

$$\begin{cases} \alpha_{\text{opt}} > m + 2, & m < m_*, \\ m + 1 < \alpha_{\text{opt}} < m + 2, & m \geq m_*. \end{cases}$$

According to our numerical experiments, $m_* = 12$ when $\mathbf{u} = 2^{-53}$ and $0 < N \leq 85$.

For the case of small m, by recalling the definition of β given in (2.38) and comparing the minimum value of $r(\alpha)$ with $\mathrm{e}^{5\alpha}$, we have

$$r(m+2) = \frac{4(m+2)^{m+2}}{\beta\left((m+2)-(m+1)\right)} = \frac{4(m+2)^{m+2}5^m}{N^{3/2}m!\,\mathbf{u}} > \mathrm{e}^{5(m+2)},$$

for sufficiently small $\mathbf{u}$ (in IEEE standard double precision arithmetic $\mathbf{u} = 2^{-53}$, while in a variable precision arithmetic $\mathbf{u}$ can be arbitrarily small). Also, when $\alpha > m + 2$ and α keeps increasing, we have $r(\alpha) < \mathrm{e}^{5\alpha}$ for sufficiently large α. Hence, in this case there exists an intersection point:

$$\alpha_{\text{opt}} > m + 2. \tag{2.44}$$

As for large m, the function $\mathrm{e}^{5(m+2)}$ becomes so large that the minimum value of $r(\alpha)$ is less than $\mathrm{e}^{5(m+2)}$, and $r(\alpha) \longrightarrow \infty$ as $\alpha \longrightarrow m + 1$. Consequently, there is an intersection point, which satisfies

$$m + 1 < \alpha_{\text{opt}} < m + 2. \tag{2.45}$$

Next, we need to show that the solution α_{opt} is unique. This can be done by comparing the slopes of $r(\alpha)$ and $e^{5\alpha}$ as follows. Using Eq. (2.43), we have

$$r'(\alpha) < \frac{4\alpha^{m+1}(m+1)}{\beta\left[\alpha-(m+1)\right]} < \frac{(m+1)}{\alpha}r(\alpha) = \frac{(m+1)}{\alpha}\mathrm{e}^{5\alpha} < 5\mathrm{e}^{5\alpha},$$

for $\alpha \geq \alpha_{\text{opt}} > m + 1$. Consequently, there can be no other intersection of the graphs of $r(\alpha)$ and $\mathrm{e}^{5\alpha}$.

In general, Eq. (2.39) admits no analytical solution, and we have used Newton's method to solve it. In Table 2.1, we have given the values of α_{opt} as solved from Eq. (2.39),

Table 2.1: Optimal values of α computed by (2.39) with $N = 36$ and $\mathbf{u} = 2^{-53}$.

m	0	1	2	3	4	5	6	7	8	9	10	11	12
α_{opt}	6.97	7.75	8.42	9.03	9.61	10.2	10.7	11.2	11.7	12.2	12.7	13.3	13.8

for various values of $m = 0, 1, \ldots, 12$. For $\mathbf{u}$, we have used the roundoff unit in IEEE standard double precision arithmetic, which is $\mathbf{u} = 2^{-53}$, and we have chosen $N = 36$. The results are fairly insensitive to N in the range [16, 64], which is practical for most problems.

In Figure 2.3, we have computed the transforms of the following model problem

$$F(s) = \frac{1}{s^{m+1}} \quad \bullet\!\!-\!\!\circ \quad f(t) = \frac{t^m}{\Gamma(m+1)}$$

for values $m = 0, 4, 8$, and 12, by the direct integration method, using $t = 1$ and $N = 36$. For each value of m, we have plotted the absolute error as a function of $\alpha = at$. Also shown in the figures are small circles which represent the values of α_{opt}, as given in Table 2.1. It is clear that these estimated values are good approximations to the true optimal values. Therefore, these values of α_{opt} in Table 2.1 can be used to compute accurate approximations of $f(t)$, provided that a sufficiently large N is given.

Observing Eq. (2.40) and (2.45) further, and increasing m as $\mathbf{u}$ and N are kept fixed, we obtain, for large m, the asymptotic estimate:

$$\alpha_{\text{opt}} \sim m + 1, \quad \text{as } m \longrightarrow \infty. \tag{2.46}$$

Therefore, for large m, a practical estimate of the optimal a is given by

$$a_{\text{opt}} \approx (m+1)/t, \tag{2.47}$$

which has been numerically verified in Figure 2.3 (bottom-right).

On the other hand, suppose that m and N are fixed and let $\mathbf{u} \longrightarrow 0$, then $\alpha_{\text{opt}} \longrightarrow$

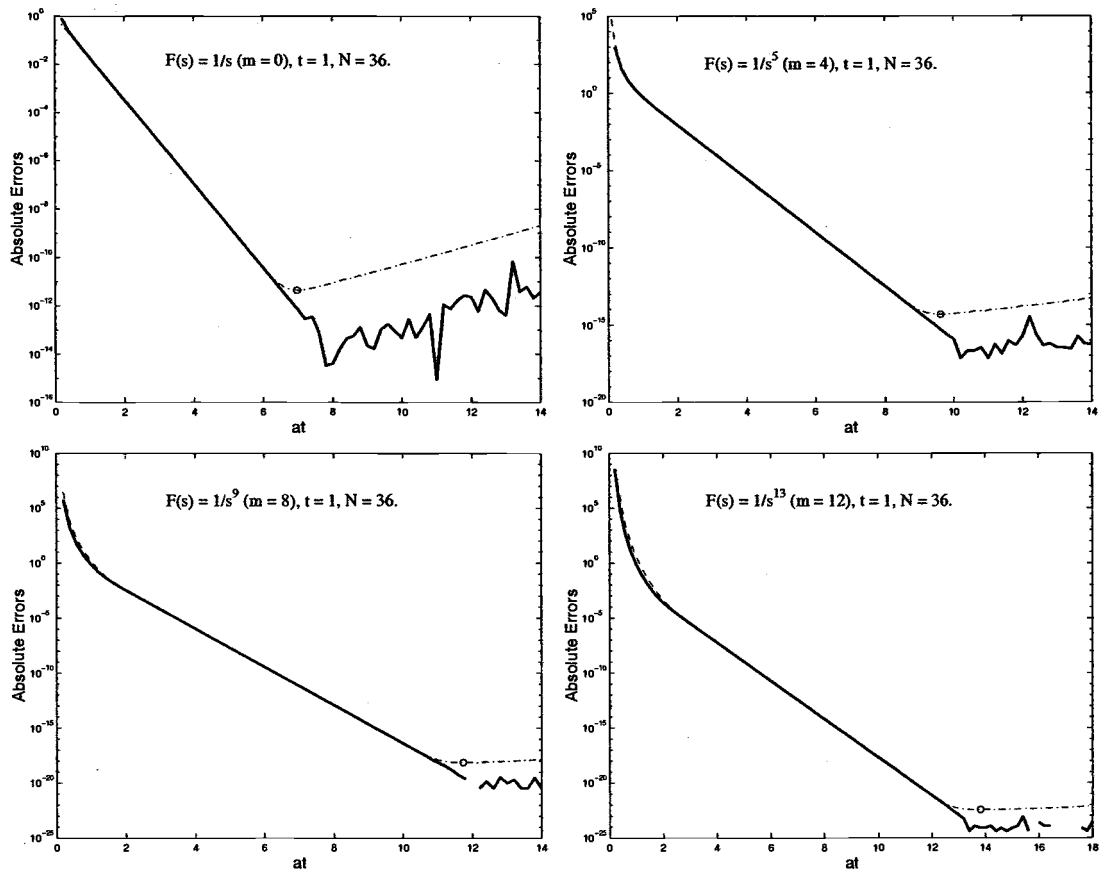

Figure 2.3: Absolute error of the direct integration method (solid line), compared with the theoretical error estimate (2.36) (dash curve) and (2.37) (dash-dot curve). The optimal value of $\alpha_{\text{opt}} = a_{\text{opt}}t$ as determined by Eq. (2.39) is shown as the circle.

∞, since, by Eq. (2.39),

$$\frac{\mathbf{u}m!N^{3/2}}{5^m} = \frac{4\alpha^{m+2}}{[\alpha-(m+1)]\mathrm{e}^{5\alpha}} > \frac{4\alpha^{m+1}}{\mathrm{e}^{5\alpha}} > 0$$

holds. This implies that

$$\alpha_{\text{opt}}^{m+1}\mathrm{e}^{-5\alpha_{\text{opt}}} \longrightarrow 0 \quad \text{as} \quad \mathbf{u} \longrightarrow 0,$$

which leads to $\alpha_{\text{opt}} \longrightarrow \infty$.

By taking logarithms in (2.39), it follows that

$$\log\beta + \log[\alpha-(m+1)] + 5\alpha = \log 4 + (m+1)\log\alpha. \tag{2.48}$$

Since $\alpha_{\text{opt}} \longrightarrow \infty$ as $\mathbf{u} \longrightarrow 0$, with m and N fixed, one observes that in this equation the first and third terms dominate. Also, since

$$\log \beta \sim \log \mathbf{u} \quad \text{as} \quad \mathbf{u} \longrightarrow 0,$$

we obtain

$$\alpha_{\text{opt}} \sim -\frac{1}{5} \log \mathbf{u} \quad \text{as} \quad \mathbf{u} \longrightarrow 0.$$

A practical estimate of a_{opt} for small $\mathbf{u}$ is therefore

$$a_{\text{opt}} t \approx -\frac{1}{5} \log \mathbf{u}. \tag{2.49}$$

With $\mathbf{u} = 2^{-53}$, this becomes

$$a_{\text{opt}} t \approx 7.35.$$

It may be observed both in Table 2.1 and in Figure 2.3 (top-left) that this is a reasonable estimate for small values of m.

In order to check the validity of formula (2.49), we have implemented this method in the Maple software system to invert the toy example $F(s) = 1/s$. Since this transform is simple, we can sum the series exactly and thus no truncation error arises. Therefore the total error is solely governed by the discretization error and the conditioning error.

Table 2.2: Comparison of the optimal values of α computed by (2.49) with different $\mathbf{u}$ and the actual optimal values of α at where the error curves in Figure 2.4 achieve the minimum.

$\mathbf{u}$	$\frac{1}{2} \times 10^{-16}$	$\frac{1}{2} \times 10^{-32}$	$\frac{1}{2} \times 10^{-48}$	$\frac{1}{2} \times 10^{-64}$
$\alpha_{\text{opt}}^{(1)}$	7.35	14.9	22.2	29.6
$\alpha_{\text{opt}}^{(2)}$	7.51	15.0	22.5	30.6

The Maple software system allows us to reduce the value of $\mathbf{u}$ by increasing the number of significant digits used in Maple's variable precision floating-point arithmetic. We have used **Digits** = 16, 32, 48, and 64, which correspond to $\mathbf{u} = \frac{1}{2} \times 10^{-16}$, $\frac{1}{2} \times 10^{-32}$,

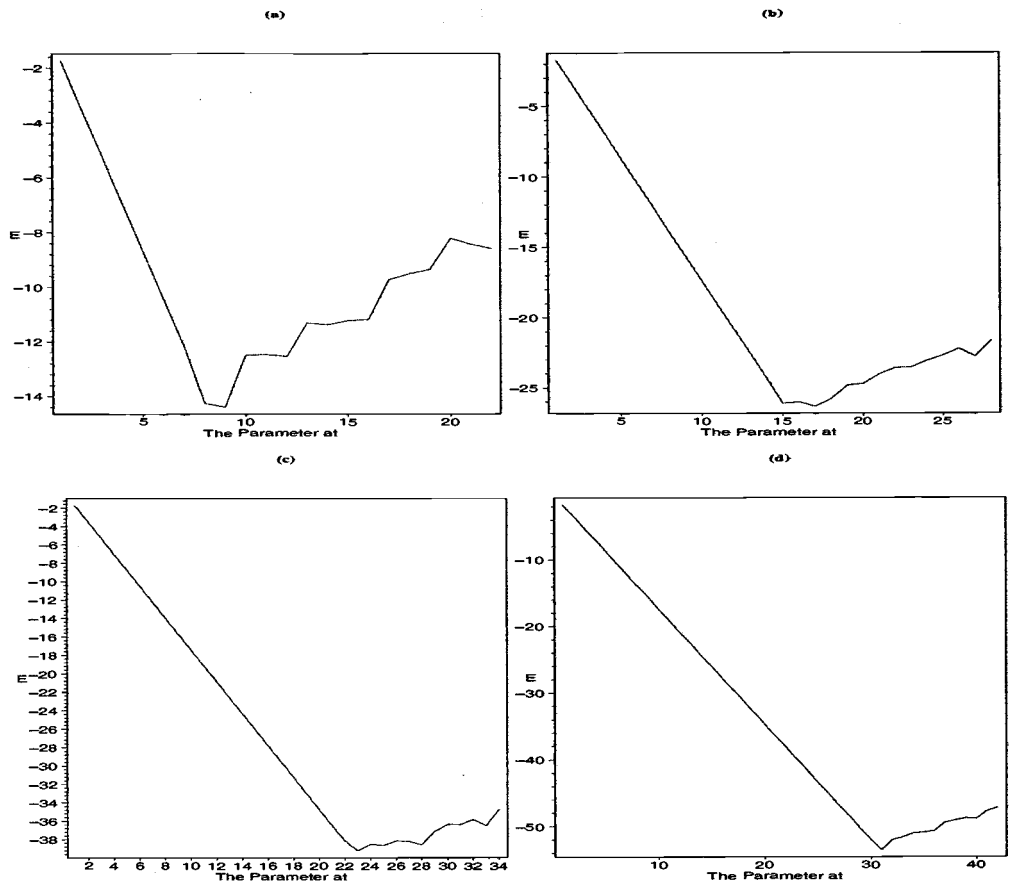

Figure 2.4: Error curves computed in variable precision system (Maple) for inverting the transform $F(s) = 1/s$ at $t = 5$ with $N = 36$. The four figures correspond to $\mathbf{u}$ = (a) $\frac{1}{2} \times 10^{-16}$, (b) $\frac{1}{2} \times 10^{-32}$, (c) $\frac{1}{2} \times 10^{-48}$, (d) $\frac{1}{2} \times 10^{-64}$.

$\frac{1}{2} \times 10^{-48}$, and $\frac{1}{2} \times 10^{-64}$, respectively. By (2.49), the corresponding optimal values are $\alpha_{\text{opt}} = a_{\text{opt}} t \approx 7.35$, 14.9, 22.2, and 29.6, respectively. These values are good estimates for the actual optimal parameter α, as may be seen in Figure 2.4 and Table 2.2. In Table 2.2, $\alpha_{\text{opt}}^{(1)}$ denotes the values computed by (2.49) and $\alpha_{\text{opt}}^{(2)}$ represents the actual values of α at which the error curves in Figure 2.4 achieve a minimum. One can see that they are in reasonable agreement.

For instance, according to Figure 2.4(d), suppose that one has a machine with precision 10^{-64} and wants to invert a transform with the property that the original function $f(t)$ is bounded (i.e., $m = 0$); also, one supplies a value of $t(> 0)$ and applies formula

(2.49) to obtain the optimal parameter $a = 29.6/t$. Thus, one could achieve an accuracy in the approximation of $f(t)$ up to about a factor of 10^{-52} by implementing the direct integration method, provided that N is sufficiently large.

Formulas (2.47) and (2.49) are reasonable estimates for large and small values of m, respectively. A formula applicable to intermediate values of m can be obtained by neglecting only the second term on the left of equation (2.48), that is,

$$\log \beta + 5\alpha_{\text{opt}} \approx \log 4 + (m+1) \log \alpha_{\text{opt}},$$

or

$$(5\alpha_{\text{opt}})^{-(m+1)} \mathrm{e}^{5\alpha_{\text{opt}}} \approx \frac{4}{5^{m+1}\beta}, \tag{2.50}$$

which is in the form of the Lambert W equation. It is possible to express the optimal parameter in terms of the Lambert W function[4]

$$\alpha_{\text{opt}} \approx -\left(\frac{m+1}{5}\right) W_{-1}\left(-\frac{v^{-1/(m+1)}}{m+1}\right), \tag{2.51}$$

where v is defined as

$$v = \frac{4}{5^{m+1}\beta}$$

and W_{-1} denotes one of the real branches of the Lambert W function.

Comparing the optimal values α_{opt} in Table 2.3 obtained by Eq. (2.51) and the values of α_{opt} in Table 2.1 computed by Eq. (2.39), one finds that there is fair agreement for all values of m in the range $[0, 12]$.

The theoretical parameter estimates presented in this section have two limitations. First, it has to be established that the value of N is sufficiently large so that the truncation

[4]The Lambert W function, denoted as $W(x)$, is defined as the root of $W(x)\,\mathrm{e}^{W(x)} = x$. It has infinitely many solutions for each nonzero value of x. The Lambert W function therefore has infinitely many branches, denoted by $W_k(x)$ (k integer). Here we seek only real solutions of $W(x)$ for the real value v, that lies within the branch $W_{-1}(x)$. In Maple, this solution can be obtained by

$$at = -(m+1)/5 * \text{LambertW}\left(-1,\ -v^{-1/(m+1)}/(m+1)\right);.$$

For further details of the Lambert W function, we refer to [31].

Table 2.3: Optimal values of α computed by (2.51) with $\mathbf{u} = 2^{-53}$ and $N = 36$.

m	0	1	2	3	4	5	6	7	8	9	10	11	12
at	6.94	7.69	8.33	8.91	9.45	9.96	10.5	10.9	11.4	11.8	12.3	12.7	13.1

error can be neglected. In practice, this is not so easy to do. Second, the value of m needs to be known. This may likewise not be easy to estimate *a priori*, particularly for the complicated transforms that arise in practice.

In the next section we shall consider a numerical approach to estimate the optimal parameters that will circumvent these problems.

2.4. Numerical Algorithm

The aim of this section is to propose a numerical algorithm for selecting the optimal parameter a (or α, $\alpha = at$ for fixed t) and a sufficiently large N in the direct integration method. That is, the users only need to supply a transform $F(s)$ with a given t and a desired relative accuracy, *tol*, say. Our algorithm will automatically choose an optimal estimate of α (denoted by α_{tol}) and a sufficiently large[5] N (denoted by N_{tol}), and will attempt to compute an approximation to $f(t)$ such that its relative error remains less than or equal to the prescribed tolerance (*tol*).

To reach this aim, we propose two methods. Our first attempt for selecting α_{tol} involves minimizing the total error, $E = D + T + C$, by a univariate unconstrained optimization algorithm, such as Brent's algorithm[6]. This means we may compute the

[5] As in Section 2.3, we mean by a sufficiently large N that, to compute an approximation to $f(t)$ with N terms, the truncation error is negligible compared to the discretization error or the conditioning error; in another words, the total error is dominated either by the discretization error or by the conditioning error.

[6] Brent's algorithm is a golden section search combined with successive parabolic interpolation. It is implemented in MATLAB as the function **fmin.m**.

discretization error, the truncation error, and the conditioning error by the formulas (2.7), (2.24), and (2.28) respectively; then we shall apply Brent's algorithm to minimize E. One needs to specify an interval $[\alpha_0, \alpha_{\max}]$, and a sufficiently large N. This approach is successful provided one supplies good choices of $[\alpha_0, \alpha_{\max}]$ and N. However, it is typically difficult to predict where α_{tol} is located and how large an N to select. Moreover, this approach requires we compute many values of the total error, E, with respect to α and then find its minimum. This takes more work than the method we shall propose. We have therefore abandoned the above strategy in favor of the following approach, which seems simpler and more robust.

Before describing the idea behind the proposed algorithm, we point out that it is not necessary to compute the optimal α to high accuracy. Any estimate of the parameter α that yields an approximation to $f(t)$ that is accurate to within an order of magnitude of the best approximation will be acceptable as long as it can be computed cheaply. To this end, we have found a strategy for estimating the optimal value of α without evaluating the integral (2.7).

The idea of our algorithm is as follows: We would like to compute parameters α and N such that the computed value of $f(t)$, say $\widehat{f}(t)$, is accurate to within a prescribed relative tolerance tol, i.e.,

$$|f(t) - \widehat{f}(t)| \leq tol|f(t)|. \tag{2.52}$$

First, assume for simplicity that both the truncation and conditioning error in the direct integration method are zero (This would be the case if the infinite series in (2.4) could be summed exactly without truncation and if all computations could be done in exact arithmetic.). Then the total error would only consist of the discretization error. Assuming that m is not too large, the estimate of the discretization error is given by the second formula of (2.33), namely

$$|f(t) - \widehat{f}(t)| \approx K(t)\mathrm{e}^{-4\alpha}. \tag{2.53}$$

Here, $K(t)$ depends on t but becomes a constant for fixed t that we need to determine.

This can be done by the method of extrapolation: Suppose one computes two values of $\widehat{f}(t)$ corresponding to two distinct values of α, say $\widehat{f}_0(t)$ and $\widehat{f}_1(t)$, corresponding to α_0 and α_1 respectively. By inserting these values into Eq. (2.53), one obtains two equations, from which the (unknown) value of $f(t)$ may be eliminated, to yield

$$|\widehat{f}_1(t) - \widehat{f}_0(t)| = K(t)\left(\mathrm{e}^{-4\alpha_0} - \mathrm{e}^{-4\alpha_1}\right). \tag{2.54}$$

This provides a computable estimate for the value of $K(t)$:

$$K(t) = \log\left(\frac{|\widehat{f}_1(t) - \widehat{f}_0(t)|}{\mathrm{e}^{-4\alpha_0} - \mathrm{e}^{-4\alpha_1}}\right). \tag{2.55}$$

By putting this value of $K(t)$ into (2.53) and the result into (2.52), one obtains the following expression for achieving the desired relative tolerance:

$$tol|f(t)| \approx |K(t)|\mathrm{e}^{-4\alpha}. \tag{2.56}$$

Since $f(t)$ is unknown, we replace it by $\widehat{f}_1(t)$ (which should be a better approximation to $f(t)$ than $\widehat{f}_0(t)$, if $D(\alpha_1, t) < D(\alpha_0, t)$) and thus the estimated optimal value of α is given by

$$\alpha_{tol} = -\frac{1}{4}\log\left(tol \cdot |\widehat{f}_1(t)/K(t)|\right). \tag{2.57}$$

In the above analysis, we made the assumption that the truncation error and conditioning error are zero, which is unrealistic. If these two errors are nonzero, however, the above estimates should still be valid provided that the two errors are negligible compared to the discretization error. This can be achieved by making sure that N is sufficiently large for the discretization error to be dominant. From a graphical viewpoint, in Figure 2.1 (right) we wish to ensure that all the values of α that we have computed above are on the left (decreasing) part of the error curve, where the discretization error dominates. To achieve this, we make use of the fact that the truncation error and conditioning error depend on N, whereas the discretization error is independent of N (see Eq. (2.7)).

In order to ensure that the discretization error dominates at any required value of α (> 0), we therefore increase the value of N until the computed value of $f(t)$ no longer

changes. Once this value becomes independent of N, one can reasonably be assured that the discretization error dominates; hence, the error model (2.53) is valid.

We now discuss the strategy for selecting a sufficiently large N so that the truncation error is negligible, compared to the discretization error or the conditioning error. We first supply an initial N (denoted N_0) and then increase N in steps of J (denoted N', i.e., $N' = N + J$, $J \geq 2$). For a given fixed t and α, we compute two values $\widehat{f}_N(t)$ and $\widehat{f}_{N'}(t)$ with N and N' terms, respectively. We take the inequality

$$\left| \frac{\widehat{f}_N(t) - \widehat{f}_{N'}(t)}{\widehat{f}_{N'}(t)} \right| \leq \frac{tol}{2} \tag{2.58}$$

as our stopping criterion. If it fails, then we keep increasing N. This formula can be justified as follows.

Suppose that the discretization error dominates and N is sufficiently large so that both $\widehat{f}_N(t)$ and $\widehat{f}_{N'}(t)$ satisfy Eq. (2.53). Consequently, we can rewrite them as

$$\begin{aligned} f(t) - \widehat{f}_N(t) &= D(\alpha, t) + \delta_1, \\ f(t) - \widehat{f}_{N'}(t) &= D(\alpha, t) + \delta_2, \end{aligned}$$

for some δ_1, δ_2, where

$$D(\alpha, t) = K(t)\mathrm{e}^{-4\alpha} \quad \text{and} \quad |\delta_i| \leq \frac{1}{2} D(\alpha, t), \quad i = 1, 2.$$

Subtract the two equations above to cancel out $f(t)$ (which is unknown) and take the absolute value to yield

$$|\widehat{f}_N(t) - \widehat{f}_{N'}(t)| \leq |\delta_1| + |\delta_2| \leq D(\alpha, t).$$

By combining this with Eq. (2.56) and substituting $\widehat{f}_{N'}(t)$ for $f(t)$, we obtain

$$\left| \frac{\widehat{f}_N(t) - \widehat{f}_{N'}(t)}{\widehat{f}_{N'}(t)} \right| \leq tol.$$

In order to ensure that the above inequality is valid, we take half of the tolerance and therefore obtain the stopping criterion as (2.58). A sufficiently large N can accordingly be selected.

