Hindawi Publishing Corporation
International Journal of Mathematics and Mathematical Sciences
Volume 2009, Article ID 898195, 38 pages
doi:10.1155/2009/898195

Research Article

Inversion of the Laplace Transform from the Real Axis Using an Adaptive Iterative Method

Sapto W. Indratno and Alexander G. Ramm

Department of Mathematics, Kansas State University, Manhattan, KS 66506-2602, USA

Correspondence should be addressed to Alexander G. Ramm, ramm@math.ksu.edu

Received 24 July 2009; Accepted 19 October 2009

Recommended by Thomas Witelski

A new method for inverting the Laplace transform from the real axis is formulated. This method is based on a quadrature formula. We assume that the unknown function $f(t)$ is continuous with (known) compact support. An adaptive iterative method and an adaptive stopping rule, which yield the convergence of the approximate solution to $f(t)$, are proposed in this paper.

1. Introduction

Consider the Laplace transform

$$\mathcal{L}f(p) := \int_0^\infty e^{-pt} f(t)dt = F(p), \quad \operatorname{Re} p > 0, \tag{1.1}$$

where $\mathcal{L} : X_{0,b} \to L^2[0,\infty)$,

$$X_{0,b} := \left\{ f \in L^2[0,\infty) \mid \operatorname{supp} f \subset [0,b) \right\}, \quad b > 0. \tag{1.2}$$

We assume in (1.2) that f has compact support. This is not a restriction practically. Indeed, if $\lim_{t\to\infty} f(t) = 0$, then $|f(t)| < \delta$ for $t > t_\delta$, where $\delta > 0$ is an arbitrary small number. Therefore, one may assume that supp$f \subset [0, t_\delta]$, and treat the values of f for $t > t_\delta$ as noise. One may also note that if $f \in L^1(0,\infty)$, then

$$F(p) := \int_0^\infty f(t)e^{-pt}dt = \int_0^b f(t)e^{-pt}dt + \int_b^\infty f(t)e^{-pt}dt := F_1(p) + F_2(p), \tag{1.3}$$

and $|F_2(p)| \le e^{-bp}\delta$, where $\int_b^\infty |f(t)|dt \le \delta$. Therefore, the contribution of the "tail" $f_b(t)$ of f,

$$f_b(t) := \begin{cases} 0, & t < b, \\ f(t), & t \ge b, \end{cases} \tag{1.4}$$

can be considered as noise if $b > 0$ is large and $\delta > 0$ is small. We assume in (1.2) that $f \in L^2[0,\infty)$. One may also assume that $f \in L^1[0,\infty)$, or that $|f(t)| \le c_1 e^{c_2 t}$, where c_1, c_2 are positive constants. If the last assumption holds, then one may define the function $g(t) := f(t)e^{-(c_2+1)t}$. Then $g(t) \in L^1[0,\infty)$, and its Laplace transform $G(p) = F(p + c_2 + 1)$ is known on the interval $[c_2 + 1, c_2 + 1 + b]$ of real axis if the Laplace transform $F(p)$ of $f(t)$ is known on the interval $[0, b]$. Therefore, our inversion methods are applicable to these more general classes of functions f as well.

The operator $\mathcal{L} : X_{0,b} \to L^2[0,\infty)$ is compact. Therefore, the inversion of the Laplace transform (1.1) is an ill-posed problem (see [1, 2]). Since the problem is ill-posed, a regularization method is needed to obtain a stable inversion of the Laplace transform. There are many methods to solve (1.1) stably: variational regularization, quasisolutions, and iterative regularization (see, e.g., [1–4]). In this paper we propose an adaptive iterative method based on the Dynamical Systems Method (DSM) developed in [2, 4]. Some methods have been developed earlier for the inversion of the Laplace transform (see [5–8]). In many papers the data $F(p)$ are assumed exact and given on the complex axis. In [9] it is shown that the results of the inversion of the Laplace transform from the complex axis are more accurate than these of the inversion of the Laplace transform from the real axis. The reason is the ill-posedness of the Laplace transform inversion from the real axis. A survey regarding the methods of the Laplace transform inversion has been given in [6]. There are several types of the Laplace inversion method compared in [6]. The inversion formula for the Laplace transform that is well known

$$f(t) = \frac{1}{2\pi i}\int_{\sigma - i\infty}^{\sigma + i\infty} F(p)e^{pt}dp, \quad \sigma > 0, \tag{1.5}$$

is used in some of these methods, and then $f(t)$ is computed by some quadrature formulas, and many of these formulas can be found in [10, 11]. Moreover, the ill-posedness of the Laplace transform inversion is not discussed in all the methods compared in [6]. The approximate $f(t)$, obtained by these methods when the data are noisy, may differ significantly from $f(t)$. There are some papers in which the inversion of the Laplace transform from the real axis was studied (see [9, 12–19]). In [12, 17] a method based on the Mellin transform is developed. In this method the Mellin transform of the data $F(p)$ is calculated first and then inverted for $f(t)$. In [13] a Fourier series method for the inversion of Laplace transform from the real axis is developed. The drawback of this method comes from the ill-conditioning of the discretized problem. It is shown in [13] that if one uses some basis functions in $X_{0,b}$, the problem becomes extremely ill-conditioned if the number m of the basis functions exceeds 20. In [15] a reproducing kernel method is used for the inversion of the Laplace transform. In the numerical experiments in [15] the authors use double and multiple precision methods to obtain high accuracy inversion of the Laplace transform. The usage of the multiple precision increases the computation time significantly which is observed in [15], so this method may be not efficient in practice. A detailed description of the multiple precision technique can be found in [20, 21]. Moreover, the Laplace transform inversion with perturbed data is not discussed in [15]. In [19] the authors develop an inversion formula, based on the eigenfunction expansion for the Laplace transform. The difficulties with this method are (i) the inversion formula is not applicable when the data are noisy, (ii) even for exact data the inversion formula is not suitable for numerical implementation.