We should mention that the process of selecting N sometimes needs to be implemented twice, once for the computation of $\widehat{f}_1(t)$ (we use the same N to compute $\widehat{f}_0(t)$) and another for the final selection of N after α_{tol} is selected. The reason is that the initial N_0 may not be large enough for computing $\widehat{f}_i(t)$ at α_i $(i = 0, 1)$ to ensure that the discretization error dominates (i.e., (2.53) is valid). In this computation, the *tol* in Eq. (2.58) should be replaced by some number, say δ (> 0), since we only need an agreement of the two values, $\widehat{f}_{\alpha_1, N}$ and $\widehat{f}_{\alpha_1, N'}$, to within a few significant digits. We denote this selected N by N_1. Similarly, after the optimal value α_{tol} is selected, the number N_1 may not be sufficiently large for computing $\widehat{f}(t)$ at α_{tol}, when $\alpha_{tol} > \alpha_i$ $(i = 0, 1)$. In this case, we must keep increasing N until a sufficiently large N (i.e., N_{tol}) is obtained. As a result, this algorithm is summarized as follows:

Algorithm 1. *Given a transform $F(s)$ in class $\mathcal{A}$ to be inverted by the direct integration method, the user supplies a value of t where $f(t)$ is required, and a tolerance (tol) for the relative accuracy in the approximation of $f(t)$. To select the optimal value of parameter α (α_{tol}) and a sufficiently large N (N_{tol}) to reach this accuracy, the following procedure can be applied:*

1. *Give initial values N_0 and α_0, set $\alpha_1 = 2\alpha_0$, and compute the first approximation $\widehat{f}_{N_0,\alpha_1}(t)$.*
2. *Increase N in steps of J (denoted N') until Eq. (2.58) is satisfied with tol replaced by some δ (in practice, $\delta = 10^{-2}$ is good enough). Denote the selected value N by N_1.*
3. *Evaluate $\widehat{f}_{N_1,\alpha_0}(t)$ and $\widehat{f}_{N_1,\alpha_1}(t)$ and then solve for $K(t)$ from Eq. (2.55).*
4. *Compute the optimal value of α (α_{tol}) by using formula (2.54).*
5. *If $\alpha_{tol} > \alpha_1$, then increase N in steps of J until Eq. (2.58) is satisfied. Denote the selected value N by N_{tol}. Otherwise, put $N_{tol} = N_1$.*
6. *With the selected α_{tol} and N_{tol}, compute $\widehat{f}_{\alpha_{tol},N_{tol}}(t)$.*

Below, we point out a few situations where this algorithm may fail.

1. If the tolerance *tol* is too small (smaller than the conditioning error), then this method cannot reach the required accuracy. In IEEE standard double precision arithmetic ($\mathbf{u}= 2^{-53}$), we suggest that, in general, *tol* should not be less than 10^{-13}.
2. In the case when m is so large, or either α_0 or α_1 are chosen to be so small, that the first equation in (2.33) (instead of the second) is relevant, the discretization error model $K(t)\mathrm{e}^{-4\alpha}$ is invalid. In this case, we should choose α_0 and α_1 to be larger than $m/4$ ($m \geq 0$), provided m is known.

3. If either α_0 or α_1 are chosen too large so that the discretization error, $K(t)e^{-4\alpha_0}$ or $K(t)e^{-4\alpha_1}$, is less than the conditioning error, then the computed values $\widehat{f_0}(t)$ and $\widehat{f_1}(t)$ may not satisfy Eq. (2.53). We therefore suggest that α_0 and α_1 should be chosen in the interval [1, 7] (the number 7 was obtained from the $m = 0$ entry in Table 2.1).

4. The initial N (N_0) is so small that the truncation error dominates for all $\alpha > 0$. This can happen for some transforms with complex singularities and at large t. Therefore, we provide two formulas below to supply N_0.

Notes for the Implementation of Algorithm 1:

- In order to ensure that Algorithm 1 works for a large range of t, we suggest two different formulas for the initial guess N_0: one for users who have no singularity information, and the other for users who are able to provide the largest imaginary part of the rightmost singularities of the transform.

  (i) For the first set of users, we have employed the initial guesses $N_0 = 8$ and $\alpha_0 = 1.5$ in our codes. It might fail for some special cases, such as the case of large t when the transform has complex singularities[7] (see Table 2.9, the test results of Transform 5). The values of N_0 and α_0 are adjustable.

  (ii) For the second set of users, the initial guess N_0 is supplied by

$$N_0 = \begin{cases} 8, & \text{if } t \leq 16, \\ [0.32\,t\,q] + 8, & \text{if } t > 16, \end{cases} \tag{2.59}$$

  where $[x]$ is the largest integer less than or equal to x and q is the largest imaginary part of the rightmost singularities ($p + iq$, say, $q \geq 0$) of a given transform. Notice that if the transform only has real singularities ($q = 0$), N_0 is still equal to 8.

[7] In [7], they referred to t *small* (respectively *large*) in the interval (0, 15.5] (respectively (15.5, 100]). Similarly, in our algorithm, we shall use *small* t in (0, 16) and *large* t in [16, ∞).

- In the computation of N_1 and N_{tol}, we have used two iteration loops in our codes. We increase N by $N' = N + 16$ for selecting N_1 and $N' = N + 2$ for selecting N_{tol} respectively. They are adjustable by the user, but one should keep in mind that choosing J to be too large might lead to N_{tol} becoming unnecessarily large.

2.5. Numerical Tests

In this section we shall test our algorithm presented in the preceding section. Both theoretical and practical problems will be presented.

2.5.1. Theoretical Problems

We have tested our algorithm on many Laplace transforms, taken from the test problems in Davies and Martin [1], Talbot [4], Murli and Rizzardi [11], and Duffy [2]. The performance of Algorithm 1 is good for most transforms in class $\mathcal{A}$. For the sake of brevity, we only demonstrate the results of the six transforms listed in Table 2.4. This table includes transforms with only real singularities and transforms with complex singularities, as well as transforms with branch points in both cases. These functions have been used as test transforms by Murli and Rizzardi[11]. We shall also test these functions by the two other methods discussed in the next two chapters.

In this test, we evaluate $f(t)$ for a large range of t, namely $t = 10^{-1}, 1, 10, 10^2$, and 10^3, and prescribe the relative accuracy as $tol = 10^{-6}$ and 10^{-12}. The numerical results are listed in Tables 2.5–2.10. Here "ALG1" represents Algorithm 1 as stated in the preceding section; "Rel. Error" denotes the relative error of $\widehat{f}(t)$ computed at the selected parameters α_{tol} and N_{tol}; "Exact $f(t)$" denotes the exact value of $f(t)$.

One can see from Tables 2.5–2.10 that Algorithm 1 performed well for all the transforms except when underflow or overflow occurred. In Table 2.5, the selected optimal values α_{tol} and N_{tol} are all the same for different t (from 10^{-1} to 10^3) for the simple

Table 2.4: Sample Transforms.

Ex.	$F(s)$	Singular/Branch Points	$f(t)$
1	$\frac{1}{s^2}$	0	$f(t) = t$
2	$\frac{\log(s)}{s}$	0	$f(t) = -\gamma - \log(t)$, γ=Euler number
3	$\exp(-4\sqrt{s})$	0	$f(t) = 2e^{-4/t}/(t\sqrt{\pi t})$
4	$\arctan\left(\frac{1}{s}\right)$	$0, \pm i$	$f(t) = \sin(t)/t$
5	$\log\left(\frac{s^2+1}{s^2+4}\right)$	$\pm i, \pm 2i$	$f(t) = 2[\cos(2t) - \cos(t)]/t$
6	$\frac{s^2}{s^3+8}$	$-2, 1 \pm \sqrt{3}i$	$f(t) = [e^{-2t} + 2e^t \cos(\sqrt{3}t)]/3$

transform $F(s) = 1/s^2$; This is also true if $F(s) = 1/s^m$ for $m \geq 1$ according to our experiments. As for the other transforms with real singularities and/or with real branch points, the α_{tol} and N_{tol} vary within a small range for the same tolerance and are independent of t. By contrast, for the transforms with complex singularities the required N_{tol} increases as t increases. In this situation, we have to supply a larger initial guess N_0 as t increases.

Looking at Table 2.9, one can see that the results of inverting the fifth sample transform failed for large t (when $t > 50$). The reason for this is that the initial guess N_0 (= 8 by default) is not large enough to obtain a good approximation, $\widehat{f}(t)$, for large t since the truncation error dominates for all $\alpha > 0$. (This is the fourth possible case where Algorithm 1 fails, as mentioned in the preceding section; generally, $\widehat{f}_{N_0}(t)$ has to approximate $f(t)$ to at least one decimal place.) This leads to the selection of an N_1 that is not sufficiently large for computing $\widehat{f}(t)$ for large t at α_1. It follows that the selections of α_{tol} and N_{tol} fail. However, this can be easily remedied by supplying a larger value of N_0.

In our algorithm, we also designed a formula for supplying an initial guess N_0

with respect to the value of t and the maximum distance to the real axis of the rightmost singularities. For the users who are able to find the rightmost singularities of the transform, this algorithm can automatically supply a sufficiently large N_0 by the formula (2.59). In Table 2.9, we also show the new results for $t = 100$ and 1000 by supplying the rightmost singularities of the given transform and by using the improved formula (2.59) to choose N_0. One can see this succeeded in producing good results with the selected α_{tol} and N_{tol}.

We have mentioned that if the required tolerance tol is smaller than the conditioning error then this method will fail to reach the required accuracy. This can be seen in Tables 2.7–2.9 at $t = 1000$: the approximations $\widehat{f}(t)$ computed at α_{tol} and N_{tol} do not satisfy the accuracy $tol = 10^{-12}$ due to the conditioning error. Therefore those results (i.e., the relative errors of $\widehat{f}_{\alpha_{tol},N_{tol}}(t)$) are considered to be good enough, subject to the limitation of machine precision.

We also include some graphs in Figures 2.5–2.10, which display the relative errors versus the parameter $\alpha(= at)$ for inverting the transforms listed in Table 2.4. In each picture there is a pair of curves that shows the errors computed with two different tolerances, namely 10^{-6} and 10^{-12}. We also considered two different values of t. The small circle indicates the selected optimal value of α paired with the relative error computed at the selected α_{tol} and N_{tol}. The selected N_{tol} are also displayed in the picture (simply shown as N). One can see from those figures that our algorithm performed well. The relative error in each circle satisfies the prescribed tolerance to a good degree.

Table 2.5: $F(s) = 1/s^2$ and $f(t) = t$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N_{tol}	Rel.Error	α_{tol}	N_{tol}	Rel.Error	Exact $f(t)$
0.1	3.9560	10	8.81e-07	7.4099	20	6.85e-13	1.0000e-01
1	3.9560	10	8.81e-07	7.4099	20	6.88e-13	1.0000e+00
10	3.9560	10	8.81e-07	7.4099	20	6.85e-13	1.0000e+01
100	3.9560	10	8.81e-07	7.4099	20	6.94e-13	1.0000e+02
1000	3.9560	10	8.81e-07	7.4099	20	6.87e-13	1.0000e+03

Table 2.6: $F(s) = \log(s)/s$ and $f(t) = -\gamma - \log(t)$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N_{tol}	Rel.Error	α_{tol}	N_{tol}	Rel.Error	Exact $f(t)$
0.1	2.8784	12	7.61e-07	6.3323	22	6.56e-13	1.7254
1	3.8867	12	2.74e-07	7.3405	22	8.89e-13	-0.5772
10	3.6648	12	6.69e-07	7.1187	22	6.72e-13	-2.8798
100	3.6215	10	6.69e-07	7.0754	22	6.36e-13	-5.1824
1000	3.6038	10	1.17e-07	7.0564	22	6.46e-13	-7.4850

Table 2.7: $F(s) = \exp(-4\sqrt{s})$ and $f(t) = 2e^{-4/t}/(t\sqrt{\pi t})$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N_{tol}	Rel.Error	α_{tol}	N_{tol}	Rel.Error	Exact $f(t)$
0.1			underflow			underflow	
1	3.7502	12	5.23e-07	7.2041	22	7.75e-13	2.0667e-02
10	3.0304	14	6.67e-07	6.4842	22	5.94e-13	2.3919e-02
100	2.9584	10	2.84e-07	6.4122	22	1.32e-14	1.0841e-03
1000	2.9512	10	1.06e-06	6.4050	22	5.27e-12	3.5540e-05

Table 2.8: $F(s) = \arctan(1/s)$ and $f(t) = \sin(t)/t$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N	Rel.Error	α_{tol}	N	Rel.Error	Exact $f(t)$
0.1	3.5438	10	6.90e-07	6.9977	22	7.30e-13	9.9833e-01
1	3.1842	10	4.61e-07	6.6381	22	9.84e-13	8.4147e-01
10	2.9690	26	6.71e-07	6.4229	26	8.43e-13	-5.4402e-02
100	3.1316	58	6.71e-07	6.5854	58	6.90e-13	-5.0637e-03
1000	3.1960	346	6.70e-07	6.6499	350	3.72e-12	8.2688e-04

Table 2.9: $F(s) = \log\left((s^2+1)/(s^2+4)\right)$ and $f(t) = 2[\cos(2t) - \cos(t)]/t$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N_{tol}	Rel.Error	α_{tol}	N_{tol}	Rel.Error	Exact $f(t)$
0.1	3.9305	10	8.89e-07	7.3844	26	1.02e-12	-2.9875e-01
1	3.1915	12	6.67e-07	6.6454	20	2.09e-13	-1.9129e+00
10	2.5273	26	6.70e-07	5.9811	28	5.69e-13	2.4943e-01
100	3.1575	60	1.30e+00*	6.6113	60	1.30e+00*	-7.5026e-03
1000	2.8285	346	3.95e-01*	6.2824	358	3.95e-01*	-1.8597e-03
*100	3.4889	90	6.70e-07	6.9428	94	1.03e-12	-7.5026e-03
*1000	3.1950	666	6.71e-07	6.6489	680	6.04e-11	-1.8597e-03

Table 2.10: $F(s) = s^2/(s^3+8)$ and $f(t) = [e^{-2t} + 2e^t\cos(\sqrt{3}t)]/3$.

ALG1	$tol = 10^{-6}$			$tol = 10^{-12}$			
t	α_{tol}	N_{tol}	Rel.Error	α_{tol}	N_{tol}	Rel.Error	Exact $f(t)$
0.1	3.4090	10	7.01e-07	6.8629	22	6.36e-13	9.9867e-01
1	3.9714	12	3.19e-07	7.4253	24	9.52e-13	-2.4585e-01
10	3.9542	26	6.71e-07	7.4081	28	7.51e-13	6.1287e+02
100	3.4002	74	6.70e-07	6.8541	76	4.25e-13	-1.6382e+43
1000			overflow			overflow	-Inf

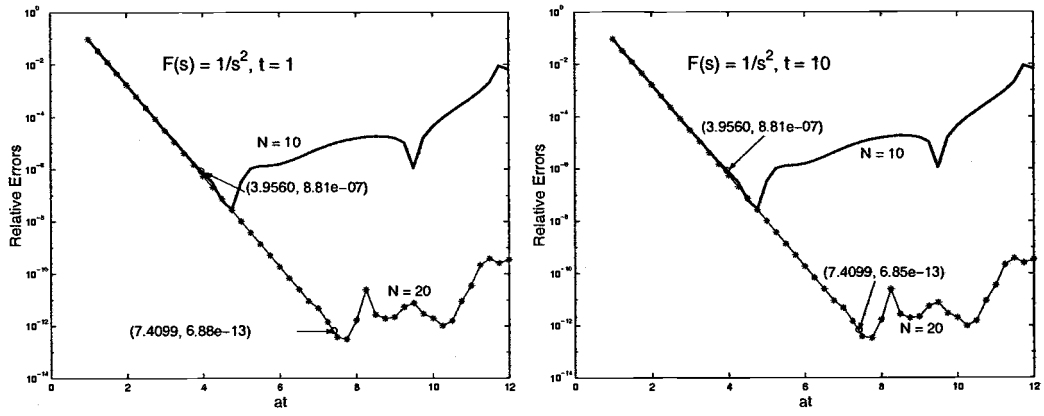

Figure 2.5: The relative error curves of inverting the transform $F(s) = 1/s^2$ at $t = 1$ and 10 with supplying $tol = 10^{-6}$ and 10^{-12}.

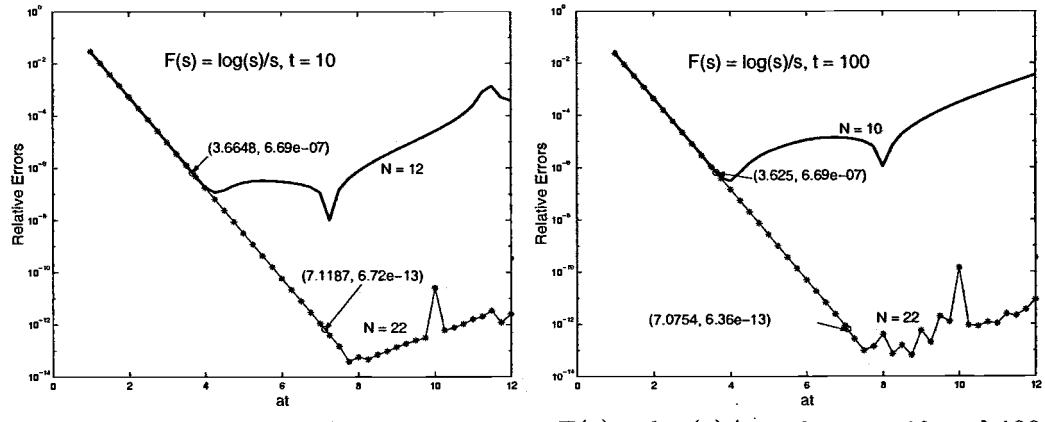

Figure 2.6: The same as Fig. 2.5 except to $F(s) = \log(s)/s$ and at $t = 10$ and 100.

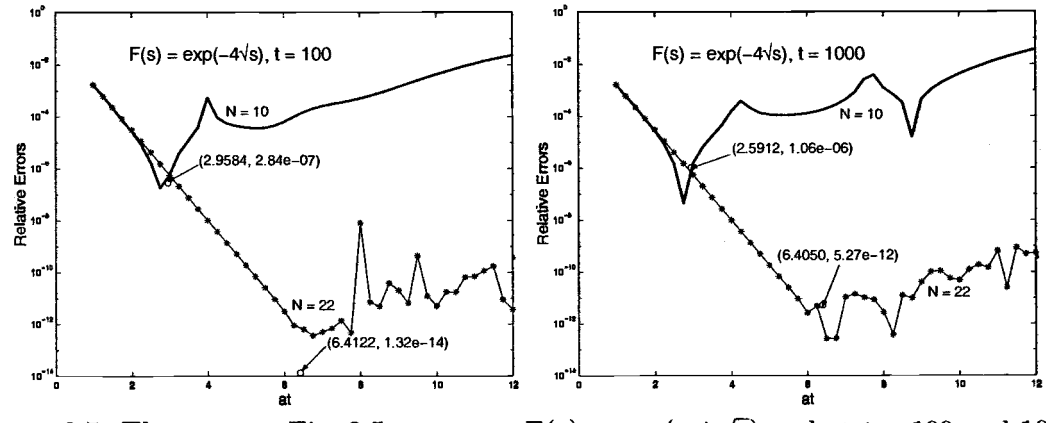

Figure 2.7: The same as Fig. 2.5 except to $F(s) = \exp(-4\sqrt{s})$ and at $t = 100$ and 1000.

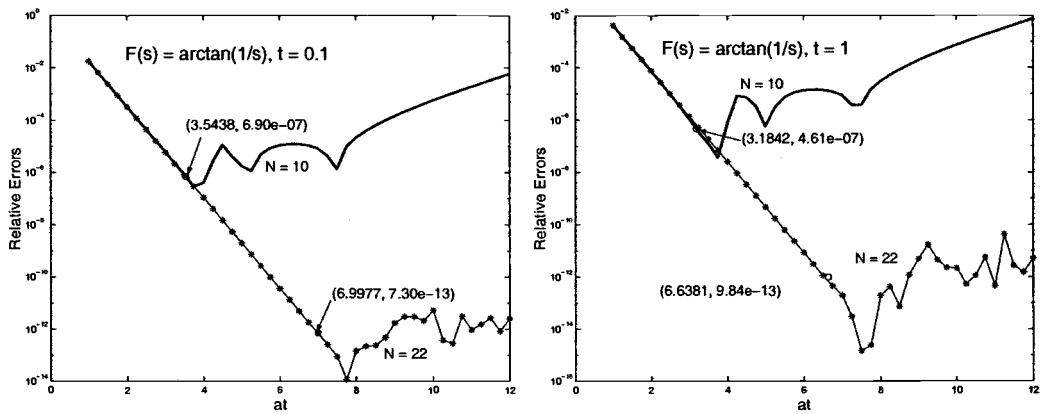

Figure 2.8: The same as Fig. 2.5 except to $F(s) = \arctan(1/s)$ and at $t = 0.1$ and 1.

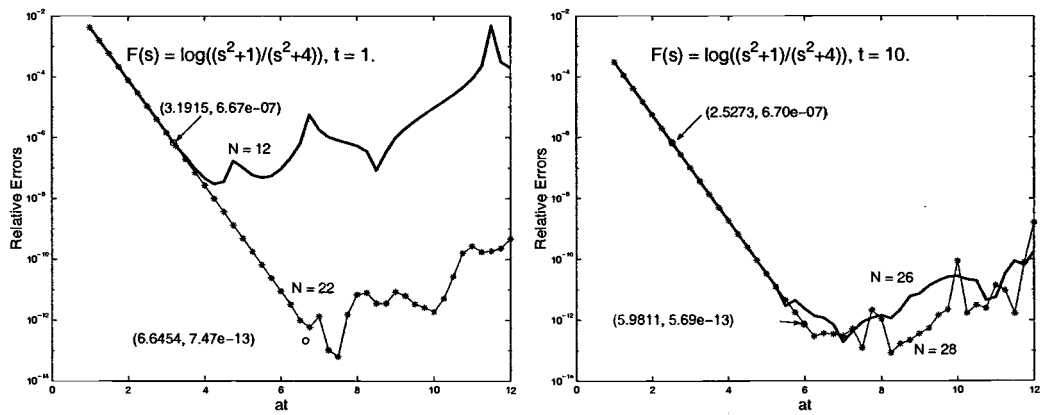

Figure 2.9: The same as Fig. 2.5 except to $F(s) = \log\left((s^2+1)/(s^2+4)\right)$ and at $t = 1$ and 10.

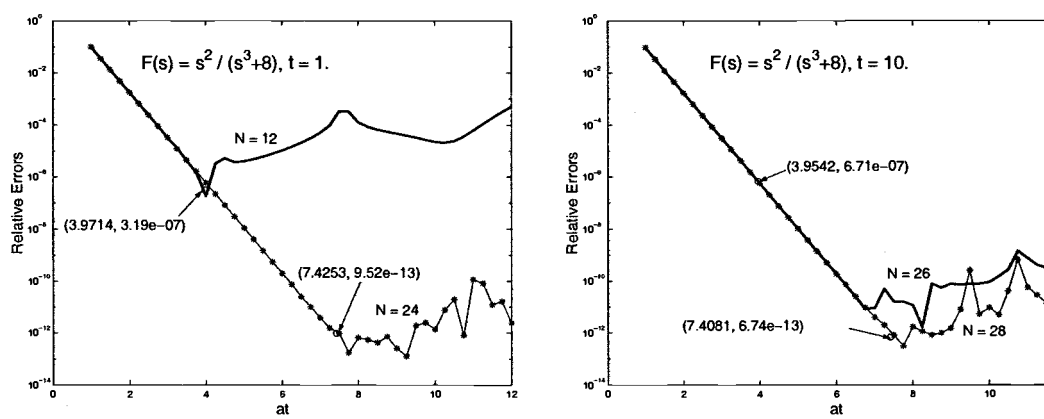

Figure 2.10: The same as Fig. 2.5 except to $F(s) = s^2/(s^3+8)$ and at $t = 1$ and 10.

2.5.2. Comparisons with D'Amore and coworkers' Algorithm

The latest work in the direct integration method (or the so-called Fourier series method) was presented by D'Amore, Laccetti, and Murli in 1999 [10, 7]. Their algorithm used the classical formula as shown in (1.4) with $h = \pi/T$ and $t \in (0, T)$. The series arising from the trapezoidal rule approximation was evaluated by a terminating continued fraction, which was computed by the q-d table. The parameter N is selected by requiring that the stopping criterion of the q-d algorithm is satisfied for the prescribed accuracy.

We first investigate the discretization error estimated by D'Amore and coworkers. They used an estimate proposed earlier by Murli and Patruno in 1978 [30], namely,

$$|D(a,t)| \leq \frac{4}{\pi} \frac{\mathrm{e}^{-dt(\psi - B)}}{(1 - \mathrm{e}^{-dt\psi})}, \tag{2.60}$$

where

$$d = (a - \sigma_0)/\gamma, \quad \psi = 2T/t = 2\pi/(ht) > 2,$$

and $\gamma > 1$ and $B > 0$ are both constants.

We now suppose the Laplace convergence abscissa σ_0 is zero (i.e., $F(s)$ is in class $\mathcal{A}$) so that $d = a/\gamma$ in (2.60). Since the values ψ, γ, and B are all constants, formula (2.60) can be simplified to

$$\begin{aligned} |D(a,t)| &\leq \frac{4}{\pi} \frac{\mathrm{e}^{(B/\gamma)at}}{(\mathrm{e}^{(\psi/\gamma)at} - 1)} \\ &= \frac{4}{\pi} \frac{\mathrm{e}^{k_1 at}}{(\mathrm{e}^{k_2 at} - 1)}, \end{aligned} \tag{2.61}$$

where both $k_1 = B/\gamma$ and $k_2 = \psi/\gamma$ are constants. This estimate may also be derived from our error representation (2.7) as follows.