The Laplace transform as an operator from C_{0k} into L^2, where $C_{0k} = \{f(t) \in C[0,+\infty) \mid \operatorname{supp} f \subset [0,k)\}$, $k = \text{const} > 0$, $L^2 := L^2[0,\infty)$, is considered in [14]. The finite difference method is used in [14] to discretize the problem, where the size of the linear algebraic system obtained by this method is fixed at each iteration, so the computation time increases if one uses large linear algebraic systems. The method of choosing the size of the linear algebraic system is not given in [14]. Moreover, the inversion of the Laplace transform when the data $F(p)$ is given only on a finite interval $[0,d]$, $d > 0$, is not discussed in [14].

The novel points in our paper are the following:

(1) the representation of the approximation solution (2.69) of the function $f(t)$ which depends only on the kernel of the Laplace transform,

(2) the adaptive iterative scheme (2.72) and adaptive stopping rule (2.83), which generate the regularization parameter, the discrete data $F_\delta(p)$, and the number of terms in (2.69), needed for obtaining an approximation of the unknown function $f(t)$.

We study the inversion problem using the pair of spaces $(X_{0,b}, L^2[0,d])$, where $X_{0,b}$ is defined in (1.2), develop an inversion method, which can be easily implemented numerically, and demonstrate in the numerical experiments that our method yields the results comparable in accuracy with the results, presented in the literature, for example, with the double precision results given in paper [15].

The smoothness of the kernel allows one to use the compound Simpson's rule in approximating the Laplace transform. Our approach yields a representation (2.69) of the approximate inversion of the Laplace transform. A number of terms in approximation (2.69) and the regularization parameter are generated automatically by the proposed adaptive iterative method. Our iterative method is based on the iterative method proposed in [22]. The adaptive stopping rule we propose here is based on the discrepancy-type principle, established in [23, 24]. This stopping rule yields convergence of the approximation (2.69) to $f(t)$ when the noise level $\delta \to 0$.

A detailed derivation of our inversion method is given in Section 2. In Section 3 some results of the numerical experiments are reported. These results demonstrate the efficiency and stability of the proposed method.

2. Description of the Method

Let $f \in X_{0,b}$. Then (1.1) can be written as

$$(\mathcal{L}f)(p) := \int_0^b e^{-pt} f(t)dt = F(p), \quad 0 \leq p. \tag{2.1}$$

Let us assume that the data $F(p)$, the Laplace transform of f, are known only for $0 \leq p \leq d < \infty$. Consider the mapping $\mathcal{L}_m : L^2[0,b] \to \mathbb{R}^{m+1}$, where

$$\left(\mathcal{L}_m f\right)_i := \int_0^b e^{-p_i t} f(t) dt = F(p_i), \quad i = 0,1,2,\ldots,m, \tag{2.2}$$

$$p_i := ih, \quad i = 0,1,2,\ldots,m, \ h := \frac{d}{m}, \tag{2.3}$$

and m is an even number which will be chosen later. Then the unknown function $f(t)$ can be obtained from a finite-dimensional operator equation (2.2). Let

$$\langle u, v\rangle_{W^m} := \sum_{j=0}^{m} w_j^{(m)} u_j v_j, \qquad \|u\|_{W^m} := \langle u, u\rangle_{W^m} \tag{2.4}$$

be the inner product and norm in $\mathbb{R}^{m+1}$, respectively, where $w_j^{(m)}$ are the weights of the compound Simpson's rule (see [10, page 58]), that is,

$$w_j^{(m)} := \begin{cases} \dfrac{h}{3}, & j = 0, m, \\ \dfrac{4h}{3}, & j = 2l-1, \ l = 1,2,\ldots,\dfrac{m}{2}, \quad h = \dfrac{d}{m}, \\ \dfrac{2h}{3}, & j = 2l, \ l = 1,2,\ldots,\dfrac{m-2}{2}, \end{cases} \tag{2.5}$$

where m is an even number. Then

$$\begin{aligned} \langle \mathcal{L}_m g, v\rangle_{W^m} &= \sum_{j=0}^{m} w_j^{(m)} \int_0^b e^{-p_j t} g(t) dt v_j \\ &= \int_0^b g(t) \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} v_j dt = \langle g, \mathcal{L}_m^* v\rangle_{X_{0,b}}, \end{aligned} \tag{2.6}$$

where

$$\mathcal{L}_m^* v = \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} v_j, \quad v := \begin{pmatrix} v_0 \\ v_1 \\ \vdots \\ v_m \end{pmatrix} \in \mathbb{R}^{m+1}, \tag{2.7}$$

$$\langle g, h\rangle_{X_{0,b}} := \int_0^b g(t) h(t) dt. \tag{2.8}$$

It follows from (2.2) and (2.7) that

$$(\mathcal{L}_m^*\mathcal{L}_m g)(t) = \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} \int_0^b e^{-p_j z} g(z)\,dz := \left(T^{(m)} g\right)(t), \tag{2.9}$$

$$\mathcal{L}_m\mathcal{L}_m^* v = \begin{pmatrix} \int_0^b e^{-p_0 t} \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} v_j dt \\ \int_0^b e^{-p_1 t} \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} v_j dt \\ \vdots \\ \int_0^b e^{-p_m t} \sum_{j=0}^{m} w_j^{(m)} e^{-p_j t} v_j dt \end{pmatrix} := Q^{(m)} v, \tag{2.10}$$

where

$$\left(Q^{(m)}\right)_{ij} := w_j^{(m)} \int_0^b e^{-(p_i+p_j)t} dt = w_j^{(m)} \frac{1 - e^{-b(p_i+p_j)}}{p_i + p_j}, \quad i, j = 0, 1, 2, \ldots, m. \tag{2.11}$$

From (2.7) we get Range$[\mathcal{L}_m^*]$ = span$\{w_j^{(m)} k(p_j, \cdot, 0)\}_{j=0}^{m}$, where

Lemma 2.1. *Let $w_j^{(m)}$ be defined in* (2.5)*. Then*

$$\sum_{j=0}^{m} w_j^{(m)} = d, \tag{2.12}$$

for any even number m.

Proof. From definition (2.5) one gets

$$\begin{aligned}\sum_{j=0}^{m} w_j^{(m)} &= w_0^{(m)} + w_m^{(m)} + \sum_{j=1}^{m/2} w_{2j-1}^{(m)} + \sum_{j=1}^{(m-2)/2} w_{2j}^{(m)} \\ &= \frac{2h}{3} + \sum_{j=1}^{m/2} \frac{4h}{3} + \sum_{j=1}^{(m-2)/2} \frac{2h}{3} \\ &= \frac{2h}{3} + \frac{2hm}{3} + \frac{h(m-2)}{3} = hm = \frac{d}{m} m = d.\end{aligned} \tag{2.13}$$

Lemma 2.1 is proved. □

Lemma 2.2. *The matrix $Q^{(m)}$, defined in* (2.11)*, is positive semidefinite and self-adjoint in $\mathbb{R}^{m+1}$ with respect to the inner product* (2.4)*.*

Proof. Let

$$
\begin{gathered}
(H_m)_{ij} := \int_0^b e^{-(p_i+p_j)t}dt = \frac{1-e^{-b(p_i+p_j)}}{p_i+p_j}, \\
(D_m)_{ij} = \begin{cases} w_i^{(m)}, & i=j, \\ 0, & \text{otherwise}, \end{cases}
\end{gathered}
\tag{2.14}
$$

$w_j^{(m)}$ are defined in (2.5). Then $\langle D_m H_m D_m u, v\rangle_{\mathbb{R}^{m+1}} = \langle u, D_m H_m D_m v\rangle_{\mathbb{R}^{m+1}}$, where

$$
\langle u, v\rangle_{\mathbb{R}^{m+1}} := \sum_{j=0}^{m} u_j v_j, \quad u, v \in \mathbb{R}^{m+1}. \tag{2.15}
$$

We have

$$
\begin{aligned}
\left\langle Q^{(m)}u, v\right\rangle_{W^m} &= \sum_{j=0}^{m} w_j^{(m)} \left(Q^{(m)}u\right)_j v_j = \sum_{j=0}^{m} (D_m H_m D_m u)_j v_j \\
&= \langle D_m H_m D_m u, v\rangle_{\mathbb{R}^{m+1}} = \langle u, D_m H_m D_m v\rangle_{\mathbb{R}^{m+1}} \\
&= \sum_{j=0}^{m} u_j (D_m H_m D_m v)_j = \sum_{j=0}^{m} u_j w_j^{(m)} (H_m D_m v)_j \\
&= \left\langle u, Q^{(m)}v\right\rangle_{W^m}.
\end{aligned}
\tag{2.16}
$$

Thus, $Q^{(m)}$ is self-adjoint with respect to inner product (2.4). We have

$$
\begin{aligned}
(H_m)_{ij} &= \int_0^b e^{-(p_i+p_j)t}dt = \int_0^b e^{-p_i t} e^{-p_j t} dt \\
&= \langle \phi_i, \phi_j\rangle_{X_{0,b}}, \quad \phi_i(t) := e^{-p_i t},
\end{aligned}
\tag{2.17}
$$

where $\langle \cdot, \cdot\rangle_{X_{0,b}}$ is defined in (2.8). This shows that H_m is a Gram matrix. Therefore,

$$
\langle H_m u, u\rangle_{\mathbb{R}^{m+1}} \geq 0, \quad \forall u \in \mathbb{R}^{m+1}. \tag{2.18}
$$

This implies

$$
\left\langle Q^{(m)}u, u\right\rangle_{W^m} = \left\langle Q^{(m)}u, D_m u\right\rangle_{\mathbb{R}^{m+1}} = \langle H_m D_m u, D_m u\rangle_{\mathbb{R}^{m+1}} \geq 0. \tag{2.19}
$$

Thus, $Q^{(m)}$ is a positive semidefinite and self-adjoint matrix with respect to the inner product (2.4). □

Lemma 2.3. *Let $T^{(m)}$ be defined in* (2.9). *Then $T^{(m)}$ is self-adjoint and positive semidefinite operator in $X_{0,b}$ with respect to inner product* (2.8).