Recall from Section 2.2 that the discretization error can be represented by a contour integral:

$$D(a,t) = \frac{1}{2\pi i} \int_{b-i\infty}^{b+i\infty} \frac{\mathrm{e}^{ts} F(s)}{\mathrm{e}^{-4t(s-a)} - 1} \, ds,$$

along the vertical line $\mathrm{Re}(s) = b$ for an arbitrary b with $0 < b < a$. This can be simplified to

$$D(a,t) = \frac{e^{bt}}{2\pi}\int_{-\infty}^{\infty} \frac{e^{ity}[u(b,y)+iv(b,y)]}{e^{4t(a-b)}e^{-i4ty}-1}\,dy, \tag{2.62}$$

where $u(b,y)$ and $v(b,y)$ are the real and imaginary parts of the Laplace transform $F(b+iy)$ respectively. Assume that for every given b, $0 < b < a$, the function $F(s)$ is absolutely integrable along the vertical line $\mathrm{Re}(s) = b$, that is,

$$\int_{-\infty}^{\infty} |F(b+iy)|\,dy < \infty,$$

and apply the fact that $|z-w| \geq ||z|-|w||$ to the denominator of the integrand in (2.62). The integral (2.62) is therefore bounded by

$$|D(a,t)| \leq \frac{2e^{bt}}{\pi(e^{4t(a-b)}-1)}[M_u(b)+M_v(b)], \tag{2.63}$$

where

$$M_u(b) = \int_0^{\infty} |u(b,\,y)|\,dy, \quad M_v(b) = \int_0^{\infty} |v(b,\,y)|\,dy, \quad 0 < b < a.$$

Analyzing these two integrals, D'Amore and coworkers suggested that [7]

$$M_u(b) \approx e^{-\kappa bt}, \quad \kappa > 0,$$

(a similar approximation holds for $M_v(b)$) where the constant κ depends on the Laplace transform $F(s)$. Accordingly, (2.63) can be estimated as

$$|D(a,t)| \leq \frac{4}{\pi}\frac{e^{bt(1-\kappa)}}{(e^{4t(a-b)}-1)}.$$

Suppose that b is chosen as $b = \rho a$, $0 < \rho < 1$ (in practice, $b = 0.1a$ typically gives accurate results), then we have

$$|D(a,t)| \leq \frac{4e^{\rho at(1-\kappa)}}{\pi(e^{4at(1-\rho)}-1)} = \frac{4}{\pi}\frac{e^{k_1 at}}{(e^{k_2 at}-1)}, \tag{2.64}$$

where the two constants become $k_1 = \rho(1-\kappa)$ and $k_2 = 4(1-\rho)$. This is the same formula as (2.61) estimated by D'Amore and coworkers. However, we did not use the estimate (2.64) in our algorithm since the constant κ is difficult to select to fit most transforms.

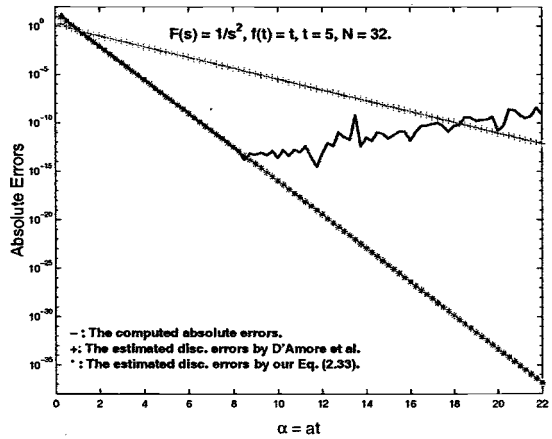

Figure 2.11: Comparison of the estimated discretization error obtained by Eq. (2.33) and Eq. (2.65) for inverting $F(s) = 1/s^2$ at $t = 5$ with $N = 32$.

Returning our attention to the estimate (2.61) of D'Amore and coworkers, with their suggested values $\psi = 4.1$, $\gamma = 1.4$, and $B = \log 10$, Eq. (2.61) can therefore be simplified to

$$\begin{aligned} |D(a,t)| &\approx \left(\frac{4 \cdot 10^{0.71at}}{\pi}\right) \mathrm{e}^{-2.93at}, \quad \text{for large } at, \\ &= \widetilde{K}(t)\mathrm{e}^{-2.93at}, \end{aligned} \tag{2.65}$$

where $\widetilde{K}(t) = 4 \cdot 10^{0.71at}/\pi$. Here we have used $1/\gamma \approx 0.71$ and $\psi/\gamma \approx 2.93$. This is similar to our error model of the discretization error (2.53). However, the estimate (2.65) is different from (2.53) by comparing the exponent and the factor $\widetilde{K}(t)$. In order to compare the accuracy of both estimates, we computed the two estimates and the actual computed (absolute) errors by this method, and show the results in Figure 2.11. One can see that our estimated discretization error is consistent with the actual computed error (in the figure the two curves overlap), but their estimate is not.

With regard to selecting the parameter a, D'Amore and coworkers required the estimated discretization error to be less than or equal to the given tolerance, which means that

$$d \leq \frac{-\log(tol)}{t(\psi - B)}$$

and thus the optimal parameter a is obtained:

$$a = \sigma_0 + \gamma d \approx \frac{-\gamma \log(tol)}{t(\psi - B)}. \tag{2.66}$$

Again, assuming that $\sigma_0 = 0$, the above expression can be simplified to read

$$\alpha_{tol} = a_{tol} t = \left(-\frac{\gamma}{\psi - B} \right) \log(tol), \tag{2.67}$$

which is similar to our estimate (2.49) in Section 2.3 with *tol* replacing **u**. Again putting their suggested values into (2.67), we obtain

$$\alpha_{tol} \approx -0.78 \log(tol). \tag{2.68}$$

If we insert $tol = \mathbf{u} = 2^{-53}$ (the roundoff error in IEEE standard double precision arithmetic) into the above formula, then $\alpha_{\mathbf{u}} \approx 28.6$, which is much different from the number 7.35 obtained from our Eq. (2.49). For comparison, we have computed the relative errors versus the parameter $\alpha(= at)$ for the first test transform in Table 2.4 and show the results in Figure 2.12. One can see that our estimated $\alpha_{\mathbf{u}} = 7.35$ is close to the real optimal point, but their estimated value $\alpha_{\mathbf{u}} = 28.6$ is far from the minimal error.

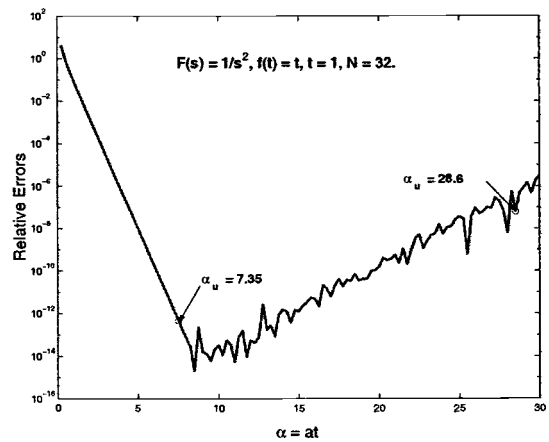

Figure 2.12: Comparison of $\alpha_{\mathbf{u}}$ obtained by Eq. (2.49) and Eq. (2.68) for inverting $F(s) = 1/s^2$ at $t = 1$ with $N = 32$.

For a further comparison, we have chosen the test transforms 1–5 listed in Table 2.4, which were also tested by D'Amore and coworkers. We evaluated the estimated values of

α by their formula (2.68) and compared them with the optimal estimates of α obtained by our numerical algorithm. For all five transforms, $\sigma_0 = 0$, and thus their estimated α_{tol} are approximately 10.76 for $tol = 10^{-6}$ and 21.52 for $tol = 10^{-12}$. However, according to our test results shown in Table 2.5–2.9, the selected optimal estimate α_{tol} range from 2.88 to 3.96 for $tol = 10^{-6}$ and from 6.33 to 7.41 for $tol = 10^{-12}$. The variation of those values mainly depends on the values of t and the individual transform. It is clear that the estimated α_{tol} by their formula are different from the estimates obtained by our algorithm.

Moreover, we also compare the number of function evaluations, N, selected by the two algorithms. The test results for the five transforms are shown in Table 2.11. The selected α_{tol} and N_{tol} required to reach the given relative error $tol = 10^{-6}$ are both displayed. One can see that the selected numbers N_{tol} by our algorithm are, on average, just about half of those used in the procedure of D'Amore and coworkers. However, in our algorithm, we do more work for parameter selection. Therefore, the actual total number of function evaluations (denoted as N_{eval}) is slightly greater than the N used in their algorithm (about 1.1 times on average).

Observing the test results further, one can see that the algorithm of D'Amore and coworkers failed on the second transform when $t = 1$, the fourth transform when $t = 50$ and $t = 100$ (in fact, when $t > 40$), and the fifth transform when $t > 5$, as marked by an asterisk. This means that their algorithm may not be suitable for large t and high accuracy. In fact, according to our implementation of their algorithm (ACM Algorithm 796 [10]), we have found that their algorithm failed to reach the required accuracy $tol = 10^{-6}$ for large values of t ($t > 40$ in general) for many transforms (Transform 4, 5, 7, 8, 11, 16, 17, 18, 20, 22, 24, 26, 28, and 29 tested in their paper [10, 7]). Obviously, it can also fail for higher tolerances.

we found several reasons for why the algorithm proposed by D'Amore and coworkers fails for large values of t and high accuracy. First, they overestimated the discretization error so that the key factor $e^{-2.93at}$ does not decrease as rapidly as the actual discretization

error (see Figure 2.11). Second, the choices of constants γ and B are not so accurate that the factor $\widetilde{K}(t)$ grows quickly (exponentially) when at increases. This may not be the case compared to the second formula in (2.33). Third, the parameter $\psi = 2\pi/ht$ must be greater than 2, that is typically difficult to preserve when t is large (h must be very small). Accordingly, their estimated parameter α does not produce high accuracy, particularly, for large value of t.

By contrast, our algorithm can evaluate the discretization error accurately as Figure 2.11 shows. As a result, our algorithm can select better estimates for the optimal α and N, which are suitable even for large values of t. Consequently, we believe that our algorithm is a significant competitor to the algorithm proposed by D'Amore and coworkers. [10, 7].

Table 2.11: Comparison of our algorithm with the one of D'Amore et al. (The asterisk (*) indicates that the algorithm failed to reach the specified tolerance).

$tol = 10^{-6}$		Algorithm 1		Alg. of D'Amore et al.	
Functions	Values of t	Rel.Error	(α_{tol}, N_{tol}, N_{eval})	Rel.Error	(α_{tol}, N_{tol})
1	1	8.81e-07	(3.96, 10, 26)	6.40e-08	(10.76, 23)
	10	8.81e-07	(3.96, 10, 26)	7.29e-08	(10.76, 23)
	50	8.81e-07	(3.96, 10, 26)	7.50e-08	(10.76, 23)
	100	8.81e-07	(3.96, 10, 26)	7.50e-08	(10.76, 23)
2	1	2.74e-07	(3.89, 12, 28)	8.50e-03*	(10.76, 467)
	10	6.69e-07	(3.67, 12, 28)	1.03e-07	(10.76, 25)
	50	6.69e-07	(3.63, 12, 28)	1.76e-08	(10.76, 25)
	100	6.69e-07	(3.62, 10, 26)	4.55e-08	(10.76, 25)
3	1	5.23e-07	(3.75, 12, 28)	3.83e-09	(10.76, 25)
	10	6.67e-07	(3.03, 14, 30)	6.26e-08	(10.76, 25)
	50	6.54e-07	(2.92, 10, 26)	4.03e-07	(10.76, 25)
	100	2.84e-07	(2.96, 10, 26)	1.98e-06	(10.76, 25)
4	1	4.61e-07	(3.18, 10, 26)	1.98e-06	(10.76, 25)
	10	6.71e-07	(2.97, 26, 42)	1.02e-06	(10.76, 33)
	50	6.87e-07	(3.47, 34, 92)	1.71e-03*	(10.76, 85)
	100	6.71e-07	(3.13, 58, 136)	1.00e-00*	(10.76, 31)
5	1	6.67e-07	(3.19, 12, 32)	3.94e-07	(10.76, 27)
	10	6.70e-07	(2.53, 26, 74)	2.11e-01*	(10.76, 71)
	50	6.71e-07	(3.75, 58, 136)	1.19e+12*	(10.76, 51)
	100	6.70e-07	(3.49, 90, 236)	1.00e-00*	(10.76, 23)

2.5.3. Practical Problems

We next apply our algorithm to the five practical problems collected by Duffy [2]; they are listed in Table 2.12 with the various application in which they occur. These transforms are representative of problems in the practical world and are challenging for several reasons. Most of the transforms contain not only a number of singularities but also branch points. The first two transforms have infinitely many poles with the poles of the first transform lying along the negative real axis, while those of the second located along the imaginary axis plus one real pole. The third transform has three real branch points in addition to a simple pole. The fourth transform has six complex branch points and one real pole. The last transforms contains no pole but five branch points located in both real and imaginary axes.

In most cases it is difficult to find the inverses of these transforms analytically. To obtain the reference solution $f(t)$ with high accuracy for each problem we appealed to a combination of numerical and symbolic techniques applied to the integral or series representations given below. For more detailed discussion of these problems, refer to [2] for references to the original papers.

Example 2..1. *The first transform in Table 2.12 is Test 2 in Duffy's collection [2]. It arises in the application of the longitudinal impact on viscoplastic rods. Its inverse can be expressed as an infinite series:*

$$f(t) = -\frac{1}{2} + \sum_{n=1}^{\infty} \left(100 + \frac{1}{x_n^2}\right) \frac{e^{-x_n^2 t}}{(2 + x_n^2)\cos(x_n/2)}. \tag{2.69}$$

This problem has a simple pole at $s = 0$, although it appears at first sight to be a branch point. There are also poles lying along the negative real axis at $s = -x_k^2$, where x_k, $k \in \mathbb{N}$, are the infinite number of real (positive) roots of the equation $x_k \tan(x_k) = 1$. Since these roots are in the left half-plane, this transform belongs to class $\mathcal{A}$. This transform was tested at the four values of $t = 0.01, 0.1, 1$, and 10, and the results are shown in Table 2.13. We also plotted the relative error curves for each individual t, as shown in

Table 2.12: Practical Problems.

Ex.	$F(s)$	Applications
1	$\frac{(100s-1)\sinh(\frac{1}{2}\sqrt{s})}{s\left[s\sinh(\sqrt{s})+\sqrt{s}\cosh(\sqrt{s})\right]}$	Viscoplastic rods
2	$\frac{1}{s(s+1)}\left[\frac{1}{2s}-\frac{1}{e^{2s}-1}\right]$	Electrical circuits
3	$\frac{1}{s}\exp\left(-\frac{1}{2}\sqrt{\frac{s(1+s)}{1+\frac{2}{5}s}}\right)$	Viscous fluids
4	$\frac{1}{s}\exp\left(-2\cosh^{-1}\sqrt{1+s^2+\left(\frac{1}{2}s\right)^4}\right)$	Shock waves in diatomic chains
5	$\frac{s-\sqrt{s^2-1}}{\sqrt{s}\sqrt{s^2-1}\sqrt{s-\frac{1}{2}\sqrt{s^2-1}}}$	Timoshenko beams

Table 2.13: The Viscoplastic Rod Problem.

Values of t		0.01	0.1	1	10
Exact $f(t)$		3.9306e-02	2.1333e+01	1.8912e+01	-4.7517e-01
$tol = 10^{-6}$	$(N,\ at)$	(12, 4.9186)	(10, 3.6170)	(12, 2.6482)	(14, 3.5666)
	Rel. Errors	8.05e-07	3.41e-07	6.89e-07	5.66e-07
$tol = 10^{-12}$	$(N,\ at)$	(22, 8.3725)	(20, 7.0708)	(22, 6.1020)	(24, 7.020)
	Rel. Errors	1.00e-12	6.69e-13	6.80e-13	1.67e-12

Figure 2.13. One can see that our algorithm performed well for this problem. Notice that there is only one case in which the required tolerance is marginally exceeded. This is when $t = 10$ with $tol = 10^{-12}$. In this case the algorithm produced a relative error 1.67×10^{-12} (see Fig. 2.13 (bottom-right)).

Example 2..2. *The second transform in Table 2.12 is Test 1 in [2]. It comes from the study of electrical circuits. This transform has a simple pole at* $s = -1$ *and pole at* $s = 0$

Table 2.14: The Electrical Circuit Problem.

Values of t		1	2	3	4
Exact $f(t)$		1.8394e-01	5.6767e-01	3.9277e-01	4.2104e-01
$tol = 10^{-6}$	$(N,\ at)$	(10, 3.7608)	(252, 3.5617)	(30, 3.5739)	(464, 3.5613)
	Rel. Errors	6.43e-07	4.87e-04	4.14e-07	4.48e-04
$tol = 10^{-12}$	$(N,\ at)$	(20, 7.2147)	(132, 7.0156)	(110, 7.0278)	(464, 7.0152)
	Rel. Errors	5.37e-13	5.31e-04	1.06e-12	4.64e-04

with multiplicity 2 in addition to an infinite number of poles on the imaginary axis. There are no singularities in the right half-plane and the transform is therefore in class $\mathcal{A}$. Its inverse is

$$f(t) = \frac{1}{2} + \left(\frac{1}{2} - \frac{e^2}{e^2 - 1}\right) e^{-t} - \frac{1}{\pi} \sum_{n=1}^{\infty} \frac{\sin[n\pi t - \tan^{-1}(n\pi)]}{n\sqrt{n^2\pi^2 + 1}}. \tag{2.70}$$

We applied our algorithm to this problem at the values $t = 1, 2, 3$, and 4, and show the results in Table 2.14. The relative error curves versus the values of at are also shown in Figure 2.14. One can see that for $t = 1, 3$ (odd values) Algorithm 1 worked well, but for $t = 2, 4$ (even values) it failed to reach the required accuracy. However, the selected optimal values of α seem correct, compared to the values computed at odd values of t. This is a difficult problem to invert, and no algorithm considered by Duffy in his test [2] could achieve high accuracy.

The phenomenon at $t = 2, 4, 6, \ldots$, may perhaps be explained by the fact that $f(t)$ is non-differentiable at these values of t, as can be deduced from the infinite series representation (2.70). The accurate inversion of this problem is an interesting research topic; therefore, we leave it as an open problem.

Example 2..3. *The third transform in Table 2.12 is Test 3 in Duffy's paper [2]. It arises in the study of a viscous fluid mechanics problem. This transform has a pole at $s = 0$ and*

Table 2.15: The Viscous Fluid Mechanics Problem (The asterisk (*) indicates the computation with $tol = 10^{-10}$ instead of $tol = 10^{-12}$).

Values of t		1/4	1	4
Exact $f(t)$		1.8394e-01	5.6767e-01	3.9277e-01
$tol = 10^{-6}$	$(N,\ at)$	(10, 3.7758)	(10, 3.6028)	(12, 3.5736)
	Rel. Errors	3.55e-07	4.40e-07	7.48e-07
$tol = 10^{-12}$	$(N,\ at)$	(20, 7.2297)	(22, 7.0566)	*(20, 7.0274)
	Rel. Errors	9.85e-13	6.68e-13	*6.59e-10

three branch points at $s = 0, -1, and\ -5/2$. *There are no singularities in the right half-plane; therefore, it can be treated as a transform in class* $\mathcal{A}$. *Its inverse can be expressed by an integral:*

$$f(t) = \frac{1}{2} + \frac{1}{\pi}\int_0^\infty \exp\left[-0.5y\sqrt{u/2}(\cos\theta - \sin\theta)\right] \times \sin\left[tu - 0.5y\sqrt{u/2}(\cos\theta + \sin\theta)\right]\frac{du}{u} \tag{2.71}$$

where

$$y = \left[\frac{1+u^2}{1+(2u/5)^2}\right]^{1/4} \quad \text{and} \quad 2\theta = \tan^{-1}(u) - \tan^{-1}(2u/5)\,.$$

We applied our algorithm to this problem at $t = 1/4$, 1, and 4, and show the results in Table 2.15 and Figure 2.15, respectively. One can see that this algorithm worked efficiently and it did reach the desired tolerance. However, notice that in the case $t = 4$, where the results are marked by asterisk, we have replaced the tolerance $tol = 10^{-12}$ with 10^{-10} since the reference solution available to us was only correct to 10 significant digits.

Example 2..4. *The fourth transform in Table 2.12 is Test 4 in Duffy's collection [2]. It is found in the study of shock waves in diatomic chains. This transform has a simple pole at* $s = 0$ *and six branch points, located at* $s = \pm 2\sqrt{2-\sqrt{3}}\,i$, $\pm 2\sqrt{2+\sqrt{3}}\,i$, *and* $\pm 4\,i$. *It has no singularities in the right half-plane and the transform therefore is in class* $\mathcal{A}$. *Its*

Table 2.16: The Shock Waves in Diatomic Chains Problem.

Values of t		1	2	4	8
Exact $f(t)$		9.8605e-02	4.3123e-01	1.0296	9.5535e-01
$tol = 10^{-6}$	$(N,\ at)$	(10, 4.1648)	(12, 3.7656)	(18, 3.5395)	(26, 3.5663)
	Rel. Errors	6.22e-07	6.72e-07	6.70e-07	6.70e-07
$tol = 10^{-12}$	$(N,\ at)$	(20, 7.6186)	(20, 7.2195)	(22, 6.9934)	(30, 7.0202)
	Rel. Errors	6.78e-13	5.63e-13	6.75e-13	9.94e-13

inverse function is given by

$$f(t) = 1 - \frac{1}{\pi}\int_0^{u_1} \frac{\sin(tu+2k)-\sin(tu-2k)}{u}\,du + \frac{1}{\pi}\int_{u_2}^{4} \frac{\sin(tu+2k)-\sin(tu-2k)}{u}\,du \tag{2.72}$$

where $\cos(k) = \frac{1}{4}\sqrt{({u_1}^2-u^2)({u_2}^2-u^2)}$, $u_1 = 2\sqrt{2-\sqrt{3}}$, *and* $u_2 = 2\sqrt{2+\sqrt{3}}$.

We applied our algorithm to this problem for the four values $t = 1, 2, 4$, and 8, and show the results in Table 2.16. The estimated optimal parameters α are reasonable since we only used small values of N. The relative error curves are also plotted in Figure 2.16.

Table 2.17: The Timoshenko Beam Problem.

Values of t		2	4	6	8
Exact $f(t)$		1.8394e-01	5.6767e-01	3.9277e-01	4.2104e-01
$tol = 10^{-6}$	$(N,\ at)$	(10, 3.4148)	(12, 3.3355)	(12, 3.3385)	(14, 3.3434)
	Rel. Errors	9.61e-07	6.88e-07	7.29e-07	7.03e-07
$tol = 10^{-12}$	$(N,\ at)$	(22, 6.8687)	(24, 6.7894)	(22, 6.7924)	(24, 6.7973)
	Rel. Errors	7.45e-13	1.01e-12	6.14e-13	6.50e-13

Example 2..5. *The last transform in Table 2.4 is Test 5 in Duffy's collection [2]. It arises in the theory of Timoshenko beams. The inverse is*

$$f(t) = \frac{2}{\pi}\int_0^1 \frac{\left(u\sqrt{(R+u)/2}+\sqrt{1-u^2}\sqrt{(R-u)/2}\right)\cosh(tu)}{R\sqrt{c^2-u^2}\sqrt{u}}\,du + \frac{2}{\pi}\int_0^{\sqrt{1/3}} \frac{(u-\sqrt{c^2+u^2})\cos(tu)}{\sqrt{u}\sqrt{c^2+u^2}\sqrt{\frac{1}{2}\sqrt{c^2-u^2}-u}}\,du \tag{2.73}$$

where $R = \sqrt{u^2+(c^2-u^2)/4}$.

This transform contains no poles but five branch points at $s = 0, \pm 1$, and $\pm i/\sqrt{3}$. It thus has one singularity in the right half-plane. We therefore need to translate the imaginary axis to the right by one unit, thus obtaining the new transform

$$F^*(s) = \frac{s+1-\sqrt{s(s+2)}}{\sqrt{s+1}\sqrt{s(s+2)}\sqrt{s+1-\frac{1}{2}\sqrt{s(s+2)}}}, \tag{2.74}$$

which has no singularity in the right half-plane and hence is in class $\mathcal{A}$.

We apply this method to $F^*(s) = F(s+1)$ and obtain an approximation of the inverse function $f^*(t)$. The actual approximation to the inverse $f(t)$ can therefore be recovered from

$$f(t) = \mathrm{e}^t f^*(t).$$

We used this strategy for values $t = 2$, 4, 6, and 8, and found that the algorithm works efficiently. The results are shown in Table 2.17, and the relative error curves are plotted in Figure 2.17.

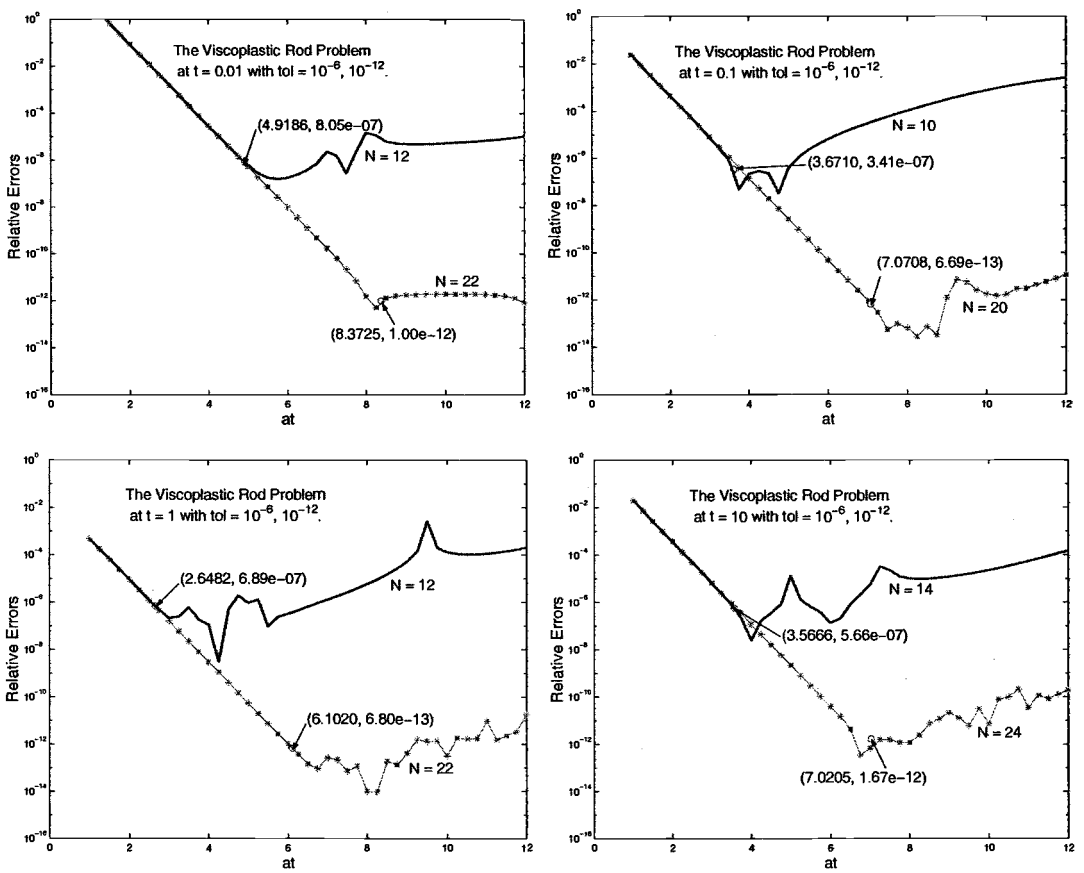

Figure 2.13: The relative error curves of inverting the viscoplastic rod problem at $t = 0.01$, 0.1, 1, and 10, computed by Algorithm 1. The tolerances are $tol = 10^{-6}$ for the upper curve and $tol = 10^{-12}$ for the lower curve.

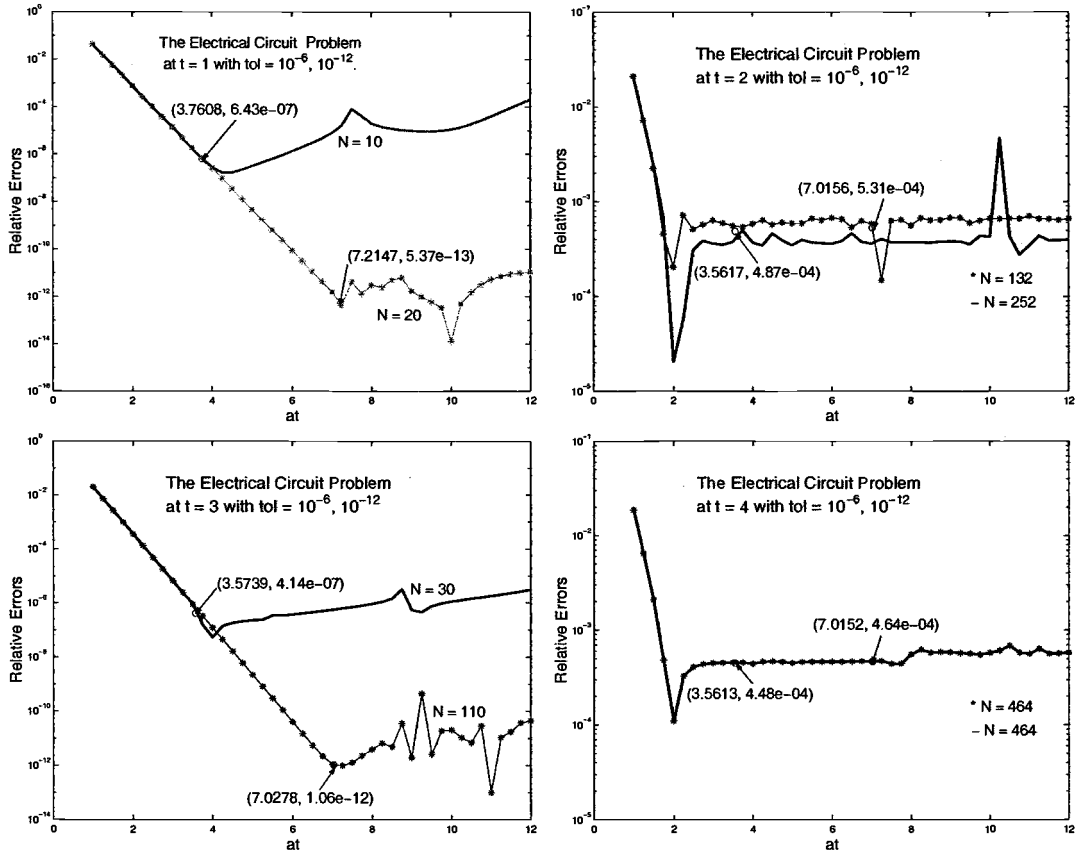

Figure 2.14: The same situation as Fig. 2.13, except for the electrical circuit problem at $t = 1$, 2, 3, and 4.

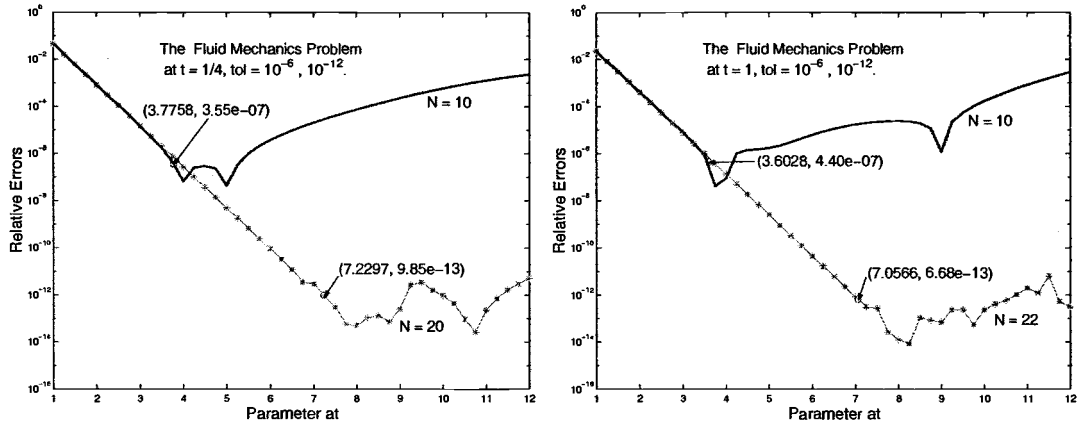

Figure 2.15: The same situation as Fig. 2.13, except for the viscous fluid mechanics problem at $t = 1/4$ and 1.

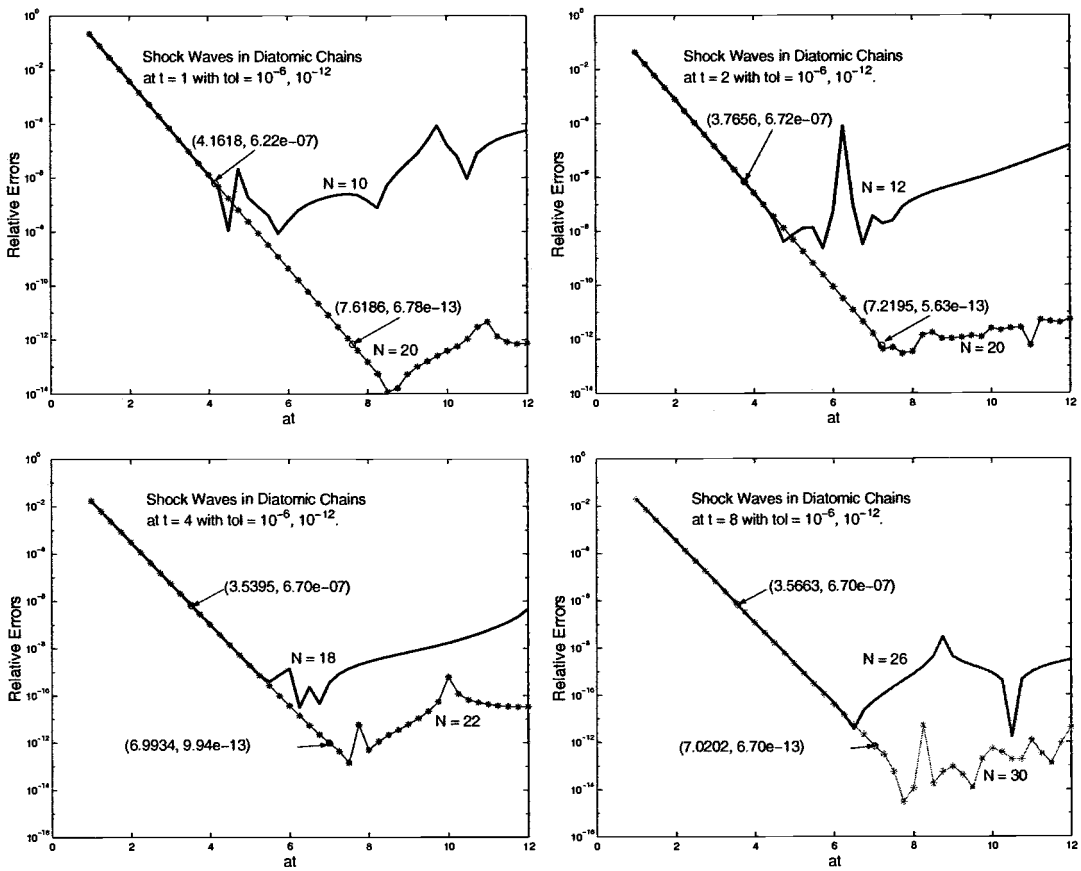

Figure 2.16: The same situation as Fig. 2.13, except for the shock waves in diatomic chains problem at $t = 1$, 2, 4, and 8.

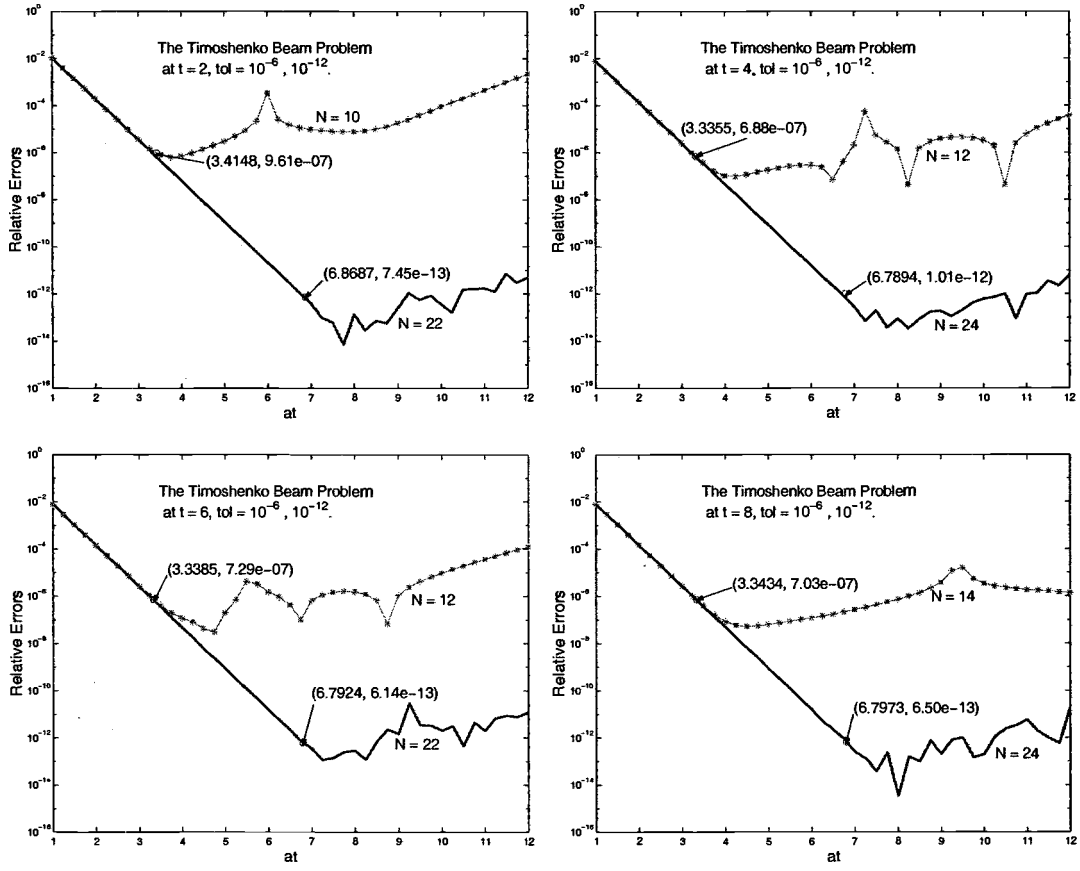

Figure 2.17: The same situation as Fig. 2.13, except for the Timoshenko beam problem at $t = 2$, 4, 6, and 8.

3. THE TALBOT METHOD

In this chapter, we turn our attention to another method, proposed by Talbot in 1979 [4]. This method consists of deforming the Bromwich line into an equivalent contour, starting and ending in the left half-plane, in such a way that $\mathrm{Re}(s) \to -\infty$ at each end. The trapezoidal or midpoint rule is then applied to the new integral. Owing to the factor e^{st} in the Bromwich integral, the integrand decays rapidly on the deformed contour as $\mathrm{Re}(s) \to -\infty$. This causes the trapezoidal or midpoint rule to converge quickly, analogous to the situation in Example 1..5 of Chapter 1.

The accuracy of Talbot's method depends critically on three geometric parameters that determine the shape of the deformed contour, as well as the number of evaluation points, N say, in the trapezoidal or midpoint rule. Talbot has proposed some general guidelines for selecting these parameters, based on various geometric, asymptotic, experimental, and heuristic analyses. Further refinements have been made by Rizzardi [12]. In this chapter, however, we propose a new numerical method for selecting the free parameters in the Talbot method.

This chapter will be organized as follows. In the first section, we shall briefly introduce Talbot's method. In the second section, we shall analyze its errors by contour integration and Fourier series expansion, and shall thereby obtain two representations of the discretization error. An algorithm for selecting the optimal parameters will be proposed in the third section. We believe it can improve the accuracy compared to Talbot's empirical formulas. Numerical tests will be given at the end.

3.1. Introduction to Talbot's Method

First, we introduce the method of Talbot. Recall that the reconstruction of the original function $f(t)$ for a given Laplace transform $F(s)$ can be expressed by the Bromwich integral:

$$f(t) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} e^{st}F(s)\,ds := \frac{1}{2\pi i}\int_{B} e^{st}F(s)\,ds, \quad t > 0.$$

Where B denotes the *Bromwich contour*, which is the vertical line from $a - i\infty$ to $a + i\infty$ with $a > \sigma_0$, and σ_0 the convergence abscissa of the Laplace transform.

When we apply numerical quadrature to evaluate $f(t)$, it may be convenient to deform contour B to an equivalent contour, L, for instance, provided no singularities are crossed. Thus, according to Cauchy's theorem, we have

$$f(t) = \frac{1}{2\pi i}\int_{L} e^{st}F(s)\,ds, \quad t > 0. \tag{3.1}$$

A special contour, introduced by Talbot, starts and ends in the left-half plane so that $\mathrm{Re}(s)$ approaches negative infinity at each end (see Figure 3.1). Accordingly, Talbot's method requires that the *convex hull* of singularities of the transform $F(s)$ be known, and that $F(s)$ must satisfy the following two conditions:

(3.a) for all the singularities s_j of $F(s)$, $|\mathrm{Im}(s_j)|$ is uniformly bounded

and

(3.b) $\lim_{s\longrightarrow\infty} |F(s)| = 0$ uniformly in $\{s \in \mathbb{C} \mid \mathrm{Re}(s) < \sigma_0\}$.

For example, the transform $1/(s(1 + \mathrm{e}^{-s}))$ does not satisfy (3.a), and e^{-5s}/s does not satisfy (3.b); see [12]. Therefore, Talbot's method is not directly applicable to these functions. We refer to any transform satisfying (3.a) and (3.b) as being of class $\mathcal{B}$ (i.e., $F(s) \in \mathcal{B}$). Notice in particular that it is not permissible for Talbot's contour to cross any branch cuts. In practice, care should be exercised when transforms involving branch-point singularities are to be implemented, as will be discussed in Example 3..2.

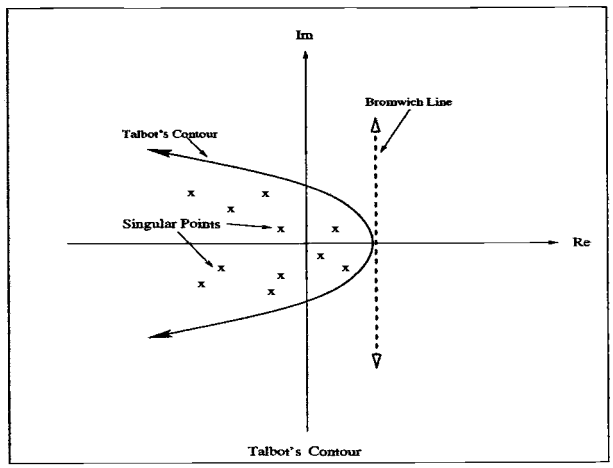

Figure 3.1: An example of Talbot's contour.

Talbot's contour is given by

$$s_\nu(z) = \frac{z}{1-e^{-z}} + \left(\frac{\nu-1}{2}\right) z, \quad z \in (-2\pi i,\, 2\pi i),$$

where ν is a positive constant. A general Talbot's contour is shown in Figure 3.1. Parameterizing this contour with $z = 2i\theta$, we obtain

$$s_\nu(\theta) = \theta \cot\theta + i\nu\theta, \quad \theta \in (-\pi,\, \pi). \tag{3.2}$$

It can be verified that the contour $s_\nu(\theta)$ approaches $-\infty$ as θ tends to $\pm\pi$ and $s_\nu(0)$ is defined as 1 (by its limit).

In order that Talbot's contour may enclose all singularities of a transform $F(s) \in \mathcal{B}$, we redefine a family of Talbot's contours by adding two more geometric parameters λ and σ as

$$s(\theta) = \lambda s_\nu(\theta) + \sigma = \lambda(\theta \cot\theta + i\nu\theta) + \sigma, \quad \theta \in (-\pi,\, \pi). \tag{3.3}$$

The parameter ν controls how wide the open mouth of Talbot's contour is (This means the contour stretches to positive or negative infinity along the vertical direction as ν increases.). The constant σ translates the contour to the right or left, and the parameter λ expands or contracts the contour. The effect of these parameters can be seen

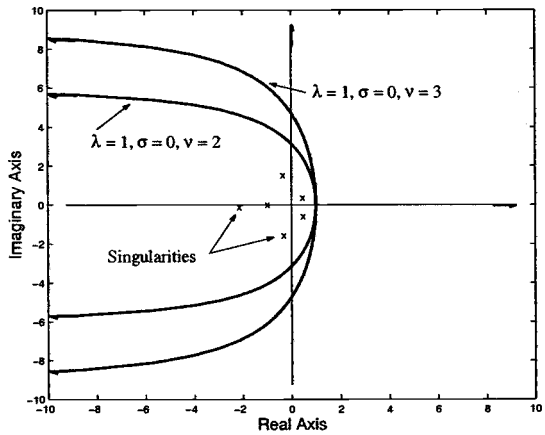

Figure 3.2: Talbot's contours with fixed $\lambda = 1$, $\sigma = 0$, but $\nu = 2$ and 3.

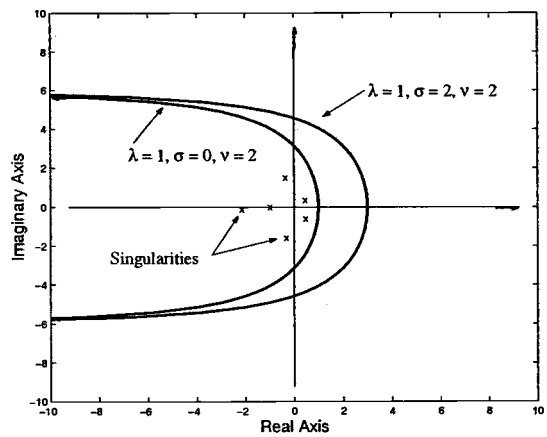

Figure 3.3: Talbot's contours with fixed $\lambda = 1$, $\nu = 2$, but $\sigma = 0$ and 2.

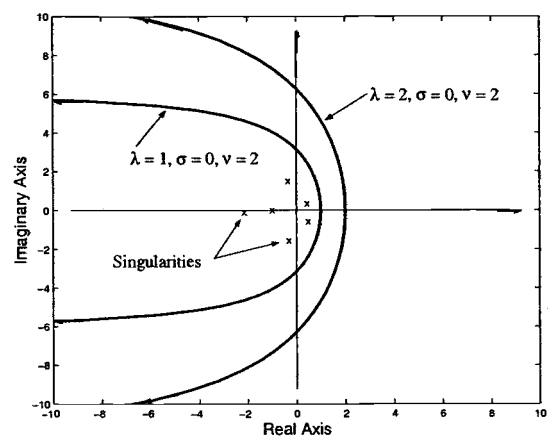

Figure 3.4: Talbot's contours with fixed $\nu = 2$, $\sigma = 0$, but $\lambda = 1$ and 2.

in Figure 3.2–3.4. We also mention that, in practice, one may fix the parameter ν for the transform which has only real singularities, since the two parameters λ and σ can be selected such that Talbot's contour always encloses all real singularities.

In general, for a given transform $F(s)$, a suitable choice of three parameters λ, σ, and ν is needed to enclose all of its singularities, and a sufficiently large N is required for reaching the user-prescribed accuracy.

The inversion formula (3.1) can now be rewritten as

$$f(t) = \frac{1}{2\pi i}\int_{-\pi}^{\pi} e^{ts(\theta)}F(s(\theta))s'(\theta)\,d\theta := \frac{1}{2\pi i}\int_{-\pi}^{\pi} Q(\theta,\, t)\,d\theta, \tag{3.4}$$

where

$$Q(\theta,\, t) := e^{ts(\theta)}F(s(\theta))s'(\theta), \tag{3.5}$$

and $s'(\theta) = \lambda(\cot\theta - \theta\csc^2\theta + i\nu)$ with $s'(0) = i\lambda\nu$ (defined by its limit). In practical computation, t is considered as a constant; therefore, for simplicity we use $Q(\theta)$ to represent $Q(\theta,\, t)$. Since $f(t)$ is a real-valued function for all positive real t, we let $U(\theta)$ and $V(\theta)$ be the respective real and imaginary parts of $Q(\theta)$, so that formula (3.4) becomes

$$f(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi} V(\theta)\,d\theta. \tag{3.6}$$

One can easily show that $U(\theta)$ is an odd function. This implies that the integral of the real part in (3.4) vanishes.

To avoid the evaluation at the end points and at $\theta = 0$, we will apply the **midpoint rule**, rather than use the trapezoidal rule, which was used by Talbot [4] and in the sequel papers and software [11]. The approximation of $f(t)$, using the midpoint rule, can therefore be expressed as

$$\widetilde{f}(t) = \frac{h}{2\pi}\sum_{j=-N}^{N-1} V(\theta_j) = \frac{1}{2N}\sum_{j=-N}^{N-1} V(\theta_j) = \frac{1}{N}\sum_{j=0}^{N-1} V(\theta_j), \tag{3.7}$$

where $h = \pi/N$ and $\theta_j = (j+1/2)h = (j+1/2)\pi/N$. This is our modified Talbot's method. As was mentioned in the introduction, this method owes its high degree of accuracy to

the fact that the magnitude of the factor $e^{ts(\theta)}$ in (3.5) decays rapidly as $\theta \longrightarrow \pm\pi$. Moreover, the domain of the function $V(\theta)$ can be extended to $[-\pi, \pi]$, with $V(\pm\pi) = 0$, and considered as a 2π-periodic function (a similar extension for $U(\theta)$). Therefore, according to Theorem 1..8, one can expect a rapid convergence of the approximation (3.7).

3.2. Error Analysis

In this section we analyze the error in Talbot's method. Our procedure is similar to the approach followed in Chapter 2 for the direct integration method. First, since the integral (3.6) is approximated by the midpoint sum (3.7), a discretization error is introduced. In contrast to the situation in the direct integration method, there is no truncation error associated with Talbot's method. This is due to the fact that the Bromwich integral (1.1), which is posed on an infinite line, has been transformed to the finite interval $(-\pi, \pi)$. No infinite sums are involved.