Proof. From definition (2.9) and inner product (2.8) we get

$$\begin{aligned}\left\langle T^{(m)}g,h\right\rangle_{X_{0,b}} &= \int_0^b \sum_{j=0}^m w_j^{(m)} e^{-p_j t}\int_0^b e^{-p_j z} g(z)dz\, h(t)dt \\ &= \int_0^b g(z)\sum_{j=0}^m w_j^{(m)} e^{-p_j z}\int_0^b e^{-p_j t}h(t)dt\, dz \\ &= \left\langle g,T^{(m)}h\right\rangle_{X_{0,b}}.\end{aligned} \tag{2.20}$$

Thus, $T^{(m)}$ is a self-adjoint operator with respect to inner product (2.8). Let us prove that $T^{(m)}$ is positive semidefinite. Using (2.9), (2.5), (2.4), and (2.8), one gets

$$\begin{aligned}\left\langle T^{(m)}g,g\right\rangle_{X_{0,b}} &= \int_0^b \sum_{j=0}^m w_j^{(m)} e^{-p_j t}\int_0^b e^{-p_j z} g(z)dz\, g(t)dt \\ &= \sum_{j=0}^m w_j^{(m)} \int_0^b e^{-p_j z}g(z)dz\int_0^b e^{-p_j t}g(t)dt \\ &= \sum_{j=0}^m w_j^{(m)}\left(\int_0^b e^{-p_j z}g(z)dz\right)^2 \geq 0.\end{aligned} \tag{2.21}$$

Lemma 2.3 is proved. □

$$k(p,t,z) := e^{-p(t+z)}. \tag{2.22}$$

Let us approximate the unknown $f(t)$ as follows:

$$f(t) \approx \sum_{j=0}^m c_j^{(m)} w_j^{(m)} e^{-p_j t} = T_{a,m}^{-1}\mathcal{L}_m^* F^{(m)} := f_m(t), \tag{2.23}$$

where p_j are defined in (2.3), $T_{a,m}$ is defined in (2.30), and $c_j^{(m)}$ are constants obtained by solving the linear algebraic system:

$$\left(aI + Q^{(m)}\right)c^{(m)} = F^{(m)}, \tag{2.24}$$

where $Q^{(m)}$ is defined in (2.10),

$$c^{(m)} := \begin{pmatrix} c_0^{(m)} \\ c_1^{(m)} \\ \vdots \\ c_m^{(m)} \end{pmatrix}, \qquad F^{(m)} := \begin{pmatrix} F(p_0) \\ F(p_1) \\ \vdots \\ F(p_m) \end{pmatrix}. \tag{2.25}$$

To prove the convergence of the approximate solution $f(t)$, we use the following estimates, which are proved in [4], so their proofs are omitted.

Estimates (2.26) and (2.27) are used in proving inequality (2.88), while estimates (2.28) and (2.29) are used in the proof of Lemmas 2.9 and 2.10, respectively.

Lemma 2.4. *Let $T^{(m)}$ and $Q^{(m)}$ be defined in* (2.9) *and* (2.10), *respectively. Then, for $a > 0$, the following estimates hold:*

$$\left\| Q_{a,m}^{-1} \mathcal{L}_m \right\| \le \frac{1}{2\sqrt{a}}, \tag{2.26}$$

$$a\left\| Q_{a,m}^{-1} \right\| \le 1, \tag{2.27}$$

$$\left\| T_{a,m}^{-1} \right\| \le \frac{1}{a}, \tag{2.28}$$

$$\left\| T_{a,m}^{-1} \mathcal{L}_m^{*} \right\| \le \frac{1}{2\sqrt{a}}, \tag{2.29}$$

where

$$Q_{a,m} := Q^{(m)} + aI, \qquad T_{a,m} := T^{(m)} + aI, \tag{2.30}$$

I is the identity operator and $a =$ const > 0.

Let us formulate an iterative method for obtaining the approximation solution of $f(t)$ with the exact data $F(p)$. Consider the following iterative scheme:

$$u_n(t) = qu_{n-1}(t) + (1-q)T_{a_n}^{-1}\mathcal{L}^{*}F, \quad u_0(t) = 0, \tag{2.31}$$

where $\mathcal{L}^*$ is the adjoint of the operator $\mathcal{L}$, that is,

$$(\mathcal{L}^* g)(t) = \int_0^d e^{-pt} g(p)\,dp, \tag{2.32}$$

$$\begin{aligned}(Tf)(t) := (\mathcal{L}^*\mathcal{L}f)(t) &= \int_0^b \int_0^d k(p,t,z)\,dp\,f(z)\,dz \\ &= \int_0^b \frac{f(z)}{t+z}\left(1 - e^{-d(t+z)}\right) dz,\end{aligned} \tag{2.33}$$

$k(p,t,z)$ is defined in (2.22),

$$T_a := aI + T, \quad a > 0, \tag{2.34}$$

$$a_n := q a_{n-1}, \quad a_0 > 0, \; q \in (0,1). \tag{2.35}$$

Lemma 2.5 together with Lemma 2.7 with $g(a) = a\|T_a^{-1} f\|$ yields

Lemma 2.5. *Let T_a be defined in* (2.34), *$\mathcal{L}f = F$, and $f \perp \mathcal{N}(\mathcal{L})$, where $\mathcal{N}(\mathcal{L})$ is the null space of $\mathcal{L}$. Then*

$$a\left\|T_a^{-1} f\right\| \longrightarrow 0 \quad \text{as } a \longrightarrow 0. \tag{2.36}$$

Proof. Since $f \perp \mathcal{N}(\mathcal{L})$, it follows from the spectral theorem that

$$\lim_{a\to 0} a^2 \left\|T_a^{-1} f\right\|^2 = \lim_{a\to 0} \int_0^\infty \frac{a^2}{(a+s)^2} d\langle E_s f, f\rangle = \left\|P_{\mathcal{N}(\mathcal{L})} f\right\|^2 = 0, \tag{2.37}$$

where E_s is the resolution of the identity corresponding to $\mathcal{L}^*\mathcal{L}$, and P is the orthogonal projector onto $\mathcal{N}(\mathcal{L})$.