Consequently, the total error of Talbot's method is only modeled by the discretization error and the conditioning error. That is,

$$\text{Total Error}(E) \quad = \quad \text{Discretization Error}(D) + \text{Conditioning Error}(C).$$

The conditioning error, similar to the analysis of Section 2.2 in the direct integration method, is generated by the roundoff errors in computing the summation in (3.7). However, the magnitude of the exponential factor $e^{ts(\theta)}$ decays rapidly when $\theta \rightarrow \pm\pi$ as previously mentioned, so that the roundoff error of each summing operation is diminishing and thus becomes negligible. Moreover, the total roundoff in summing (3.7) is divided by N. When N is large, the total roundoff in Talbot's method is not as large as in the direct integration method. Accordingly, the conditioning error in Talbot's method is so small, compared to the discretization error, that it may be neglected altogether. We shall

therefore focus exclusively on the discretization error in the following analysis.

3.2.1. The Discretization Error

To analyze the discretization error of Talbot's method, we shall apply the same idea as in Theorem 2.1, thus representing the discretization error by a contour integral. This idea is also based on Theorem 1..8 and its corollary regarding the error representation of the trapezoidal-type rules for an integral with a 2π-periodic and analytic integrand. We summarize this approach in the following theorem.

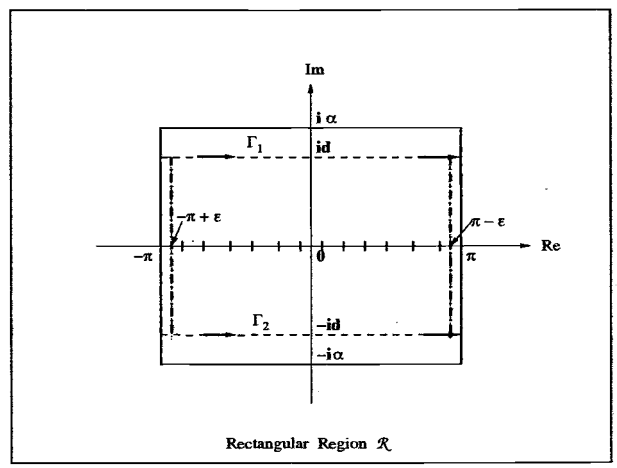

Figure 3.5: The contour used in the proof of Theorem 1..3

Theorem 3..1. *Suppose that the original function $f(t)$ of a Laplace transform $F(s)$ can be reconstructed by the integral (3.6) along a Talbot contour (3.3) for all $t > 0$, and its approximation $\widetilde{f}(t)$, approximated by the midpoint rule with step size $h = \pi/N$, can be expressed as*

$$\widetilde{f}(t) = \frac{1}{2N} \sum_{j=-N}^{N-1} V(\theta_j), \qquad V(\theta) = \mathrm{Im}\left\{ e^{ts(\theta)} F(s(\theta)) s'(\theta) \right\}. \tag{3.8}$$

Then the discretization error, $D = \widetilde{f}(t) - f(t)$, of Talbot's method can be represented by the integral:

$$D = -\frac{1}{2\pi} \left\{ \int_{-\pi}^{\pi} \frac{V(x+id)}{\mathrm{e}^{-i2N(x+id)} + 1}\, dx + \int_{-\pi}^{\pi} \frac{V(x-id)}{\mathrm{e}^{i2N(x-id)} + 1}\, dx \right\}, \tag{3.9}$$

where d (> 0) is some constant.

Proof: Based on the idea of the proof of Theorem 1..8, we observe that the integrand $V(\theta)$ is 2π-periodic and analytic on $(-\pi, \pi)$. According to Theorem 1..8, there exists a rectangular region $\mathcal{R} = [-\pi + \epsilon, \pi - \epsilon] \times (-\alpha, \alpha)$ in the complex-plane with $\alpha > 0$ and $\epsilon > 0$ (sufficiently small), such that $V(\theta)$ can be extended to an analytic and 2π-periodic bounded function on the region $\mathcal{R}$. Now considering the approximate inversion formula (3.8) as the sum of the residues of $-\tan(Nw)V(w)/2$ we have

$$\begin{aligned}
\widetilde{f}(t) &= \frac{1}{2N}\sum_{j=-N}^{N-1} V(w_j) \\
&= \sum_{j=-N}^{N-1} \mathrm{Res}\left(-\frac{1}{2}\tan(Nw)V(w),\ \ w_j = \frac{(j+1/2)\pi}{N}\right) \\
&= \frac{1}{4\pi i}\left(\int_{\Gamma_1} - \int_{\Gamma_2}\right)\tan(Nw)V(w)\,dw, \qquad (3.10)
\end{aligned}$$

Since $V(w)$ approaches zero when $\mathrm{Re}(w)$ tends to $\pm\pi$, the integrals along the vertical line segments, $\mathrm{Re}(w) = \pm(\pi - \epsilon)$, vanish. Here Γ_1 denotes the path from $id - (\pi - \epsilon)$ to $id + (\pi - \epsilon)$ and Γ_2 is from $-id - (\pi - \epsilon)$ to $-id + (\pi - \epsilon)$, $0 < d < \alpha$ (see Figure 3.2.1.).

Since $V(w)$ is analytic inside the region $\mathcal{R}$, by Cauchy's theorem the integral (3.4) is identical to the integral along either of the contours Γ_1 or Γ_2. That is,

$$\begin{aligned}
f(t) &= \frac{1}{2\pi}\int_{-\pi}^{\pi} V(\theta)\,d\theta = \frac{1}{2\pi}\int_{\Gamma_1} V(w)\,dw = \frac{1}{2\pi}\int_{\Gamma_2} V(w)\,dw, \\
&= \frac{1}{4\pi}\left(\int_{\Gamma_1} + \int_{\Gamma_2}\right) V(w)\,dw. \qquad (3.11)
\end{aligned}$$

Therefore, by (3.10) and (3.11), we obtain

$$\begin{aligned}
D &= \widetilde{f}(t) - f(t) \\
&= -\frac{1}{4\pi}\left(\int_{\Gamma_1}(1 + i\tan(Nw))V(w)\,dw + \int_{\Gamma_2}(1 - i\tan(Nw))V(w)\,dw\right) \\
&= -\frac{1}{2\pi}\left(\int_{id-\pi}^{id+\pi}\frac{V(w)}{e^{-i2Nw}+1}\,dw + \int_{-id-\pi}^{-id+\pi}\frac{V(w)}{e^{i2Nw}+1}\,dw\right), \qquad (3.12)
\end{aligned}$$

which is equal to the formula (3.9). Hence the proof is completed. □

In numerical work, one may use formula (3.9) to compute approximations of the discretization error. However, this requires a good choice for the value of d, which is not always easy to select. We therefore prefer the following approach.

Recall the error representations of the direct integration method, where we have seen that the discretization error can be represented by both a contour integral and a series representation resulting from a Fourier series approach. The same can be done for Talbot's method.

Theorem 3..2. *The discretization error of Talbot's method associated with the midpoint rule can also be represented as:*

$$D = \sum_{k=1}^{\infty}(-1)^k(a_{-2kN} + a_{2kN}) \tag{3.13}$$

where

$$a_n = \frac{1}{2\pi}\int_{-\pi}^{\pi} V(\theta)\mathrm{e}^{-in\theta}\,d\theta. \tag{3.14}$$

Moreover, this series representation for the discretization error is identical to the representation of the contour integral (3.9) in Theorem 3..1.

Proof. Recall that the function $V(\theta)$ is well-defined on the interval $[-\pi, \pi]$ and analytic in any open neighborhood of $(-\pi, \pi)$. Also, it can be shown that $V(\theta)$ is piecewise smooth on its domain. Therefore, $V(\theta)$ is integrable and has the *Fourier series expansion*

$$V(\theta) = \sum_{n=-\infty}^{\infty} a_n\mathrm{e}^{in\theta}, \quad \theta \in [-\pi, \pi], \tag{3.15}$$

where the coefficients a_n are defined by (3.14). Substituting the series (3.15) for $V(\theta)$ into the inversion formula (3.6) and using the fact that term by term integration is permissible because of analyticity of $V(\theta)$, we have

$$f(t) = \frac{1}{2\pi}\int_{-\pi}^{\pi}\left(\sum_{n=-\infty}^{\infty} a_n\mathrm{e}^{in\theta}\right)d\theta = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} a_n\int_{-\pi}^{\pi}\mathrm{e}^{in\theta}\,d\theta = a_0. \tag{3.16}$$

Also, plugging the series (3.15) into the approximate inversion (3.7) and interchanging the summations, we obtain

$$\widetilde{f}(t) = \frac{1}{2\pi}\sum_{j=-N}^{N-1} V(\theta_j)h = \frac{1}{2\pi}\sum_{n=-\infty}^{\infty} a_n\left(h\sum_{j=-N}^{N-1}\mathrm{e}^{in\theta_j}\right), \tag{3.17}$$

where $\theta_j = (j + 1/2)h$. We now sum the finite geometric series inside the parentheses as follows:

$$\begin{aligned} h\sum_{j=-N}^{N-1} \mathrm{e}^{in\theta_j} &= h\mathrm{e}^{inh/2}\sum_{j=-N}^{N-1} \mathrm{e}^{inhj} \\ &= h\frac{\mathrm{e}^{-inhN} - \mathrm{e}^{inhN}}{\mathrm{e}^{-inh/2} - \mathrm{e}^{inh/2}} \\ &= \frac{\pi \sin(n\pi)}{N\sin(n\pi/2N)}, \quad \text{if } n \neq 2kN,\ k = 0, \pm 1, \pm 2, \ldots . \end{aligned} \tag{3.18}$$

We therefore have

$$h\sum_{j=-N}^{N-1} e^{in\theta_j} = \begin{cases} (-1)^k 2\pi, & \text{if } n = 2kN,\ \forall\, k = 0, \pm 1, \pm 2, \ldots, \\ 0, & \text{otherwise.} \end{cases}$$

Putting the above result back into (3.17), we obtain the series representation for the approximate inversion

$$\begin{aligned} \widetilde{f}(t) &= a_0 + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} (-1)^k a_{2kN} \\ &= f(t) + \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} (-1)^k a_{2kN}, \end{aligned} \tag{3.19}$$

and the discretization error hence is

$$\begin{aligned} D = \widetilde{f}(t) - f(t) &= \sum_{\substack{k=-\infty \\ k\neq 0}}^{\infty} (-1)^k a_{2kN} \\ &= \sum_{k=1}^{\infty} (-1)^k (a_{-2kN} + a_{2kN}). \end{aligned}$$

Next, we want to show that the two representations (3.9) and (3.13) for the discretization error are identical. Following the same idea as the proof of Theorem 3..1, we consider each integral in (3.13):

$$a_{-2kN} = \int_{-\pi}^{\pi} V(\theta)\mathrm{e}^{i2kN\theta}\, d\theta, \quad a_{2kN} = \int_{-\pi}^{\pi} V(\theta)\mathrm{e}^{-i2kN\theta}\, d\theta \tag{3.20}$$

for $k = 1, 2, \ldots$, as a path integral in the complex plane along a contour starting at $-\pi + \epsilon$ and ending at $\pi - \epsilon$ inside the region $\mathcal{R}$, in which $V(w)$ is analytic. Therefore, the

contours Γ_1 and Γ_2 can also be used in these integrals, and thus, by Cauchy's theorem, these integrals may be expressed as

$$a_{-2kN} = \int_{\Gamma_1} V(w)\mathrm{e}^{i2kNw}\,dw = \int_{id-\pi}^{id+\pi} V(w)\mathrm{e}^{i2kNw}\,dw \tag{3.21}$$

and

$$a_{2kN} = \int_{\Gamma_2} V(w)\mathrm{e}^{-i2kNw}\,dw = \int_{-id-\pi}^{-id+\pi} V(w)\mathrm{e}^{-i2kNw}\,dw. \tag{3.22}$$

The equalities (3.21) and (3.22) hold since again $V(w)$ approaches zero as $\mathrm{Re}(w)$ tends to $\pm\pi$, so that the integrals along the vertical line segments, $\mathrm{Re}(w) = \pm(\pi - \epsilon)$, vanish. Hence, the discretization error (3.13) can be rewritten as

$$\begin{aligned} D &= \frac{1}{2\pi}\sum_{k=1}^{\infty}(-1)^k\left(\int_{\Gamma_1} V(w)\mathrm{e}^{i2kNw}\,dw + \int_{\Gamma_2} V(w)\mathrm{e}^{-i2kNw}\,dw\right). \\ &= \frac{1}{2\pi}\sum_{k=1}^{\infty}(-1)^k\left\{\int_{-\pi}^{\pi} V(x+id)\mathrm{e}^{i2kN(x+id)}\,dx + \int_{-\pi}^{\pi} V(x-id)\mathrm{e}^{-i2kN(x-id)}\,dx\right\}. \end{aligned} \tag{3.23}$$

Observing that, for $d > 0$, $|\mathrm{e}^{2N(-d\pm ix)}| = |\mathrm{e}^{-2Nd}| < 1$, we see that both geometric series converge uniformly:

$$\sum_{k=0}^{\infty}(-\mathrm{e}^{-2Nd\pm i2Nx})^k = \frac{1}{1+\mathrm{e}^{-2N(d\pm ix)}} \tag{3.24}$$

for all $x \in [-\pi+\epsilon,\ \pi-\epsilon]$. Again, since $V(w)$ is analytic inside the region $\mathcal{R}$, interchanging the summation and integration is permissible. As a result, we obtain

$$\begin{aligned} D &= \frac{1}{2\pi}\int_{-\pi}^{\pi} V(x+id)\sum_{k=1}^{\infty}(-\mathrm{e}^{-2Nd+i2Nx})^k\,dx \\ &\quad + \frac{1}{2\pi}\int_{-\pi}^{\pi} V(x-id)\sum_{k=1}^{\infty}(-\mathrm{e}^{-2Nd-i2Nx})^k\,dx \\ &= -\frac{1}{2\pi}\int_{-\pi}^{\pi}\frac{V(x+id)\mathrm{e}^{i2N(x+id)}}{1+\mathrm{e}^{i2N(x+id)}}\,dx - \frac{1}{2\pi i}\int_{-\pi}^{\pi}\frac{V(x-id)\mathrm{e}^{-i2N(x-id)}}{1+\mathrm{e}^{-i2N(x-id)}}\,dx \\ &= -\frac{1}{2\pi}\left\{\int_{-\pi}^{\pi}\frac{V(x+id)}{\mathrm{e}^{-i2N(x+id)}+1}\,dx + \int_{-\pi}^{\pi}\frac{V(x-id)}{\mathrm{e}^{i2N(x-id)}+1}\,dx\right\}, \end{aligned} \tag{3.25}$$

which is the same as the representation (3.9). Hence the proof is completed. □

3.2.2. Error Approximation

We have mentioned that the total error of Talbot's method is the sum of the discretization error and the conditioning error, and that the conditioning error is negligible. Accordingly, to estimate the total error of this method we only need to evaluate the discretization error. Since we have two error representations, (3.9) and (3.13), for the discretization error, it can be approximated by computing the corresponding integrals (see Figure 3.7 and 3.8). However, to evaluate (3.9) accurately requires a suitable choice for d, which would greatly increase the work. We therefore prefer to adopt formula (3.13) for the discretization error in our algorithm.

Since both sequences $\{a_{-2kN}\}_{k=1}^{\infty}$ and $\{a_{2kN}\}_{k=1}^{\infty}$ in (3.13) converge rapidly, owing to the analyticity of $V(\theta)$, the discretization error can be approximated by the first two terms. That is,

$$\begin{aligned} D &= \sum_{k=1}^{\infty}(-1)^k(a_{-2kN}+a_{2kN}) \\ &\approx -(a_{-2N}+a_{2N}) \\ &= -\frac{1}{2\pi}\left(\int_{-\pi}^{\pi} V(\theta)\mathrm{e}^{i2N\theta}\,d\theta + \int_{-\pi}^{\pi} V(\theta)\mathrm{e}^{-i2N\theta}\,d\theta\right). \end{aligned} \tag{3.26}$$

To approximate the above two integrals, one may apply the **midpoint rule** once again, with supplying suitable choices for the three parameters λ, σ, and ν, and the number of function-evaluations, N, to obtain

$$D \approx -\frac{h'}{2\pi}\left(\sum_{j=-M}^{M-1} V(x_j)\mathrm{e}^{-i2Nx_j} + \sum_{j=-M}^{M-1} V(x_j)\mathrm{e}^{i2Nx_j}\right) \tag{3.27}$$

where $h' = \pi/M$ and $x_j = (j+1/2)h' = (j+1/2)\pi/M$. Note that in order to evaluate the two integrals in (3.26) accurately, the number M must be substantially greater than N, since both integrals are highly oscillatory with frequency π/N. In practical tests, the choice of $M = 4N$ gave accurate results (see Figure 3.8). Similarly, to approximate the error formula (3.9), one may apply the same strategy of computation.

In the next example, we present numerical evidence that the error formulas (3.9) and (3.13) are reliable predictors of the actual error in the Talbot method.

Example 3..1. *Consider the Laplace transform*

$$F(s) = \mathrm{e}^{-4\sqrt{s}}, \tag{3.28}$$

which has a branch point at 0. Its inversion is

$$f(t) = \frac{2\mathrm{e}^{-4/t}}{t\sqrt{\pi t}}, \quad t > 0.$$

In this example, we have plotted the level curves of the computed (absolute) errors, as computed by Talbot's method (3.7), shown in Figure 3.6. In this figure, we have used $t = 10$, $N = 16$, $\nu = 1$, and the values of (λ, σ) in the rectangle $[0.1,\ 1.5] \times [-1,\ 1.5]$. The labelling of the level curves indicate base 10 logarithm of the computed errors. For instance, -12 means that the magnitude of the absolute errors is 10^{-12} on that curve. In Figure 3.7 and 3.8, we represent the approximations to the true error, as computed by formulas (3.9) and (3.27), respectively. In these two figures, we have used the midpoint rule with the same $h = \pi/M$ and $M = 4N$, in addition to $d = 0.2$, for computing (3.9).

Comparing the three pictures, it is evident that the error curves are almost the same, except that in Figure 3.8 the approximated error curves computed by (3.27) have some unexpected results on the bottom-left corner. This phenomenon can be explained by noting that those contours correspond to unsuitable parameter choices (the contours are very close to the negative real axis), so that the computation of the discretization error by (3.27) is no longer valid[8]. Substantially, we can conclude that (3.9) and (3.27) are reliable and practical error estimators, at least for this example. We will therefore proceed to use them in our algorithm for finding the optimal parameters in the next section.

We conclude this section with another example to show that care should be exercised when applying Talbot's method to the transforms with branch-point singularities.

[8]This phenomenon underscores the fact that, in order to converge when an optimization algorithm is applied, a good choice of initial parameters (λ, σ, ν) is necessary.

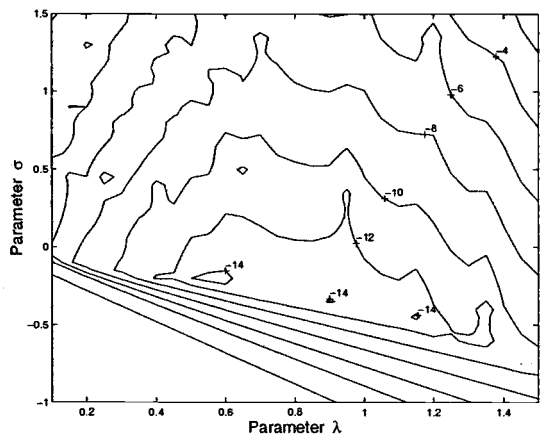

Figure 3.6: The error curves computed by Talbot's method for the transform (3.28).

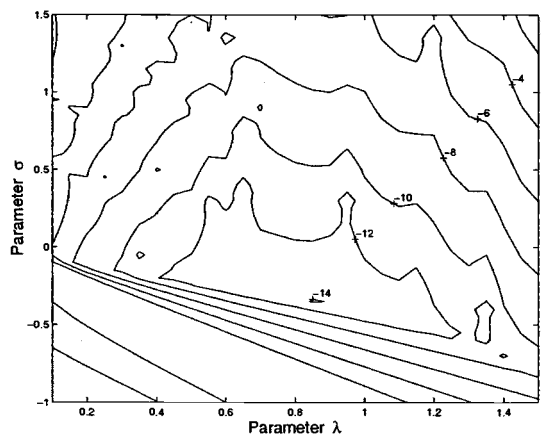

Figure 3.7: The error curves approximated by the formula (3.9) with $d = 0.2$.

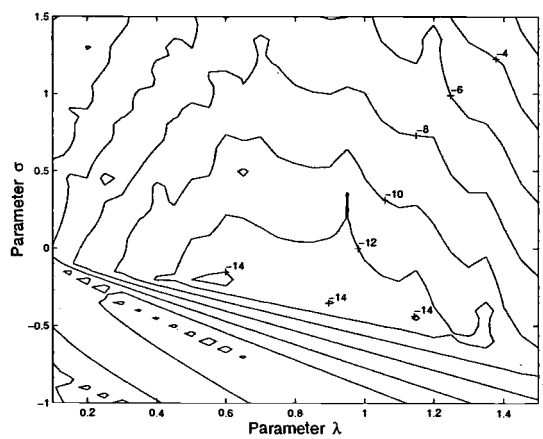

Figure 3.8: The error curves approximated by the formula (3.27) with $M = 4N$.

Example 3..2. *The Laplace transform*

$$F(s) = \frac{1}{\sqrt{s^2+1}} \tag{3.29}$$

has branch points at $\pm i$. *Its inversion is the Bessel function* $f(t) = J_0(t)$.

In MATLAB, the branch cut of the square root is defined as the negative real axis, which implies that the branch cuts of $F(s)$ are line segments on the imaginary axis $|Im(z)| > 1$, as shown in Figure 3.9. Therefore, Talbot's contour cannot avoid crossing these branch cuts. A direct application of Talbot's method to the transform (3.29) will therefore yield incorrect results. If we rewrite (3.29) as

$$F(s) = \frac{1}{\sqrt{s+i}\sqrt{s-i}}, \tag{3.30}$$

then its branch cuts are two rays parallel to the negative real axis (see Figure 3.10). By this process, Talbot's contour avoids crossing the branch cuts, provided the parameters λ, σ and ν are sufficiently large. This method therefore yields the high accuracy to the transform (3.30), shown in Figure 3.11, which was computed at $t = 5$ with $\nu = 1$ and $N = 16$.

3.3. An Algorithm for Selecting Parameters

It has been demonstrated in Figure 3.6 and 3.11 that Talbot's method can reach high accuracy with moderate N, provided that a suitable choice of Talbot's contour is made. Talbot has presented a strategy for parameter selection. It is based on geometric approaches, asymptotic estimates, and many experimental results, which imply that the selected parameters may not be truly optimal. Since we have an error representation (3.9), we can estimate the total error numerically and then apply a multivariate optimization algorithm to compute the optimal parameters. In this section, we shall present such a numerical algorithm for selecting the optimal parameters in Talbot's method.

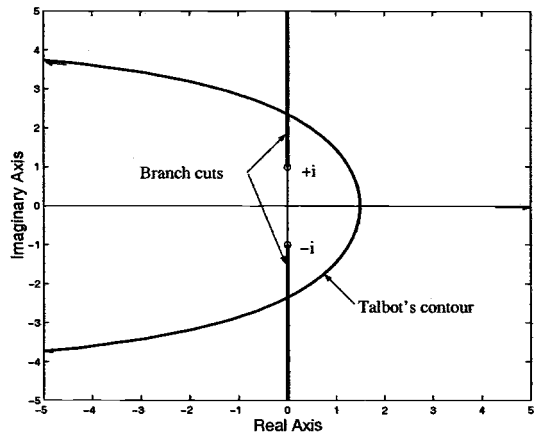

Figure 3.9: Branch cuts for the transform (3.29).

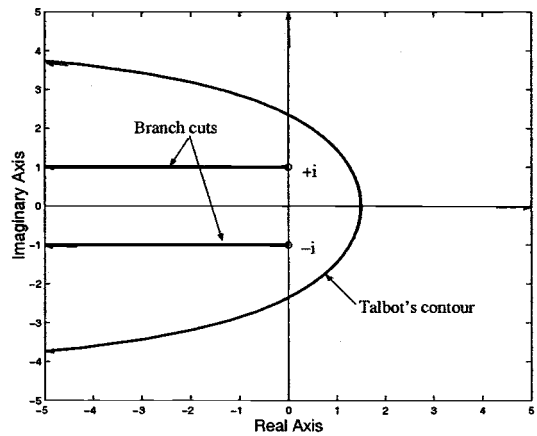

Figure 3.10: Branch cuts for the transform (3.30).

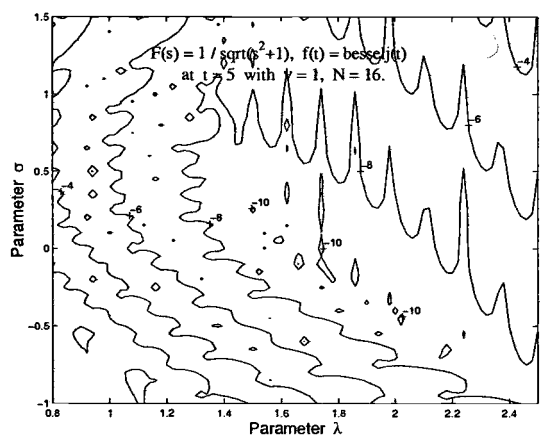

Figure 3.11: The error curves computed by Talbot's method for the transform (3.30).

Many optimization algorithms may be applied to this problem. We tested some popular maximizers/minimizers built into MATLAB, such as the multi-directional search (**mdsmax.m**), the Nelder-Mead simplex method (**nmsmax.m**), a genetic algorithm (**genetic.m**), and also our own three dimensional grid maximizer. We have found that the method of alternating directional search (**adsmax.m**) is efficient and stable for most transforms and for a large range of t. Therefore, we have adopted this algorithm for the purpose of computing the optimal parameters.

In our proposed numerical algorithm for parameter selection, it is necessary to supply a suitable initial guess for all parameters, including the accuracy parameter N, the number of transform evaluations. Talbot's empirical formulas are useful for finding an initial guess, and will be utilized in our algorithm. The basic idea of Talbot's prescription for parameter selection involves finding the dominant singularity[9]. With the dominant singularity and geometric analysis, in addition to numerical experiments, a suitable Talbot's contour can be determined. That is, the parameters λ, σ, and ν can be obtained.

As to the selection of the number N, it can be determined by combining the error estimates and the empirical formulas, provided the users supply the desired accuracy (tol) as well as the machine precision (Talbot was also concerned about the roundoff error). Since Talbot's empirical formulas are very lengthy, we do not reproduce them here but refer to the paper [4]. We consequently summarize our algorithm as follows.

[9]The dominant singularity (denoted as $s_d = p_d + iq_d$) is the most significant singularity of a transform $F(s)$. It is defined as follows: Let the singularities of $F(s)$ be $s_j = p_j + iq_j$, and let $\hat{p} = \max_j(p_j)$ and $\sigma_0 = \max(0, \hat{p})$. Applying a shift σ_0 to $F(s)$, we have $s_j{}' = s_j - \sigma_0 = p_j{}' + iq_j$, $p_j{}' = p_j - \sigma_0 \leq 0$. Then the dominant singularity s_d is the one having the greatest ratio:

$$\frac{q_d}{\theta_d} = \max_{q_j > 0} \frac{q_j}{\theta_j}, \quad \theta_j = \arg(s_j{}').$$

Algorithm 2. *Given a transform $F(s)$ in class $\mathcal{B}$ with known convex-hull of singularities and a value of t, to select the optimal parameters λ, σ, ν for finding the inverse function $f(t)$ by Talbot's method, the following procedure can be applied:*

1. *Supply the desired relative accuracy (tol) and the machine precision (ε_m), input the singularities with their multiplicities, and then apply Talbot's formulas to get a set of the initial values λ, σ, ν and N. Also, set $M = 4N$.*

2. *Evaluate the discretization error with the values λ, σ, ν, and M (obtained from Step 1) by using (3.27) to obtain an estimate for the error.*

3. *Apply the alternating directional search optimization routine (**adsmax.m**) to find the improved parameters λ, σ, and ν that minimize the error. (In order to search the minimum error, the optimization routine will repeat the computation in step 2 with new parameters.)*

4. *With the selected λ, σ, ν, and the original N (selected in Step 1), compute the approximation $\widetilde{f}(t)$ using (3.7).*

This is our improved algorithm for the Talbot method. It can achieve better results than the original Talbot's algorithm for most Laplace transforms. We shall present numerical tests in the next section.

3.4. Numerical Tests

We have tested our algorithm on many Laplace transforms, taken from the test problems in Davies and Martin [1], Talbot [4], Murli and Rizzardi [11], and Duffy [2]. The performance of Algorithm 2 is good for most transforms in class $\mathcal{B}$, as we now proceed to show. We start our comparison by considering the transforms listed in Table 2.4. Those

transforms have been tested by Murli and Rizzardi's algorithm (Algorithm 682 in TOMS), which is a FORTRAN implementation of Talbot's method [11].

In the numerical tests, we translated Talbot's method into MATLAB codes, which we refer to be as the CTB-ALG (the classical Talbot algorithm). Also, we refer to our modified Talbot method, associated with Algorithm 2, as the NPS-ALG (an improved Talbot algorithm with numerical parameter selection). In the following tests, we apply the same required accuracy $tol = 10^{-6}$, the same machine precision $\varepsilon_m = 2^{-52}$ (IEEE standard double precision), and the same number of function evaluations N obtained by Talbot's parameter selection strategy, to both algorithms. The numerical results are shown in Table 3.1 to 3.6.

In those tables, we display two sets of parameters (λ, σ, ν, N): the first, as computed by Algorithm 2, and the second, as computed by Talbot's original parameter selection. The abbreviation 'R.Error' represents the relative errors computed by each algorithm.

One can see that our algorithm, NPS-ALG, does improve the accuracy on average by about a factor of 10^{-3} (for functions with only real singularities) to 10^{-2} (for functions with complex singularities). However, it is certainly true that the algorithm NPS-ALG performs more work than CTB-ALG does. Therefore, we also tested the running time (CPU-time) of both algorithms on computing the inversion of Transforms 2, 3, and 5 (in Table 2.4) for 1000 values of t, from 1 to 1000 (for Transforms 2 and 3) or from 0.1 to 100 (for Transform 5). The average running times for each computation are about 2.2×10^{-3} seconds for CTB-ALG algorithm and about 1.8×10^{-1} seconds for algorithm NPS-ALG, while the average relative errors are 4.0×10^{-6} and 2.3×10^{-8} respectively. This shows that the algorithm NPS-ALG improves the accuracy on average by about 6×10^{-3} , but it also increases the computing cost by about 80 times. These tests were implemented on a Pentium III 500.

We also applied the new algorithm to the practical problems listed in Table 2.12, Chapter 2. For a description of those problems, we refer to Subsection 2.5.3. In this test,

we prefer to implement our algorithm NPS-ALG with two different required tolerances $tol = 10^{-6}$ and $tol = 10^{-12}$ (consistent with the tests in Chapter 2), rather than comparing with algorithm CTB-ALG, since, as we have seen, if Talbot's method is applicable to the problem, then algorithm NPS-ALG give better results. The numerical results are shown in Table 3.7 to 3.10.

We should mention that the Laplace transform of the electrical circuit problem is not in class $\mathcal{B}$ since it does not satisfy property (3.a). Talbot's method therefore is not applicable to this problem, so no results can be shown. For the other problems, our algorithm NPS-ALG works well, as the tables clearly demonstrate.

We also point out that in Duffy's test [2] Talbot's method performed poorly for the beam problem (Transform 5 in Table 2.12). However, our numerical results show that Talbot's method does work efficiently for this problem (see Table 3.10), provided one avoids crossing the branch cuts; recall Example 3..2 in Section 3.2. We conjecture that in this paper [2], the branch-cuts were not properly treated.

Considering all the numerical tests, we have found that using our algorithm (NPS-ALG) to select the geometric parameters improves the accuracy by two or three significant decimal digits. This means that our algorithm can achieve the desired accuracy (tol) with smaller values of N than the N generated by Talbot's strategy of parameter selection. Therefore, we have also tested how small an N can be taken to reach the desired accuracy for the Transform 4 in Table 2.4 and for the problem of shock waves in diatomic chains (the fourth transform in Table 2.12). The results are shown in Table 3.11 and 3.12. One can see that, to reach the desired accuracy, our algorithm NPS-ALG used smaller values of N than did the classical Talbot's algorithm. This saves, on average, about 27% of the number of transform evaluations.

Finally, in any future work, attempting to improve our numerical algorithm using smaller N, will be an interest challenge.

Table 3.1: $F(s) = 1/s^2$ and $f(t) = t$

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	$(\lambda, \sigma, \nu, N)$	R.Error	$(\lambda, \sigma, \nu, N)$	R.Error
0.1	(6.76e+1, -6.76e-3, 1.00, 13)	1.2e-14	(6.80e+1, 0, 1, 13)	2.8e-11
1	(6.76e+0, -1.08e-2, 1.00, 13)	9.8e-14	(6.80e+0, 0, 1, 13)	2.8e-11
10	(6.76e-1, 1.00e-4, 1.00, 13)	4.1e-15	(6.80e-1, 0, 1, 13)	2.8e-11
100	(6.76e-2, -1.28e-2, 1.01, 13)	5.1e-14	(6.80e-2, 0, 1, 13)	2.8e-11
1000	(6.76e-3, -1.61e-3, 1.01, 13)	1.2e-12	(6.80e-3, 0, 1, 13)	2.8e-11

Table 3.2: $F(s) = \log(s)/s$ and $f(t) = -\gamma - \log(t)$

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	$(\lambda, \sigma, \nu, N)$	R.Error	$(\lambda, \sigma, \nu, N)$	R.Error
0.1	(6.71e+1, 1.72e+0, 1.00, 12)	9.2e-09	(6.80e+1, 0, 1, 12)	5.9e-08
1	(6.71e+0, -1.33e+0, 9.94e-1, 12)	8.3e-12	(6.80e+0, 0, 1, 12)	1.4e-07
10	(6.63e-1, -2.02e-1, 9.94e-1, 12)	5.7e-11	(6.80e-1, 0, 1, 12)	2.1e-08
100	(6.96e-2, -9.74e-3, 9.94e-1, 12)	3.4e-13	(6.80e-2, 0, 1, 12)	8.1e-09
1000	(6.82e-3, -1.61e-3, 1.01e+0, 12)	1.3e-12	(6.80e-3, 0, 1, 12)	2.9e-09

Table 3.3: $F(s) = \exp(-4\sqrt{s})$ and $f(t) = 2\exp(-4/t)/(t\sqrt{\pi t})$

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	(λ, σ, ν, N)	R.Error	(λ, σ, ν, N)	R.Error
0.1	underflow		underflow	
1	(7.50e+0, 4.79e-2, 1.00, 12)	6.0e-12	(6.80e+0, 0, 1, 12)	1.81e-09
10	(6.82e-1, 5.35e-2, 1.00, 12)	1.7e-10	(6.80e-1, 0, 1, 12)	2.22e-09
100	(6.79e-2, -2.87e-3, 9.90e-1, 12)	9.9e-11	(6.80e-2, 0, 1, 12)	1.28e-07
1000	(6.63e-3, -3.24e-3, 9.87e-1, 12)	1.2e-09	(6.80e-3, 0, 1, 12)	1.93e-05

Table 3.4: $F(s) = \arctan(1/s)$ and $f(t) = \sin(t)/t$

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	(λ, σ, ν, N)	R.Error	(λ, σ, ν, N)	R.Error
0.1	(6.76e+1, -1.33e+1, 1.00, 12)	8.1e-13	(6.85e+1, 0.00, 1.00, 12)	1.9e-8
1	(7.37e+0, -2.99e+0, 1.03, 12)	4.9e-09	(7.30e+0, 0.00, 1.00, 12)	2.9e-8
10	(4.84e-1, 6.37e-1, 1.64, 17)	5.4e-08	(4.97e-1, 6.37e-1, 1.65, 17)	8.7e-7
100	(3.50e-2, 8.04e-2, 19.0, 50)	1.1e-07	(3.50e-2, 7.83e-2, 19.0, 50)	3.7e-7
1000	(1.09e-3, 1.04e-2, 500, 320)	3.0e-06	(1.09e-3, 1.02e-2, 500, 320)	4.5e-6

Table 3.5: $F(s) = \log\left((s^2+1)/(s^2+4)\right)$ and $f(t) = 2[\cos(2t) - \cos(t)]/t$.

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	$(\lambda, \sigma, \nu, N)$	R.Error	$(\lambda, \sigma, \nu, N)$	R.Error
0.1	(6.99e+1, 6.99e-3, 1.01, 12)	1.2e-12	(6.90e+1, 0.00, 1.00, 12)	6.5e-10
1	(7.93e+0, -1.62e+0, 1.01, 13)	1.0e-12	(7.80e+0, 0.00, 1.00, 13)	4.9e-10
10	(4.71e-1, 6.59e-1, 3.28, 21)	6.9e-11	(4.74e-1, 6.59e-1, 3.29, 21)	4.9e-08
100	(1.96e-2, 9.23e-2, 60.1, 86)	2.8e-06	(1.97e-2, 9.37e-2, 60.1, 86)	2.6e-06
1000	(1.09e-3, 1.03e-2, 9997, 1611)	1.6e-08	(1.09e-3, 1.02e-2, 9998, 1611)	2.2e-07

Table 3.6: $F(s) = s^2/(s^3+8)$ and $f(t) = [e^{-2t} + 2e^t\cos(\sqrt{3}t)]/3$.

$tol = 10^{-6}$	NPS-ALG		CTB-ALG	
t	$(\lambda, \sigma, \nu, N)$	R.Error	$(\lambda, \sigma, \nu, N)$	R.Error
0.1	(6.72e+1, -25.1, 1.01, 12)	8.8e-10	(6.89e+1, 1.00, 1.00, 12)	2.9e-08
1	(7.69e+0, 1.02, 1.00, 13)	8.8e-12	(7.67e+0, 1.00, 1.00, 13)	2.5e-08
10	(4.86e-1, 1.56, 2.85, 20)	1.2e-09	(4.79e-1, 1.65, 2.84, 20)	1.2e-06
100	(2.39e-2, 1.12, 43.9, 85)	4.2e-10	(2.41e-2, 1.09, 43.9, 85)	9.1e-07
1000	overflow		overflow	

Table 3.7: Viscoplastic Rod Problem

NPS-ALG	$tol = 10^{-6}$		$tol = 10^{-12}$	
t	$(\lambda, \sigma, \nu, N)$	R.Error	$(\lambda, \sigma, \nu, N)$	R.Error
0.01	(7.50e+2 1.21, 1.00, 12)	1.6e-12	(6.97e+2, -8.93, 1.00, 21)	1.6e-12
0.1	(6.45e+1, 3.30, 1.00, 12)	2.2e-10	(6.80e+1 6.80e-3, 1.00, 21)	1.0e-15
1	(6.45e+0, -2.64, 1.01, 12)	4.5e-10	(6.80, 6.8e-4, 1.00, 21)	7.5e-16
10	(6.79e-1 -6.40e-3 , 1.00, 12)	2.1e-09	(6.8e-1, 1.0e-4, 1.00, 21)	5.1e-12

Table 3.8: Viscous Fluid Mechanics Problem

NPS-ALG	$tol = 10^{-6}$		$tol = 10^{-12}$	
t	(λ, σ, ν, N)	R.Error	(λ, σ, ν, N)	R.Error
0.25	(2.69e+1, -2.75, 1, 12)	1.5e-11	(2.72e+1, 2.72e-3, 1, 21)	7.1e-16
1	(6.76e+0 -7.01e-1 1, 12)	3.7e-12	(6.80e+0, -6.80e-4, 1, 21)	1.2e-14

Table 3.9: Shock Waves in Diatomic Chains Problem

NPS-ALG	$tol = 10^{-6}$		$tol = 10^{-12}$	
t	(λ, σ, ν, N)	R.Error	(λ, σ, ν, N)	R.Error
1	(9.64, 4.97e-1, 0.988, 13)	7.8e-12	(8.80, 0.00, 1.00, 21)	1.3e-15
2	(6.51, -8.33e-2, 1.00, 15)	3.5e-11	(5.40, -5.40e-4, 1.03, 22)	2.8e-14
4	(1.21, 1.63, 2.63, 20)	1.6e-12	(1.21, 1.63, 2.63, 34)	1.2e-14
8	(5.64e-1, 8.51e-1, 5.24, 26)	6.7e-12	(5.65e-1, 8.52e-1, 5.31, 41)	4.3e-13

Table 3.10: Timoshenko Beam Problem

NPS-ALG	$tol = 10^{-6}$		$tol = 10^{-12}$	
t	(λ, σ, ν, N)	R.Error	(λ, σ, ν, N)	R.Error
2	(3.60, 1.00, 1.01, 12)	6.6e-13	(3.69, 1.00, 1.00, 21)	7.6e-14
4	(2.04, 9.99e-1, 9.98e-1, 13)	1.3e-12	(1.99, 1.00, 1.00, 21)	7.7e-14
6	(1.46, 8.98e-1, 1.00, 13)	6.6e-12	(1.42, 1.00, 1.00, 21)	6.8e-14
8	(1.13, 1.00, 1.00, 14)	1.5e-11	(1.14, 1.00, 1.00, 21)	1.6e-13

Table 3.11: Test of the smallest N to reach the desired accuracy for the transform $\arctan(1/s)$

	$tol = 10^{-6}$			$tol = 10^{-12}$	
t	CTB-ALG	NPS-ALG	t	CTB-ALG	NPS-ALG
1	11	7	80	42	30
10	16	9	100	50	40
20	22	13	200	88	70
40	28	20	400	162	124

Table 3.12: Test of the smallest N to reach the desired accuracy for the Diatomic Chains Problem

	$tol = 10^{-6}$		$tol = 10^{-12}$	
t	CTB-ALG	NPS-ALG	CTB-ALG	NPS-ALG
1	9	6	15	9
2	11	8	23	13
4	12	8	22	15
8	18	13	32	22

4. A NEW METHOD BASED ON THE DE-TRANSFORMATION

We have considered two of the most efficient numerical methods for inverting the Laplace transform, namely, the direct integration method and Talbot's method. In this chapter, we consider a third method, also based on the trapezoidal-type rules, which we believe to be new. It is an application of the double exponential formula for Fourier-type integrals [13, 14], which we have modified for the inverse Laplace transform problem.

This chapter will be organized as follows. In the first section, we shall introduce the technique of the double exponential transformation for numerical integration. In the next, a couple of double exponential formulas for Fourier-type integrals are presented. We extend this method to the numerical inversion of the Laplace transform in the third section, and finally, numerical tests will be given in the concluding section.

4.1. Introduction to DE-Formulas

It is well-known that the double exponential formula (abbreviated, DE-formula) is one of the most efficient methods for the numerical integration of analytic functions. It was first proposed by Takahasi and Mori in 1974 [32]. The basic idea of the DE-formula is based on the double exponential transformation, which we introduce as follows.

First, recall from Example 1..5, the high degree of accuracy achieved by the trapezoidal rule when applied to an integral such as

$$\int_{-\infty}^{+\infty} \mathrm{e}^{-x^2}\, dx. \tag{4.1}$$

The reason for this high accuracy is due to the rapid decay of the integrand, $\exp(-x^2)$, as $x \longrightarrow \pm\infty$; and the high accuracy then follows from the Euler-Maclaurin formula (see Theorem 1..7 in Section 1.4). Such a fast decay rate is called *super-exponential decay.* The idea of a DE-formula is to apply a change of variables to a given integral to transform it into an integral similar to (4.1), which is then approximated by the trapezoidal-type rules. In fact, the DE-formula–as the name suggests–aims for an even more rapid decay

than obtained in (4.1), namely, one of the form

$$\exp(-c\exp(|x|)) \longrightarrow 0 \quad \text{as } x \longrightarrow \pm\infty, \quad c > 0. \tag{4.2}$$

This is referred to as *double exponential decay.* We shall see several double exponential functions, which can be used as DE-transformation, in what follows.

To introduce the method formally, we consider the integral:

$$I = \int_{\alpha}^{\beta} g(x)\, dx, \tag{4.3}$$

in which we suppose that $g(x)$ is analytic in an open neighborhood of of the interval (α, β), where α and β may be real numbers, or $\pm\infty$. We also allow $g(x)$ to be singular at α or β or both, provided I is integrable. A change of variables can be made by using an increasing function ϕ such that

$$x = \phi(u), \quad \phi(-\infty) = \alpha, \quad \phi(\infty) = \beta. \tag{4.4}$$

Thus, the integral (4.3) becomes

$$I = \int_{-\infty}^{\infty} g(\phi(u))\phi'(u)\, du. \tag{4.5}$$

Next, by applying the trapezoidal rule with step size h to (4.5), we obtain

$$I_h = h \sum_{n=-\infty}^{+\infty} g(\phi(nh))\phi'(nh), \tag{4.6}$$

which serves as a quadrature formula provided a suitable function $\phi(u)$ can be found.

If we select a function $\phi(u)$ with the property that $\phi'(u)$ approaches zero double exponentially fast as u tends to $\pm\infty$, then $g(\phi(nh))\phi'(nh)$ decays double exponentially fast when n goes to $\pm\infty$, which means that the summation can be truncated at a small number of terms [33].

In practical implementations, the formula (4.6) must be truncated below at a certain N_- for negative n and at $(N_+ - 1)$ above for positive n, so that the actual computation is

$$I_h^{(N)} = h \sum_{n=-N_-}^{N_+-1} g(\phi(nh))\phi'(nh), \quad N = N_- + N_+. \tag{4.7}$$

This is called the *DE-formula*, and the variable transformation $x = \phi(u)$ is called the *DE-transformation* [33, 32].

4.2. DE-Formulas for Fourier-Type Integrals

Many types of DE-transformations are used to approximate various kinds of integrals. For example, the transformation

$$\phi(u) = \tanh\left(\frac{\pi}{2}\sinh u\right), \quad u \in (-\infty, \infty) \tag{4.8}$$

gives a good DE-formula for integrals over the interval $[-1, 1]$. Likewise, the transformation

$$\phi(u) = \sinh\left(\frac{\pi}{2}\sinh u\right), \quad u \in (-\infty, \infty) \tag{4.9}$$

gives an efficient DE-formula for integrals over the interval $(-\infty, \infty)$ [33].

As for the integral

$$I = \int_0^\infty g(x)\,dx, \tag{4.10}$$

if $g(x)$ is a slowly decaying function as $x \longrightarrow \infty$, then the transformation

$$\phi(u) = \exp\left(2\sinh u\right), \quad u \in (-\infty, \infty) \tag{4.11}$$

gives an effective DE-formula [33]. However, the DE-formula obtained by (4.11) fails for the class of integrals of slowly decaying oscillatory functions defined on $[0, \infty)$, such as

$$I_s = \int_0^\infty g_1(x)\sin(\omega x)\,dx \tag{4.12}$$

or

$$I_c = \int_0^\infty g_1(x)\cos(\omega x)\,dx, \tag{4.13}$$

where $g_1(x)$ is a slowly decaying algebraic function. The integrals (4.12) and (4.13) are of *Fourier-type*.

An efficient DE-formula for Fourier-type integrals was proposed by T. Ooura and M. Mori in 1991 [13]. Their transformation is

$$x = M\phi(u)/\omega, \qquad \phi(u) = \frac{u}{1 - \exp(-K \sinh u)}, \tag{4.14}$$

where M and K are positive constants. This transformation has the properties that

$$\phi(u) \longrightarrow \begin{cases} 0, & \text{as} \quad u \longrightarrow -\infty, \\ 1/K, & \text{as} \quad u \longrightarrow 0, \\ u, & \text{as} \quad u \longrightarrow \infty, \end{cases} \tag{4.15}$$

double exponentially, and that

$$\phi'(u) \longrightarrow 0, \quad \text{as } u \longrightarrow -\infty, \tag{4.16}$$

double exponentially. It therefore satisfies the requirements that

$$\phi(-\infty) = 0, \quad \phi(\infty) = \infty, \quad \text{and} \quad \phi'(-\infty) = 0. \tag{4.17}$$

Accordingly, the sine-type integral (4.12) becomes

$$I_s = \frac{M}{\omega} \int_{-\infty}^{\infty} g_1(M\phi(u)/\omega) \sin(M\phi(u))\phi'(u)\, du. \tag{4.18}$$

Applying the trapezoidal rule to this integral, we obtain

$$I_{s,h} = \frac{Mh}{\omega} \sum_{n=-\infty}^{\infty} g_1\left(M\phi(nh)/\omega\right) \sin(M\phi(nh))\phi'(nh). \tag{4.19}$$

In the above formula, we need to select the constant M and the step size h such that

$$Mh = \pi. \tag{4.20}$$

Subsequently, by (4.15), we have

$$\sin(M\phi(nh)) \sim \sin(Mnh) = \sin(n\pi) = 0, \quad \text{as} \quad n \longrightarrow \infty, \tag{4.21}$$

double exponentially. Also, by (4.16), we obtain

$$\phi'(nh) \longrightarrow 0, \quad \text{as } n \longrightarrow -\infty, \tag{4.22}$$

double exponentially. Therefore, (4.21) and (4.22) imply that the summand in formula $I_{s,h}$ approaches zero double exponentially as n tends to $\pm\infty$. As a result, one can find moderate positive integers N_+ and N_- so that the truncated formula:

$$I_{s,h}^{(N)} = \frac{\pi}{\omega} \sum_{n=-N_-}^{N_+-1} g_1\left(M\phi(nh)/\omega\right) \sin(M\phi(nh))\phi'(nh), \quad N = N_+ + N_-, \tag{4.23}$$

is a good approximation to the infinite summation (4.19). In practical implementations, $K = 6$ and $M = \pi/h$ was recommended by Ooura and Mori [13]; the step size h is supplied by the user.

As for the cosine-type integral (4.13), we prefer to apply the **midpoint rule** rather than the trapezoidal rule that was used by Ooura and Mori [13, 14]. It has the advantage that the evaluation points have a translation of $\frac{1}{2}h$ and, therefore, the cosine factor goes to zero, similar to the sine factor in (4.21). Hence, the DE-formula for the integral I_c can be expressed in the form:

$$I_{c,h} = \frac{\pi}{\omega} \sum_{n=-\infty}^{\infty} g_1\left(M\phi\left((n+1/2)\,h\right)/\omega\right) \cos\left(M\phi\left(n+1/2\right)h\right)\phi'\left((n+1/2)\,h\right). \tag{4.24}$$

In practice, the formula (4.24) is truncated, the same as in (4.23). Notice that, throughout this chapter, the total number of function evaluations, N, is set as $N = N_+ + N_-$.

An improved version of the DE-transformation for Fourier-type integrals, designed for the integrals whose integrand have singularities in the finite complex-plane, was presented by the same authors in 1999 [14]. This transformation is

$$x = M\phi(u)/\omega, \qquad \phi(u) = \frac{u}{1 - \exp\left(-2u - \alpha\left(1 - e^{-u}\right) - \beta\left(e^{u} - 1\right)\right)}, \tag{4.25}$$

where M, α, and β are positive constants. M is chosen by (4.20), and α and β are chosen to satisfy

$$\alpha = o((M\log(M))^{-1/2}), \quad \beta = O(1), \quad 0 < \alpha \leq \beta \leq 1. \tag{4.26}$$

In actual implementations, the choices

$$\alpha = \frac{\beta}{\sqrt{1 + M\log(1+M)/(4\pi)}}, \quad \beta = \frac{1}{4}, \tag{4.27}$$

typically give accurate results [14].