Lemma 2.5 is proved. □

Theorem 2.6. *Let $\mathcal{L}f = F$, and u_n be defined in* (2.31). *Then*

$$\lim_{n\to\infty} \|f - u_n\| = 0. \tag{2.38}$$

Proof. By induction we get

$$u_n = \sum_{j=0}^{n-1} \omega_j^{(n)} T_{a_{j+1}}^{-1} \mathcal{L}^* F, \tag{2.39}$$

where T_a is defined in (2.34), and

$$\omega_j^{(n)} := q^{n-j-1} - q^{n-j}. \tag{2.40}$$

Using the identities

$$\begin{gathered}\mathcal{L}f = F,\\ T_a^{-1}\mathcal{L}^*\mathcal{L} = T_a^{-1}(T + aI - aI) = I - aT_a^{-1},\\ \sum_{j=0}^{n-1}\omega_j^{(n)} = 1 - q^n,\end{gathered} \tag{2.41}$$

we get

$$\begin{aligned} f - u_n &= f - \sum_{j=0}^{n-1}\omega_j^{(n)} f + \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}T_{a_{j+1}}^{-1} f\\ &= q^n f + \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}T_{a_{j+1}}^{-1} f.\end{aligned} \tag{2.42}$$

Therefore,

$$\|f - u_n\| \le q^n\|f\| + \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}\left\|T_{a_{j+1}}^{-1} f\right\|. \tag{2.43}$$

To prove relation (2.38) the following lemma is needed. □

Lemma 2.7. *Let $g(x)$ be a continuous function on $(0,\infty)$, $c > 0$ and $q \in (0,1)$ constants. If*

$$\lim_{x\to 0^+} g(x) = g(0) := g_0, \tag{2.44}$$

then

$$\lim_{n\to\infty}\sum_{j=0}^{n-1}\left(q^{n-j-1} - q^{n-j}\right)g\left(cq^{j+1}\right) = g_0. \tag{2.45}$$

Proof. Let

$$F_l(n) := \sum_{j=1}^{l-1}\omega_j^{(n)} g\left(cq^j\right), \tag{2.46}$$

where $\omega_j^{(n)} := q^{n-j} - q^{n+1-j}$. Then

$$\left|F_{n+1}(n) - g_0\right| \le |F_l(n)| + \left|\sum_{j=l}^{n}\omega_j^{(n)} g\left(cq^j\right) - g_0\right|. \tag{2.47}$$

Take $\epsilon > 0$ arbitrarily small. For sufficiently large fixed $l(\epsilon)$ one can choose $n(\epsilon) > l(\epsilon)$, such that

$$\left|F_{l(\epsilon)}(n)\right| \leq \frac{\epsilon}{2}, \quad \forall n > n(\epsilon), \tag{2.48}$$

because $\lim_{n\to\infty} q^n = 0$. Fix $l = l(\epsilon)$ such that $|g(cq^j) - g_0| \leq \epsilon/2$ for $j > l(\epsilon)$. This is possible because of (2.44). One has

$$\begin{aligned} \left|F_{l(\epsilon)}(n)\right| &\leq \frac{\epsilon}{2}, \quad n > n(\epsilon) > l(\epsilon), \\ \left|\sum_{j=l(\epsilon)}^{n} \omega_j^{(n)} g\left(cq^j\right) - g_0\right| &\leq \sum_{j=l(\epsilon)}^{n} \omega_j^{(n)} \left|g\left(cq^{j+1}\right) - g_0\right| + \left|\sum_{j=l(\epsilon)}^{n} \omega_j^{(n)} - 1\right| |g_0| \\ &\leq \frac{\epsilon}{2} \sum_{j=l(\epsilon)}^{n} \omega_j^{(n)} + q^{n-l(\epsilon)} |g_0| \\ &\leq \frac{\epsilon}{2} + |g_0| q^{n-l(\epsilon)} \leq \epsilon, \end{aligned} \tag{2.49}$$

if $n(\epsilon)$ is sufficiently large. Here we have used the relation

$$\sum_{j=l}^{n} \omega_j^{(n)} = 1 - q^{n+1-l}. \tag{2.50}$$

Since $\epsilon > 0$ is arbitrarily small, relation (2.45) follows.

Lemma 2.7 is proved. □

$$\lim_{n\to\infty} \sum_{j=0}^{n-1} \omega_j^{(n)} a_{j+1} \left\|T_{a_{j+1}}^{-1} f\right\| = 0. \tag{2.51}$$

This together with estimate (2.43) and condition $q \in (0,1)$ yields relation (2.38). Theorem 2.6 is proved.

Lemma 2.9 leads to an adaptive iterative scheme:

$$u_{n,m_n}(t) = q u_{n-1,m_{n-1}} + (1-q) T_{a_n,m_n}^{-1} \mathcal{L}_{m_n}^{*} F^{(m_n)}, \quad u_{0,m_0}(t) = 0, \tag{2.52}$$

where $q \in (0,1)$, a_n are defined in (2.35), $T_{a,m}$ is defined in (2.30), $A_m \mathcal{L}$ is defined in (2.2), and