We now give some examples to demonstrate how efficient these DE-formulas are and to compare them for a few Fourier-type integrals. We refer to the two DE-formulas obtained by (4.14) and (4.25) as DE-formula(I) and DE-formula(II), respectively.

Example 4..1. *Consider the integral*

$$\int_0^\infty \frac{\cos(x)}{1+x^2}\,dx, \tag{4.28}$$

which has the exact sum $\pi/(2\mathrm{e})$. *Its integrand is a slowly decaying cosine-type function with singularities at* $\pm i$ *in the complex-plane.*

To approximate this integral, we apply the DE-formula(I) and DE-formula(II), as well as the standard trapezoidal rule without the DE-transformation. We have plotted the relative errors with respect to the step size h in Figure 4.1–4.2. The curves in Figure 4.1 are the comparison of the relative errors, computed by the trapezoidal rule and the DE-formula(I), with $N = 100$ ($N_+ = N_- = 50$). One can see that the optimal accuracy of the trapezoidal rule is only about a factor of 10^{-3}, while the DE-formula(I) can attain an accuracy up to a factor of 10^{-14}, provided one supplies a good choice of step size h (So far, there is no complete theory for selecting the optimal h). In Figure 4.2, we observe that, by comparing their minimum errors, both DE-formula(I) and DE-formula(II) work efficiently for this integral when $N = 64$ ($N_+ = N_- = 32$). One can see that the DE-formula(II) improves the optimal accuracy by about a factor of 10^{-2}, compared to the DE-formula(I).

Example 4..2. *We next consider the sine-type integral*

$$\int_0^\infty \frac{\sin(x)}{x}\,dx, \tag{4.29}$$

which has the exact value $\pi/2$.

Just as in Example 4.28, we have applied the trapezoidal rule and the two DE-formulas to this problem and plotted the relative errors in Figure 4.3. The performance of the trapezoidal rule is rather poor, therefore, it was not displayed, while both the DE-formulas work well with $N = 50$ ($N_+ = N_- = 25$). The optimal accuracy is about a

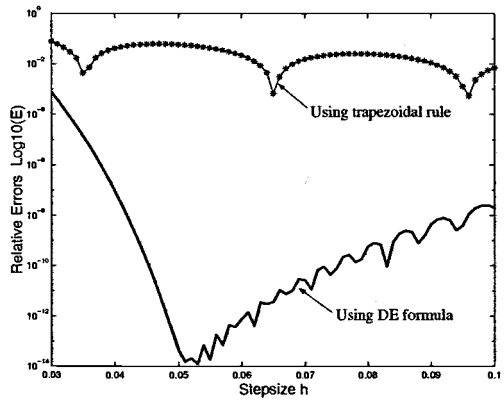

Figure 4.1: The relative error curves of approximating (4.28) by the trapezoidal rule and DE-formula(I) with $N = 100$.

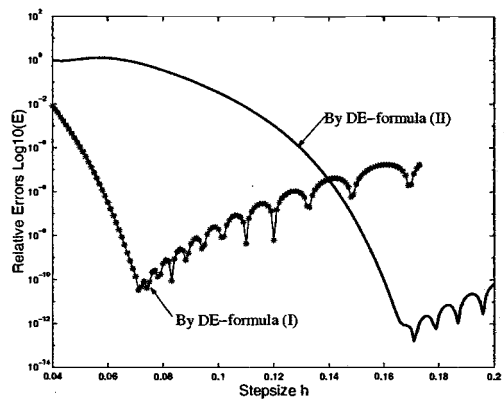

Figure 4.2: The same as Figure 4.1 except by DE-formulas (I) and (II) with $N = 64$.

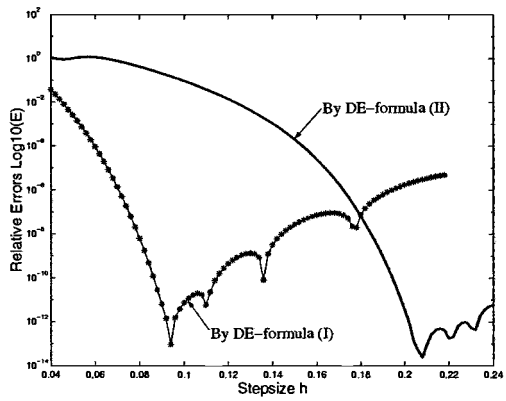

Figure 4.3: The relative error curves of approximating (4.29) by DE-formulas (I) and (II) with $N = 50$.

factor of 10^{-13} or 10^{-14} by the DE-formula(I) or DE-formula(II), respectively. Again, the DE-formula(II) performs slightly better than the DE-formula(I).

From these pictures, we point out that the minimum errors computed by the two DE-formulas occur at different step sizes. Also, the accuracy of the DE-formulas depends significantly on the number of function evaluations N and a suitable selection of the step size h (which also determines the constant M). It appears that, for a given N, there is an optimal step size h that minimizes the total error. Finding the optimal h and N seems to be an open problem, and it will not be addressed here.

We have seen the power of the DE-formulas for Fourier-type integrals. Now, we shall apply these formulas to the inverse problem of the Laplace transform. We refer to this method as the DET method (double exponential transformation method).

4.3. A DET Method for Inverting the Laplace Transform

Recall that the original function $f(t)$ of a given Laplace transform $F(s)$ can be reconstructed using the Bromwich integral (1.1):

$$f(t) = \frac{1}{2\pi i}\int_{a-i\infty}^{a+i\infty} \mathrm{e}^{st}F(s)\,ds = \frac{\mathrm{e}^{at}}{2\pi}\int_{-\infty}^{\infty} e^{ity}F\left(a+iy\right)\,dy, \quad t>0.$$

This integral can be equivalently represented by the following formulas [6]:

$$f(t) = \frac{2\mathrm{e}^{at}}{\pi}\int_0^{\infty} \mathrm{Re}\{F(a+iy)\}\cos(ty)\,dy, \tag{4.30}$$

$$f(t) = \frac{-2\mathrm{e}^{at}}{\pi}\int_0^{\infty} \mathrm{Im}\{F(a+iy)\}\sin(ty)\,dy, \tag{4.31}$$

or their average

$$f(t) = \frac{e^{at}}{\pi}\int_0^{\infty} [\mathrm{Re}\{F(a+iy)\}\cos(ty) - \mathrm{Im}\{F(a+iy)\}\sin(ty)]\,dy. \tag{4.32}$$

Here $s = a+iy$, $a, y \in \mathbb{R}$ and $a > \sigma_0$ (the Laplace convergence abscissa). Since these integrals are of Fourier-type, we can apply the DE-transformations (4.14) or (4.25). For

example, by making the variable transformation $y = M\phi(u)/t$ in inversion formula (4.31), we have

$$f(t) = -\frac{2Me^{at}}{\pi t}\int_{-\infty}^{\infty} \mathrm{Im}\{F(a+iM\phi(u)/t)\}\sin(M\phi(u))\phi'(u)\,du. \tag{4.33}$$

Next, by applying the trapezoidal rule to this integral, we obtain

$$\widetilde{f}(t) = -\frac{2e^{at}}{t}\sum_{n=-\infty}^{\infty} \mathrm{Im}\{F(a+iM\phi(nh)/t)\}\sin(M\phi(nh))\phi'(nh). \tag{4.34}$$

If we truncate the approximate formula (4.34) at a certain term N_- for negative n and $(N_+ - 1)$ for positive n, we get the approximation:

$$\widehat{f}(t) = -\frac{2e^{at}}{t}\sum_{n=-N_-}^{N_+-1} \mathrm{Im}\{F(a+iM\phi(nh)/t)\}\sin(M\phi(nh))\phi'(nh). \tag{4.35}$$

Similarly, if we apply the DE-formula (4.24) to the Laplace inversion formula (4.30) and truncate it at $(N_+ - 1)$ and N_-, for positive n and negative n, respectively, then we obtain

$$\begin{aligned}\widehat{f}(t) \;=\; \frac{2e^{at}}{t}\sum_{n=-N_-}^{N_+-1} &\mathrm{Re}\{F(a+iM\phi((n+1/2)h)/t)\} \\ &\times\cos(M\phi((n+1/2)h))\phi'((n+1/2)h).\end{aligned} \tag{4.36}$$

Both (4.35) and (4.36) are similar to the direct integration method, except the variable has been transformed. Therefore, this method is applicable to the same class of Laplace transforms as the direct integration method; that is, the family of transforms which are in class $\mathcal{A}$ (defined in Section 2.1), and which can be translated into class $\mathcal{A}$.

Since there are three equivalent formulas (4.30)–(4.32) for the inverse Laplace transform and two DE-transformations (4.14) and (4.25), we actually have six numerical formulas for inverting the Laplace transform by the DET method. The two formulas obtained by applying the DE-formulas to (4.32) are just the average of (4.30) and (4.31). Accordingly, they do not improve the accuracy with the same number of points, N, but increase the work. We shall therefore test the four formulas only by applying the DE-transformations, (4.14) and (4.25), to the two inversion formulas of the Laplace transform, (4.30) and (4.31),

in the next section. We refer to these formulas as DET(Is) [applying DE-formula(I) to sine formula (4.31)], DET(Ic) [applying DE-formula(I) to cosine formula (4.30)], DET(IIs) [applying DE-formula(II) to sine formula (4.31)], and DET(IIc) [applying DE-formula(II) to cosine formula (4.30)].

This method involves three parameters: the number of function evaluations N, the step size h, and the parameter a. Just as in the direct integration method, the parameter a determines the location of the Bromwich line. It therefore must be larger than the rightmost singularity of the given Laplace transform. Also, just like in Chapter 2, no theory presently exists for selecting the optimal value of a, or for the optimal value of h. In the next section, we shall empirically determine the optimal parameters a and h by plotting the level curves of the error in the (at, h) parameter plane.

4.4. Numerical Tests

In this section, we shall first test the four DET-formulas to see which formula performs best for the inversion of the Laplace transform. After that, we shall show tests of the sample transforms listed in Table 2.4, and the practical problems listed in Table 2.12 (the same as the tests in Chapter 2). We mention here that, for convenience, we shall choose $N_+ = N_-$ in all the test problems. We first consider the following example:

Example 4..3. *The Laplace transform*

$$F(s) = \frac{\exp(-1/s)}{\sqrt{s}} \tag{4.37}$$

has an essential singularity at 0 and is associated with the inversion of

$$f(t) = \frac{\cos(2\sqrt{t})}{\sqrt{\pi t}}, \quad t > 0.$$

In this example, we first chose $N = 64$ and applied the four formulas, DET(Is), DET(Ic), DET(IIs), and DET(IIc), to approximate the inversion of the transform (4.37).

This was done for a large number of a and h values. We then plotted the level curves of the relative error as a function of the parameters at and h. The results are shown in Figures 4.4–4.7. The labels of the level curves indicate the base 10 logarithm of the computed errors. For instance, -8 denotes that the magnitude of the relative error is 10^{-8} on that curve, which means roughly 8 significant digits of precision.

These figures show that this method can reach high accuracy, provided that the choices of the parameter at and the step size h are carefully made, and, in addition, N is sufficiently large. One can also see that the performance of the DE-formula for sine-type integrals is slightly better than that of the DE-formula for cosine-type integrals. Moreover, we have learned from Example 4.2 that the DE-transformation (II) performs slightly better than the DE-formula (I). Therefore, we shall use the DE-formula(IIs) as our main formula to test the sample transforms and the practical problems.

In Figures 4.6–4.7 we see that this method is also efficient for a large range of values for t (t from 1 to 1000), provided N is sufficiently large. (In those figures, we have used $N = 64$, 100, and 128.) In Figures 4.8–4.13, we show the test results for the sample transforms, listed in Table 2.4. The level curves indicate accurate results for a large range of values t (=1, 10, and 100) with the same N (=64). Notice that, in Figures 4.11–4.13 (right), the inside-most level curve is not as small as that in the right-most pictures, since N is not sufficiently large. In this case (for the transforms with complex singularities), N should increase as t increases. Generally speaking, all the results indicate that this method can achieve high accuracy, provided the parameters a and h are chosen inside the lowest level curve, as well as N sufficiently large.

Next, we tested the practical problems listed in Table 2.12; those problems have been described in Section 2.5, Chapter 2. The results are shown in Figures 4.14–4.18. Again, one can see that this method is accurate and applicable to the five practical problems, provided N is sufficiently large. We also point out in Figure 4.18 that the selection of parameter at must be larger than 1, since the beam problem has a singularity at $s = 1$,

as we have mentioned in the preceding section.

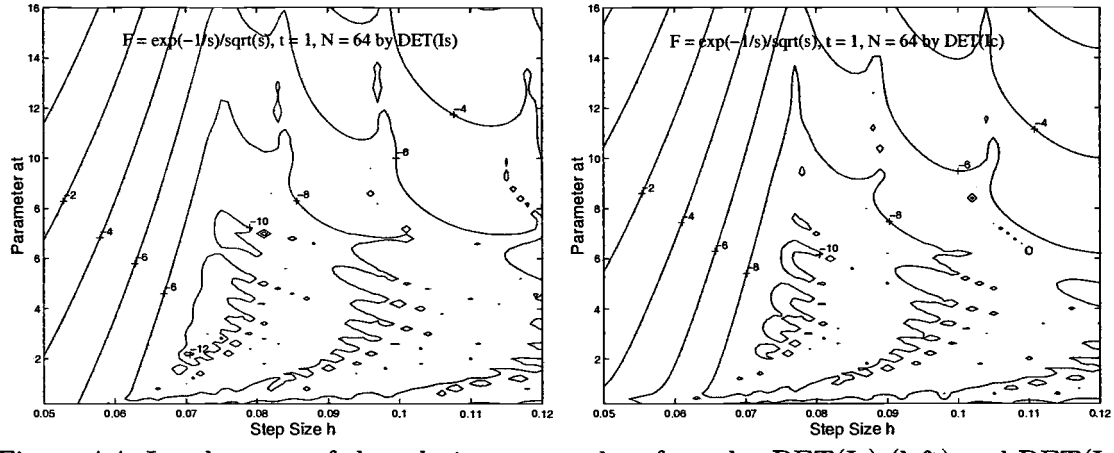

Figure 4.4: Level curves of the relative error, when formulas DET(Is) (left) and DET(Ic) (right) are applied to the transform (4.37). Here $t = 1$, $N = 64$.

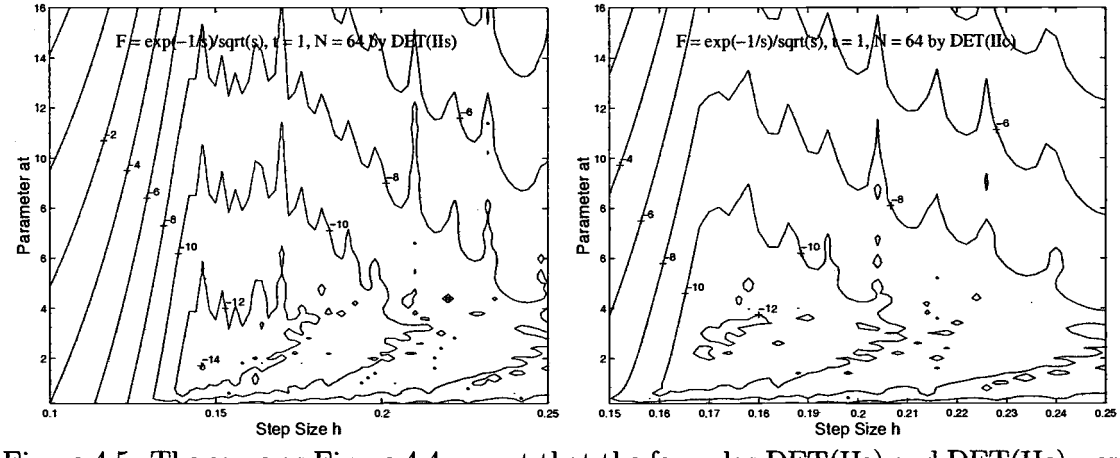

Figure 4.5: The same as Figure 4.4 except that the formulas DET(IIs) and DET(IIc) were used.

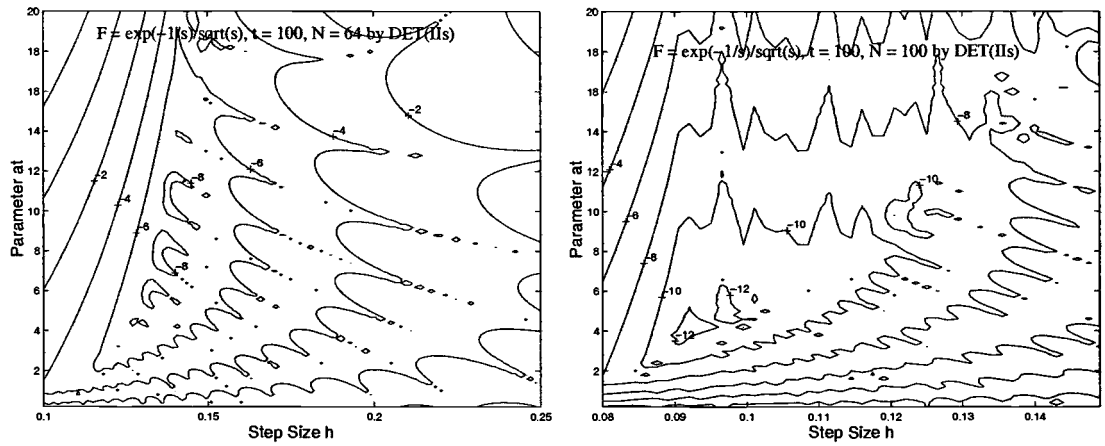

Figure 4.6: The same as Figure 4.4 except that the formula DET(IIs) was used and they were computed at $t = 100$ with $N = 64$ (left) and 100 (right).

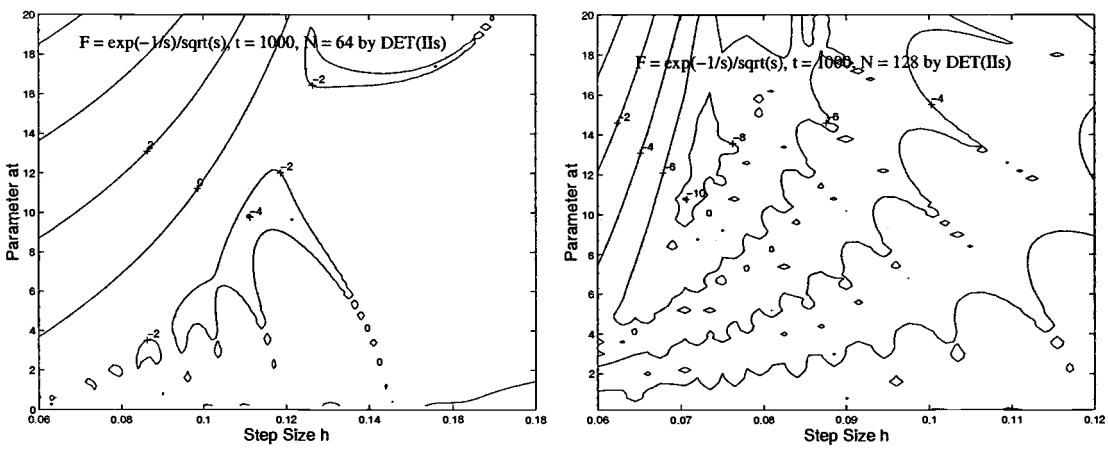

Figure 4.7: The same as Figure 4.6 except that they were computed at $t = 1000$ with $N = 64$ (left) and 128 (right).

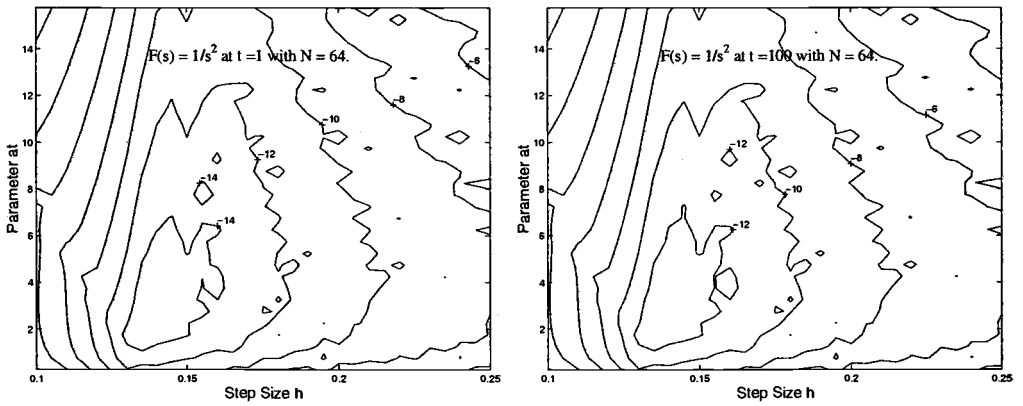

Figure 4.8: Level curves of the relative error, when formula DET(IIs) is applied to $F(s) = 1/s^2$, at $t = 1$ (left) and 100 (right), and with $N = 64$.

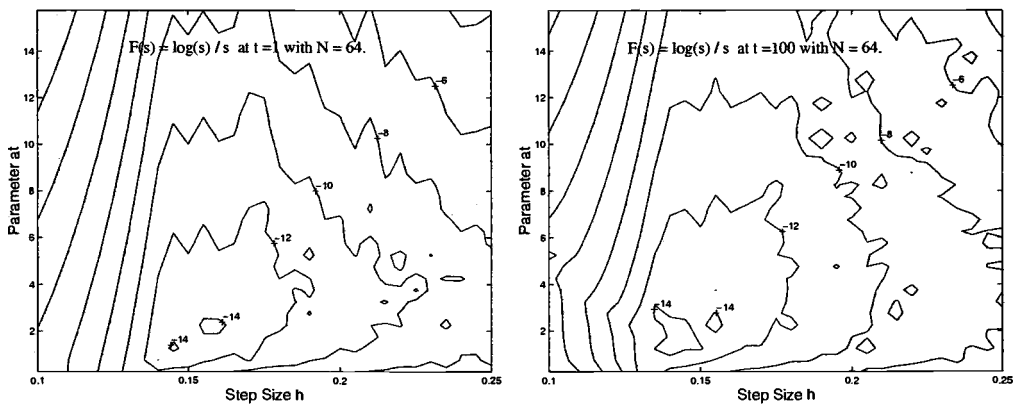

Figure 4.9: The same as Figure 4.8 except for $F(s) = \log(s)/s$.

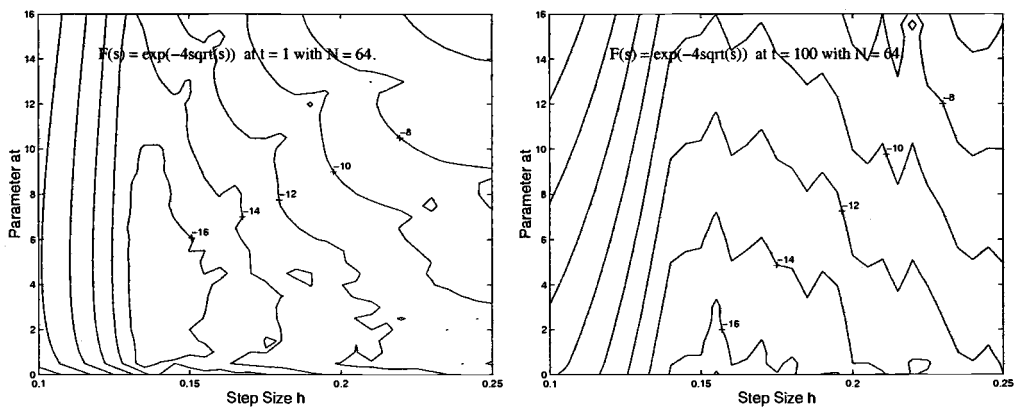

Figure 4.10: The same as Figure 4.8 except for $F(s) = \exp(-4\sqrt{s})$.

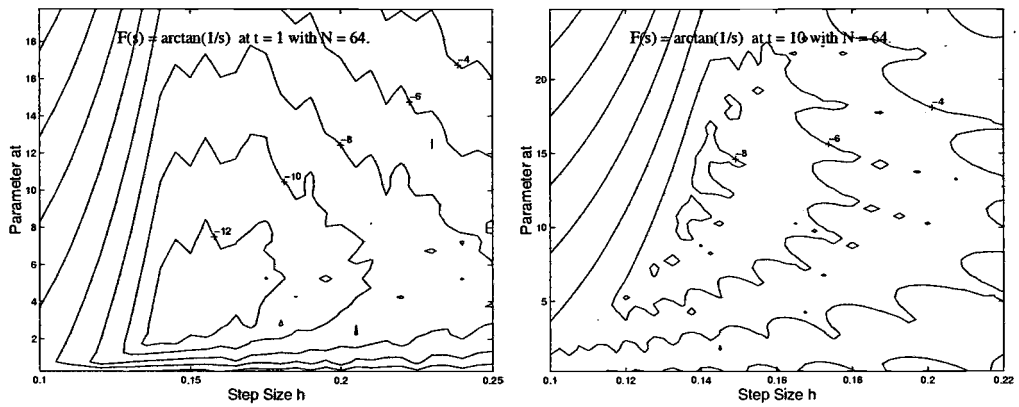

Figure 4.11: The same as Figure 4.8 except for $F(s) = \arctan(1/s)$ and $t = 1$ (left) and $t = 10$ (right).