Lemma 2.8. *Let T and $T^{(m)}$ be defined in* (2.33) *and* (2.9), *respectively. Then*

$$\left\|T - T^{(m)}\right\| \leq \frac{(2bd)^5}{540\sqrt{10} m^4}. \tag{2.53}$$

Proof. From definitions (2.33) and (2.9) we get

$$\begin{aligned}\left|\left(T-T^{(m)}\right)f(t)\right| &\leq \int_0^b \left|\int_0^d k(p,t,z)dp - \sum_{j=0}^{m} w_j^{(m)} k(p_j,t,z)\right| |f(z)|dz \\ &\leq \int_0^b \left|\frac{d^5}{180m^4}\max_{p\in[0,d]}(t+z)^4 e^{-p(t+z)}\right| |f(z)|dz \\ &= \int_0^b \frac{d^5}{180m^4}(t+z)^4|f(z)|dz \leq \frac{d^5}{180m^4}\left(\int_0^b (t+z)^8 dz\right)^{1/2}\|f\|_{X_{0,b}} \\ &= \frac{d^5}{180m^4}\left[\frac{(t+b)^9-t^9}{9}\right]^{1/2}\|f\|_{X_{0,b}},\end{aligned} \tag{2.54}$$

where the following upper bound for the error of the compound Simpson's rule was used (see [10, page 58]). For $f \in C^{(4)}[x_0, x_{2l}]$, $x_0 < x_{2l}$,

$$\left|\int_{x_0}^{x_{2l}} f(x)dx - \frac{h}{3}\left[f_0 + 4\sum_{j=1}^{l} f_{2(j-1)} + 2\sum_{j=1}^{l-1} f_{2j} + f_{x_{2l}}\right]\right| \leq R_l, \tag{2.55}$$

where

$$\begin{gathered} f_j := f(x_j), \quad x_j = x_0 + jh, \quad j = 0,1,2,\dots,2l, \quad h = \frac{x_{2l}-x_0}{2l}, \\ R_l = \frac{(x_{2l}-x_0)^5}{180(2l)^4}\left|f^{(4)}(\xi)\right|, \quad x_0 < \xi < x_{2l}. \end{gathered} \tag{2.56}$$

This implies

$$\left\|\left(T-T^{(m)}\right)f\right\|_{X_{0,b}} \leq \frac{d^5}{540m^4}\left[\frac{(2b)^{10}-2b^{10}}{10}\right]^{1/2}\|f\|_{X_{0,b}} \leq \frac{(2bd)^5}{540\sqrt{10}m^4}\|f\|_{X_{0,b}}, \tag{2.57}$$

so estimate (2.53) is obtained.

Lemma 2.8 is proved. □

Lemma 2.9. *Let* $0 < a < a_0$,

$$m = \kappa\left(\frac{a_0}{a}\right)^{1/4}, \quad \kappa > 0. \tag{2.58}$$

Then

$$\left\|T - T^{(m)}\right\| \leq \frac{(2bd)^5}{540\sqrt{10}a_0\kappa^4}a, \tag{2.59}$$

where T *and* $T^{(m)}$ *are defined in* (2.33) *and* (2.9), *respectively.*

Proof. Inequality (2.59) follows from estimate (2.53) and formula (2.58). □

$$F^{(m)} := \begin{pmatrix} F(p_0) \\ F(p_1) \\ \vdots \\ F(p_m) \end{pmatrix} \in \mathbb{R}^{m+1}, \tag{2.60}$$

p_j are defined in (2.3). In the iterative scheme (2.52) we have used the finite-dimensional operator $T^{(m)}$ approximating the operator T. Convergence of the iterative scheme (2.52) to the solution f of the equation $\mathcal{L}f = F$ is established in the following lemma.

Lemma 2.10. *Let $\mathcal{L}f = F$ and u_{n,m_n} be defined in* (2.52). *If m_n are chosen by the rule*

$$m_n = \left[\left[\kappa\left(\frac{a_0}{a_n}\right)^{1/4}\right]\right], \quad a_n = qa_{n-1},\ q \in (0,1),\ \kappa, a_0 > 0, \tag{2.61}$$

where $[[x]]$ is the smallest even number not less than x, then

$$\lim_{n\to\infty} \|f - u_{n,m_n}\| = 0. \tag{2.62}$$

Proof. Consider the estimate

$$\|f - u_{n,m_n}\| \le \|f - u_n\| + \|u_n - u_{n,m_n}\| := I_1(n) + I_2(n), \tag{2.63}$$

where $I_1(n) := \|f - u_n\|$ and $I_2(n) := \|u_n - u_{n,m_n}\|$. By Theorem 2.6, we get $I_1(n) \to 0$ as $n \to \infty$. Let us prove that $\lim_{n\to\infty} I_2(n) = 0$. Let $U_n := u_n - u_{n,m_n}$. Then, from definitions (2.31) and (2.52), we get

$$U_n = qU_{n-1} + (1-q)\left(T_{a_n}^{-1}\mathcal{L}^*F - T_{a_n,m_n}^{-1}\mathcal{L}_{m_n}^*F^{(m_n)}\right), \quad U_0 = 0. \tag{2.64}$$

By induction we obtain

$$U_n = \sum_{j=0}^{n-1}\omega_j^{(n)}\left(T_{a_{j+1}}^{-1}\mathcal{L}^*F - T_{a_{j+1},m_{j+1}}^{-1}\left(\mathcal{L}_{m_{j+1}}\right)^*F^{(m_{j+1})}\right), \tag{2.65}$$

where ω_j are defined in (2.40). Using the identities $\mathcal{L}f = F$, $\mathcal{L}_m f = F^{(m)}$,

$$\begin{gathered}
T_a^{-1}T = T_a^{-1}(T + aI - aI) = I - aT_a^{-1}, \\
T_{a,m}^{-1}T^{(m)} = T_{a,m}^{-1}\left(T^{(m)} + aI - aI\right) = I - aT_{a,m}^{-1}, \\
T_{a,m}^{-1} - T_a^{-1} = T_{a,m}^{-1}(T - T^{(m)})T_a^{-1},
\end{gathered} \tag{2.66}$$

one gets

$$\begin{aligned}
U_n &= \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}\left(T_{a_{j+1},m_{j+1}}^{-1} - T_{a_{j+1}}^{-1}\right)f \\
&= \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}T_{a_{j+1},m_{j+1}}^{-1}\left(T - T^{(m_{j+1})}\right)T_{a_{j+1}}^{-1}f.
\end{aligned} \tag{2.67}$$

This together with the rule (2.61), estimate (2.28), and Lemma 2.8 yields

$$\begin{aligned}
\|U_n\| &\le \sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}\left\|T_{a_{j+1},m_{j+1}}^{-1}\right\|\left\|T - T^{(m_{j+1})}\right\|\left\|T_{a_{j+1}}^{-1}f\right\| \\
&\le \frac{(2bd)^5}{540\sqrt{10}a_0\kappa^4}\sum_{j=0}^{n-1}\omega_j^{(n)} a_{j+1}\left\|T_{a_{j+1}}^{-1}f\right\|.
\end{aligned} \tag{2.68}$$

Applying Lemmas 2.5 and 2.7 with $g(a) = a\|T_a^{-1}f\|$, we obtain $\lim_{n\to\infty}\|U_n\| = 0$.

Lemma 2.10 is proved. □

2.1. Noisy Data

When the data $F(p)$ are noisy, the approximate solution (2.23) is written as

$$f_m^\delta(t) = \sum_{j=0}^{m} w_j^{(m)} c_j^{(m,\delta)} e^{-p_j t} = T_{a,m}^{-1}\mathcal{L}_m^* F_\delta^{(m)}, \tag{2.69}$$

where the coefficients $c_j^{(m,\delta)}$ are obtained by solving the following linear algebraic system:

$$Q_{a,m}c^{(m,\delta)} = F_\delta^{(m)}, \tag{2.70}$$

$Q_{a,m}$ is defined in (2.30),

$$c^{(m,\delta)} := \begin{pmatrix} c_0^{(m,\delta)} \\ c_1^{(m,\delta)} \\ \vdots \\ c_m^{(m,\delta)} \end{pmatrix}, \qquad F_\delta^{(m)} := \begin{pmatrix} F_\delta(p_0) \\ F_\delta(p_1) \\ \vdots \\ F_\delta(p_m) \end{pmatrix}, \tag{2.71}$$

$w_j^{(m)}$ are defined in (2.5), and p_j are defined in (2.3).

To get the approximation solution of the function $f(t)$ with the noisy data $F_\delta(p)$, we consider the following iterative scheme:

$$u_{n,m_n}^\delta = q u_{n-1,m_{n-1}}^\delta + (1-q) T_{a_n,m_n}^{-1} \mathcal{L}_{m_n}^* F_\delta^{(m_n)}, \quad u_{0,m_0}^\delta = 0, \tag{2.72}$$

where $T_{a,m}$ is defined in (2.30), a_n are defined in (2.35), $q \in (0,1)$, $F_\delta^{(m)}$ is defined in (2.71), and m_n are chosen by the rule (2.61). Let us assume that

$$F_\delta(p_j) = F(p_j) + \delta_j, \quad 0 < |\delta_j| \le \delta, \quad j = 0,1,2,\ldots,m, \tag{2.73}$$

where δ_j are random quantities generated from some statistical distributions, for example, the uniform distribution on the interval $[-\delta,\delta]$, and δ is the noise level of the data $F(p)$. It follows from assumption (2.73), definition (2.5), Lemma 2.1, and the inner product (2.4) that

$$\left\| F_\delta^{(m)} - F^{(m)} \right\|_{W^m}^2 = \sum_{j=0}^{m} w_j^{(m)} \delta_j^2 \le \delta^2 \sum_{j=0}^{m} w_j^{(m)} = \delta^2 d. \tag{2.74}$$

Lemma 2.11. *Let u_{n,m_n} and u_{n,m_n}^δ be defined in* (2.52) *and* (2.72), *respectively. Then*

$$\left\| u_{n,m_n} - u_{n,m_n}^\delta \right\| \le \frac{\sqrt{d}\delta}{2\sqrt{a_n}}(1-q^n), \quad q \in (0,1), \tag{2.75}$$

where a_n are defined in (2.35).

Proof. Let $U_n^\delta := u_{n,m_n} - u_{n,m_n}^\delta$. Then, from definitions (2.52) and (2.72),

$$U_n^\delta = q U_{n-1}^\delta + (1-q) T_{a_n,m_n}^{-1} \mathcal{L}_{m_n}^* \left(F^{(m_n)} - F_\delta^{(m_n)} \right), \quad U_0^\delta = 0. \tag{2.76}$$

By induction we obtain

$$U_n^\delta = \sum_{j=0}^{n-1} \omega_j^{(n)} T_{a_{j+1},m_{j+1}}^{-1} \left(\mathcal{L}_{m_{j+1}} \right)^* \left(F^{(m_{j+1})} - F_\delta^{(m_{j+1})} \right), \tag{2.77}$$

where $\omega_j^{(n)}$ are defined in (2.40). Using estimates (2.74) and inequality (2.29), one gets

$$\left\|U_n^\delta\right\| \le \sqrt{d}\sum_{j=0}^{n-1}\omega_j^{(n)}\frac{\delta}{2\sqrt{a_{j+1}}} \le \frac{\sqrt{d}\delta}{2\sqrt{a_n}}\sum_{j=0}^{m}\omega_j^{(n)} = \frac{\sqrt{d}\delta}{2\sqrt{a_n}}(1-q^n), \tag{2.78}$$

where ω_j are defined in (2.40).