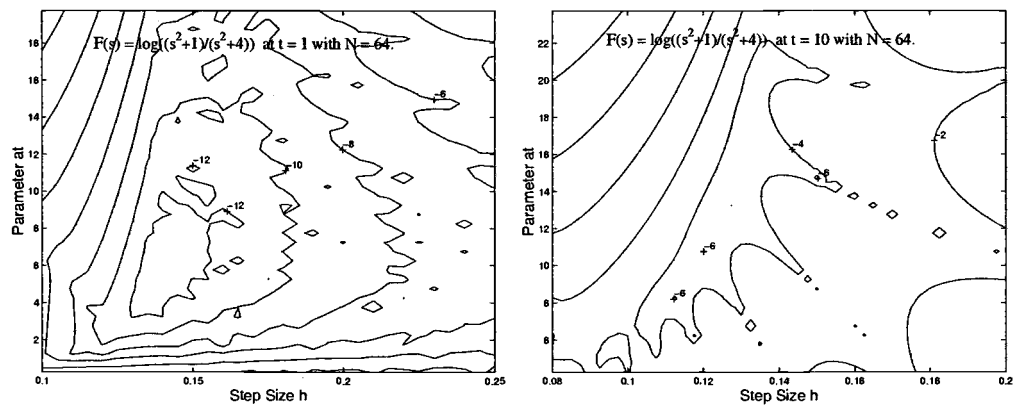

Figure 4.12: The same as Figure 4.11 except for $F(s) = \log((s^2 + 1)/(s^2 + 4))$.

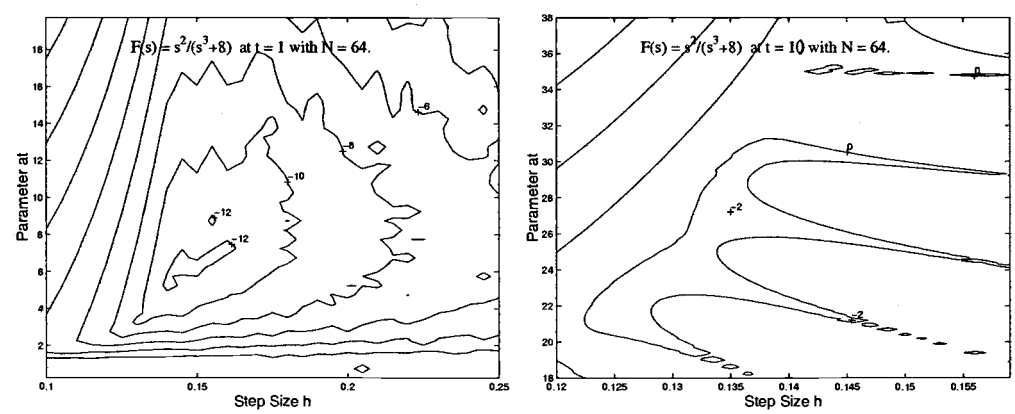

Figure 4.13: The same as Figure 4.11 except for $F(s) = s^2/(s^3 + 8)$.

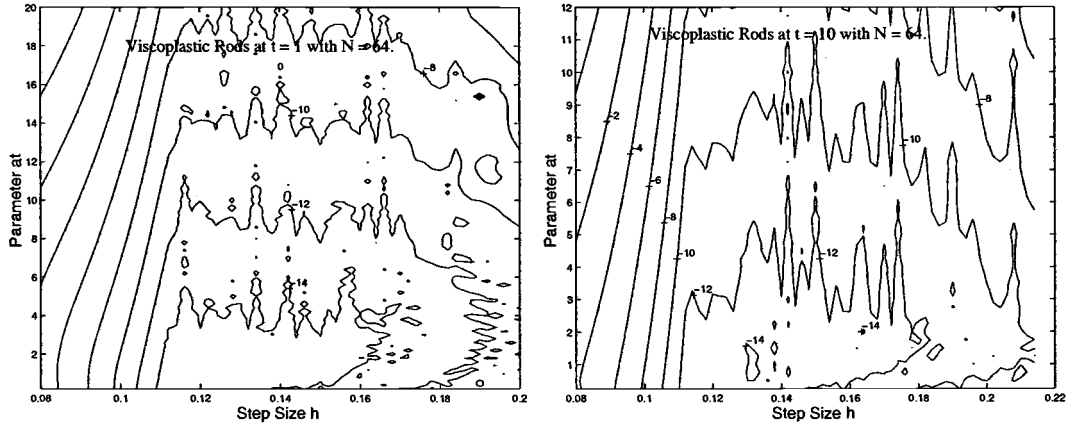

Figure 4.14: Level curves of the relative error when formula DET(IIs) is applied to the viscoplastic rod problem (see Table 2.12) with $N = 64$. The two figures correspond to $t = 1$ (left) and 10 (right).

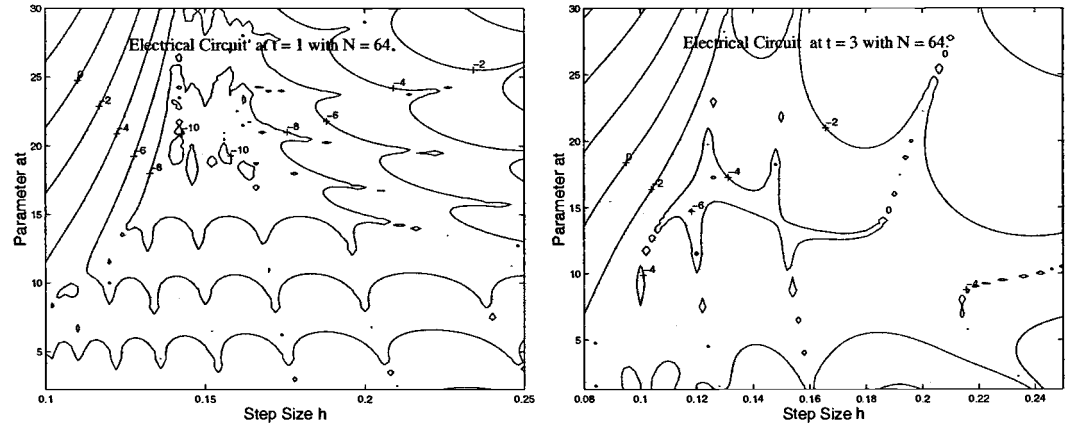

Figure 4.15: The same as Figure 4.14, but the transform is the electrical circuit problem and $t = 1$ (left) and 3 (right).

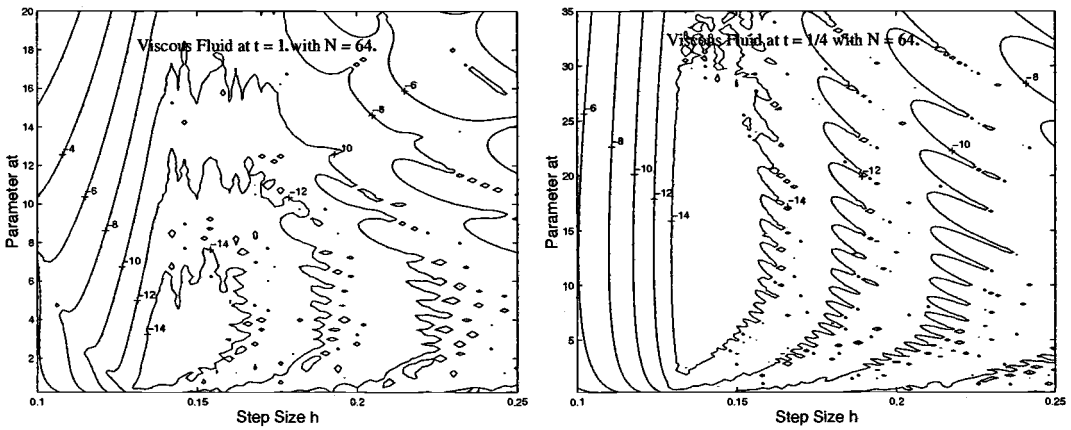

Figure 4.16: The same as Figure 4.14, but the transform is the viscous fluid mechanics problem and $t = 1/4$ (left) and 1 (right).

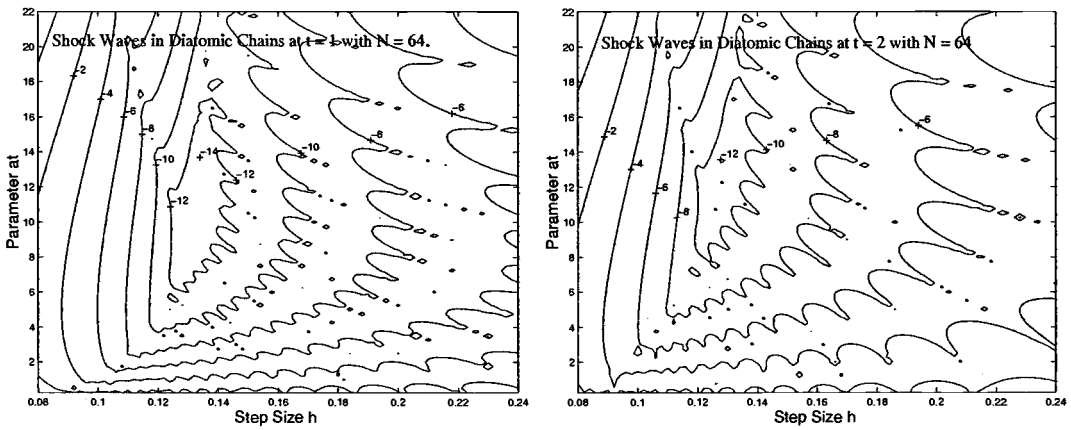

Figure 4.17: The same as Figure 4.14, but the transform is the diatomic chains problem and $t = 1$ (left) and 2 (right).

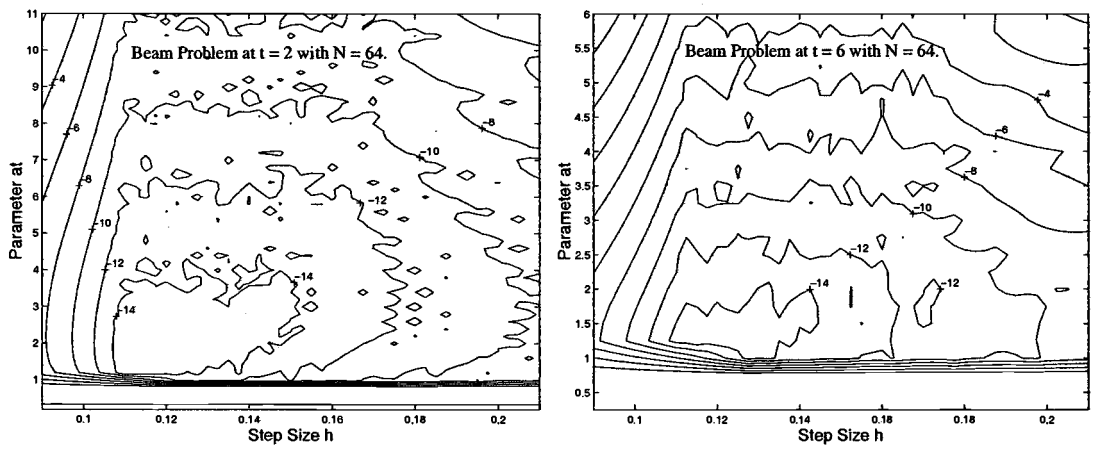

Figure 4.18: The same as Figure 4.14, but the transform is the Timoshenko beam problem and $t = 2$ (left) and 6 (right).

5. CONCLUSION

It is well-known that the numerical inversion of the Laplace transform is a difficult problem. In this dissertation, we have investigated a class of efficient methods for dealing with this problem, namely the trapezoidal-type rules enhanced by three different techniques, to improve accuracy.

The methods in this class are all based on the application of the trapezoidal (or midpoint) rule to the Bromwich integral, a representation of the Laplace inversion formula. The resulting infinite sums typically converge slowly, which makes the method inefficient from a computational point of view. However, by applying certain enhancement techniques, however, the efficiency may be greatly improved.

The three enhancement techniques that we have investigated in this thesis are: convergence acceleration (Chapter 2), deformation of contour (Chapter 3), and a novel change-of-variables technique (Chapter 4). The first two methods are well-known and have been introduced by Dubner and Abate [5], and Talbot [4], respectively. The third method is new: It is based on a transformation method introduced by Ooura and Mori [13, 14] for Fourier-type integrals that we have modified here for application to the Laplace inversion problem.

Even though the methods of Chapter 2 and 3 are well-known, we believe that we have made some contributions to the theory as well as some the practical algorithmic implementation for these methods. These contributions are summarized as follows.

In Chapter 2, we have analyzed the so-called direct integration method, which is nothing but the trapezoidal rule approximation enhanced by convergence acceleration. As convergence accelerators, we considered both the Euler transformation and the epsilon algorithm; the former for theoretical analysis (because it allows a readily computable error bound) and the latter for practical computation.

The accuracy of this method depends critically on a free parameter a (essentially,

it is the position of the line along which the Bromwich integral is evaluated). In order to estimate the optimal value of this parameter, we performed a theoretical error analysis. In particular, we identified three sources of error associated with this method: namely, (a) the discretization error (which arises from approximating the integral by an infinite trapezoidal sum), (b) the truncation error (which is due to the truncation of the infinite sum), and (c) the conditioning error (which occurs in the computation of a finite precision machine).

Consequently, a theorem that represents the discretization error by a contour integral and an assertion that the truncation error is negligible provided the number of evaluation terms is sufficiently large, have been established. Also, when the truncation error is neglected, the upper bounds of the discretization error and conditioning error have been estimated. Owing to these results, we were able to derive the theoretical estimates for the optimal parameter a. Of course, the drawback common to all theoretical approaches is that they rely on a prior information about the solution $f(t)$ that we are seeking, and are therefore of limited practical use.

As an alternative, we have therefore devised a practical algorithm that aims to automatically determine both the optimal parameter a and a sufficiently large number of terms in the trapezoidal series, to ensure that the computed accuracy lies within a user-specified tolerance. This algorithm has been compared to a similar algorithm, owing to D'Amore and coworkers [10, 7], on a large class of test problems. Our conclusion is that the new algorithm performs better for large values of t.

In Chapter 3, we considered Talbot's algorithm, in which the Bromwich line is modified to a different contour. Provided that the new contour is appropriately selected, the trapezoidal sums converge much more rapidly than on the original Bromwich line. Similar to the method of Chapter 2, this method contains a number of free parameters which need to be tuned for optimum accuracy. There are in fact three parameters, all three of which determine the geometric features of Talbot's contour. In Talbot's original

paper [4], estimates for these parameters were given, but those involved a certain amount of heuristic and empirical reasoning, and the parameters may not be truly optimal for all transforms.

Our contribution to Talbot's method lies in the fact that two new and equivalent error representations have been derived. Either can be easily computed and used as an accurate error estimate for this method. Using this error estimate as the objective function, we employed a numerical minimization routine to compute approximate values of the three parameters. Numerical tests have shown that this new algorithm does indeed improve the accuracy of the original Talbot method. On the negative side, however, the minimization routine introduces additional computational work.

In Chapter 4, we introduced a new method, based on a change-of-variables in the Bromwich integral. This transformation is aimed toward making the trapezoidal sums converge more quickly. We have demonstrated the potential of this method on a large number of test examples. As with the other two methods, however, this method has free parameters that need to be selected, and we have not yet addressed the issue of optimal parameter selection in this method.

Throughout the thesis, we have tested our methods and algorithms on a large class of test problems. We used not only test examples taken from tables of the Laplace transform, but also a selection of five practical problems taken from the engineering literature.

BIBLIOGRAPHY

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

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

3. W.T. Weeks. Numerical inversion of the Laplace transform using Laguerre functions. *J. Assoc. Comp. Mach.*, 13:419–429, 1966.

4. A. Talbot. The accurate numerical inversion of Laplace transforms. *J. Inst. Math. Appl.*, 23:97–120, 1979.

5. R. Dubner and J. Abate. Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform. *J. ACM*, 15:115–123, 1968.

6. F. Durbin. An efficient improvement to Dubner and Abate's method. *Comput. J.*, 17:371–376, 1973.

7. L. D'Amore, G. Laccetti, and A. Murli. An implementation of a Fourier series method for the numerical inversion of the Laplace transform. *ACM Trans. Math. Soft.*, 25:279–305, 1999.

8. K.S. Crump. Numerical inversion of Laplace transforms using Fourier series approximation. *ACM Trans. Math. Soft.*, 23:89–96, 1976.

9. G.Honig and U. Hirdes. Algorithm27: A method for the numerical inversion of Laplace transforms. *J. Comp. Appl. Math.*, 10:113–132, 1984.

10. L. D'Amore, G. Laccetti, and A. Murli. Algorithm 796: A Fortran software package for the numerical inversion of the Laplace transform based on a Fourier series method. *ACM Trans. Math. Soft.*, 25:306–315, 1999.

11. A. Murli and M. Rizzardi. Algorithm 682: Talbot's method for the Laplace inversion problem. *ACM Trans. Math. Soft.*, 16:158–168, 1990.

12. M. Rizzardi. A modification of Talbot's method for the simultaneous approximation of several values of the inverse Laplace transform. *ACM Trans. Math. Soft.*, 21:346–371, 1995.

13. T. Ooura and M. Mori. The double exponential formula for oscillatory functions over the half infinite interval. *J. Comp. Appl. Math.*, 38:353–360, 1991.

14. T. Ooura and M. Mori. A robust double exponential formula for Fourier-type integrals. *J. Comp. Appl. Math.*, 112:229–241, 1999.

15. Gustav Doetsch. *Introduction to the Theory and Application of the Laplace Transformation.* Springer-Verlag, New York, 1970.

16. D. V. Widder. *The Laplace Transform.* Princeton University Press, Princeton, 1941.

17. Claude Brezinski. *Extrapolation Methods: Theory and Practice.* Elsevier Science, New York, 1991.

18. D. A. Smith and W. F. Ford. Acceleration of linear and logarithm convergence. *SIAM J. Numer. Aanl.*, 16:223–240, 1979.

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

20. K. Knopp. *Theory and Application of Infinite Series.* Dover, New York, 1951.

21. C.C. Grosjean. On a transformation procedure for the step-by-step conversion of an alternating series into a series of purely positive terms. *Simon Stevin*, 38:149–172, 1965.

22. K. Knopp. *Infinite Sequences and Series.* Dover, New York, 1956.

23. J.H. Wilkinson. *Rounding Errors in Algebraic Processes.* Prentic-Hall, England, 1963.

24. Nicholas J. Higham. *Accuracy and Stability of Numerical Algorithms.* SIAM, New York, 1996.

25. R. W. Hamming. *Numerical Methods for Scientists and Engineers, Second Ed.* McGraw-Hill, Tokyo, 1973.

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

27. J.A.C. Weideman. Numerical integration of periodic functions: A few examples. *MAA Monthly*, 109:21–36, 2002.

28. Rainer Kress. *Numerical Analysis.* Springer-Verlag, New York, 1998.

29. E.T. Goodwin. The evaluation of integrals of the form $\int_{-\infty}^{\infty} f(x)\mathrm{e}^{-x^2}\,dx$. *Proc. Camb. Philos. Soc.*, 45:241–245, 1949.

30. A. Murli and V. Patruno. Un metodo per l'inversione numerica della trasformata di laplace. *Calcolo*, XV:51–58, 1978.

31. R.M. Corless, G.H. Gonnet, D.E.G. Hare, D.J. Jeffrey, and D.E. Knuth. On the Lambert W function. *Adv. Comp. Math.*, 5:329–359, 1996.

32. H. Takahasi and M. Mori. Double exponential formulas for numerical integration. *Publ. Res. Inst. Math. Sci.*, 9:721–741, 1974.

33. M. Mori. The double exponential formulas for numerical integration over the half infinite interval. *Int. Ser. Numer. Math.*, 86:3–18, 1988.

APPENDIX

THE SUM $\sum_{N=1}^{\infty} N^M \mathrm{E}^{-\lambda N}$

Lemma .0.1. *Let λ and m be two positive real numbers. Then*

$$\sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} \sim \begin{cases} \frac{m!}{\lambda^{m+1}}, & \text{for a fixed } \lambda \text{ and as } m \longrightarrow \infty, \\ \mathrm{e}^{-\lambda}, & \text{for a fixed } m \text{ and as } \lambda \longrightarrow \infty. \end{cases} \tag{0.1}$$

Proof: Suppose that m is a fixed number. Given $\epsilon > 0$, there exists an integer $K > 0$ such that

$$\sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} = \mathrm{e}^{-\lambda} + \sum_{n=2}^{K} n^m \mathrm{e}^{-\lambda n} + b, \tag{0.2}$$

where $|b| < \epsilon$. Since ϵ is arbitrary and $\lambda \longrightarrow \infty$, the second term in (0.2) satisfies:

$$\sum_{n=2}^{K} n^m \mathrm{e}^{-\lambda n} \sim 0, \quad \text{as } \lambda \longrightarrow \infty.$$

Hence

$$\sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} \sim \mathrm{e}^{-\lambda}, \quad \text{if } \lambda \longrightarrow \infty, \tag{0.3}$$

which is the second case of (0.1).

On the other hand, let

$$g(x) = x^m \mathrm{e}^{-\lambda x}. \tag{0.4}$$

It is easy to see that $g(x)$ has the maximum value at $x = \frac{m}{\lambda}$. Furthermore, $g(x)$ is increasing if $0 \leq x < \frac{m}{\lambda}$ and decreasing if $x > \frac{m}{\lambda}$.

Suppose that λ is a fixed number and $m \longrightarrow \infty$ (hence $\frac{m}{\lambda} \gg 1$). Let us write

$$\sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} = \sum_{n=0}^{[\frac{m}{\lambda}]-1} n^m \mathrm{e}^{-\lambda n} + \sum_{[\frac{m}{\lambda}]+1}^{\infty} n^m \mathrm{e}^{-\lambda n} + \left[\frac{m}{\lambda}\right]^m \mathrm{e}^{-\lambda [\frac{m}{\lambda}]}. \tag{0.5}$$

By the above observations of the function $g(x)$, the following two inequalities hold:

$$\int_0^{[\frac{m}{\lambda}]-1} x^m \mathrm{e}^{-\lambda x}\, dx \leq \sum_{n=0}^{[\frac{m}{\lambda}]-1} n^m \mathrm{e}^{-\lambda n} \leq \int_0^{[\frac{m}{\lambda}]} x^m \mathrm{e}^{-\lambda x}\, dx \tag{0.6}$$

and

$$\begin{aligned}\int_{[\frac{m}{\lambda}]+1}^{\infty} x^m \mathrm{e}^{-\lambda x}\,dx &\leq \sum_{[\frac{m}{\lambda}]+1}^{\infty} n^m \mathrm{e}^{-\lambda n} \\ &\leq \int_{[\frac{m}{\lambda}]+1}^{\infty} x^m \mathrm{e}^{-\lambda x}\,dx + \left(\left[\frac{m}{\lambda}\right]+1\right)^m \mathrm{e}^{-\lambda([\frac{m}{\lambda}]+1)}. \qquad (0.7)\end{aligned}$$

Applying the above two inequalities and the equation (0.5), we have

$$\begin{aligned}&\int_0^{\infty} x^m \mathrm{e}^{-\lambda x}\,dx - \int_{[\frac{m}{\lambda}]-1}^{[\frac{m}{\lambda}]+1} x^m \mathrm{e}^{-\lambda x}\,dx + \left[\frac{m}{\lambda}\right]^m \mathrm{e}^{-\lambda[\frac{m}{\lambda}]} \leq \sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} \\ &\quad \leq \int_0^{\infty} x^m \mathrm{e}^{-\lambda x}\,dx + \left[\frac{m}{\lambda}\right]^m \mathrm{e}^{-\lambda[\frac{m}{\lambda}]} + \left(\left[\frac{m}{\lambda}\right]+1\right)^m \mathrm{e}^{-\lambda([\frac{m}{\lambda}]+1)}. \qquad (0.8)\end{aligned}$$

By integration by parts, we have

$$\int_0^{\infty} x^m \mathrm{e}^{-\lambda x}\,dx = \frac{m!}{\lambda^{m+1}}, \qquad (0.9)$$

and by the Stirling formula, as $m \longrightarrow \infty$,

$$\left[\frac{m}{\lambda}\right]^m \mathrm{e}^{-\lambda[\frac{m}{\lambda}]} \sim \left(\frac{m}{\lambda}\right)^m \mathrm{e}^{-m} \sim \frac{m!}{\lambda^m \sqrt{2\pi m}} = o\left(\frac{m!}{\lambda^{m+1}}\right),$$

$$\left(\left[\frac{m}{\lambda}\right]+1\right)^m \mathrm{e}^{-\lambda([\frac{m}{\lambda}]+1)} = o\left(\frac{m!}{\lambda^{m+1}}\right),$$

and

$$\int_{[\frac{m}{\lambda}]-1}^{[\frac{m}{\lambda}]+1} x^m \mathrm{e}^{-\lambda x}\,dx \sim 2\left[\frac{m}{\lambda}\right]^m \mathrm{e}^{-\lambda[\frac{m}{\lambda}]} = o\left(\frac{m!}{\lambda^{m+1}}\right)$$

hold. Therefore,

$$\sum_{n=1}^{\infty} n^m \mathrm{e}^{-\lambda n} \sim \frac{m!}{\lambda^{m+1}} \quad \text{as} \quad m \longrightarrow \infty,$$

which is the first case of (0.1). The proof is therefore completed. □