Lemma 2.11 is proved. □

Theorem 2.12. *Suppose that conditions of Lemma 2.10 hold, and n_δ satisfies the following conditions:*

$$\lim_{\delta\to 0} n_\delta = \infty, \qquad \lim_{\delta\to 0}\frac{\delta}{\sqrt{a_{n_\delta}}} = 0. \tag{2.79}$$

Then

$$\lim_{\delta\to 0}\left\|f - u^\delta_{n_\delta,m_{n_\delta}}\right\| = 0. \tag{2.80}$$

Proof. Consider the estimate

$$\left\|f - u^\delta_{n_\delta,m_{n_\delta}}\right\| \le \left\|f - u_{n_\delta,m_{n_\delta}}\right\| + \left\|u_{n_\delta,m_{n_\delta}} - u^\delta_{n_\delta,m_{n_\delta}}\right\|. \tag{2.81}$$

This together with Lemma 2.11 yields

$$\left\|f - u^\delta_{n_\delta,m_{n_\delta}}\right\| \le \left\|f - u_{n_\delta,m_{n_\delta}}\right\| + \frac{\sqrt{d}\delta}{2\sqrt{a_{n_\delta}}}(1-q^n). \tag{2.82}$$

Applying relations (2.79) in estimate (2.82), one gets relation (2.80).

Theorem 2.12 is proved. □

In the following subsection we propose a stopping rule which implies relations (2.79).

2.2. Stopping Rule

In this subsection a stopping rule which yields relations (2.79) in Theorem 2.12 is given. We propose the stopping rule

$$G_{n_\delta,m_{n_\delta}} \le C\delta^\varepsilon < G_{n,m_n}, \quad 1 \le n < n_\delta,\ C > \sqrt{d},\ \varepsilon \in (0,1), \tag{2.83}$$

where

$$G_{n,m_n} = qG_{n-1,m_{n-1}} + (1-q)\left\|\mathcal{L}_{m_n} z^{(m_n,\delta)} - F_\delta^{(m_n)}\right\|_{W^{m_n}}, \quad G_{0,m_0} = 0, \tag{2.84}$$

$\|\cdot\|_{W^m}$ is defined in (2.4),

$$z^{(m,\delta)} := \sum_{j=0}^{m} c_j^{(m,\delta)} w_j^{(m)} e^{-p_j t}, \tag{2.85}$$

$w_j^{(m)}$ and p_j are defined in (2.5) and (2.3), respectively, and $c_j^{(m,\delta)}$ are obtained by solving linear algebraic system (2.70).

We observe that

$$\begin{aligned}
\mathcal{L}_{m_n} z^{(m_n,\delta)} - F_\delta^{(m_n)} &= Q^{(m_n)} c^{(m_n,\delta)} - F_\delta^{(m_n)} \\
&= Q^{(m_n)} \left(a_n I + Q^{(m_n)}\right)^{-1} F_\delta^{(m_n)} - F_\delta^{(m_n)} \\
&= \left(Q^{(m_n)} + a_n I - a_n I\right)\left(a_n I + Q^{(m_n)}\right)^{-1} F_\delta^{(m_n)} - F_\delta^{(m_n)} \\
&= -a_n \left(a_n I + Q^{(m_n)}\right)^{-1} F_\delta^{(m_n)} = -a_n c^{(m_n,\delta)}.
\end{aligned} \tag{2.86}$$

Thus, the sequence (2.84) can be written in the following form:

$$G_{n,m_n} = qG_{n-1,m_{n-1}} + (1-q) a_n \left\| c^{(m_n,\delta)} \right\|_{W^{m_n}}, \quad G_{0,m_0} = 0, \tag{2.87}$$

where $\|\cdot\|_{W^m}$ is defined in (2.4), and $c^{(m,\delta)}$ solves the linear algebraic system (2.70).

It follows from estimates (2.74), (2.26), and (2.27) that

$$\begin{aligned}
a_n \left\| c^{(m_n,\delta)} \right\|_{W^{m_n}} &= a_n \left\| \left(a_n I + Q^{(m_n)}\right)^{-1} F_\delta^{(m_n)} \right\|_{W^{m_n}} \\
&\leq a_n \left\| \left(a_n I + Q^{(m_n)}\right)^{-1} \left(F_\delta^{(m_n)} - F^{(m_n)}\right) \right\|_{W^{m_n}} \\
&\quad + a_n \left\| \left(a_n I + Q^{(m_n)}\right)^{-1} F^{(m_n)} \right\|_{W^{m_n}} \\
&\leq \left\| F_\delta^{(m_n)} - F^{(m_n)} \right\|_{W^{m_n}} \\
&\quad + a_n \left\| \left(a_n I + Q^{(m_n)}\right)^{-1} \mathcal{L}_{m_n} f \right\|_{W^{m_n}} \\
&\leq \delta\sqrt{d} + \sqrt{a_n} \|f\|_{X_{0,b}}.
\end{aligned} \tag{2.88}$$

This together with (2.87) yields

$$G_{n,m_n} \leq qG_{n-1,m_{n-1}} + (1-q)\left(\delta\sqrt{d} + \sqrt{a_n}\|f\|_{X_{0,b}}\right), \tag{2.89}$$

or

$$G_{n,m_n} - \delta\sqrt{d} \leq q\left(G_{n-1,m_{n-1}} - \delta\sqrt{d}\right) + (1-q)\sqrt{a_n}\|f\|_{X_{0,b}}. \tag{2.90}$$

Lemma 2.13. *The sequence* (2.87) *satisfies the following estimate:*

$$G_{n,m_n} - \delta\sqrt{d} \leq \frac{(1-q)\sqrt{a_n}\|f\|_{X_{0,b}}}{1-\sqrt{q}}, \tag{2.91}$$

where a_n are defined in (2.35).

Proof. Define

$$\begin{gathered} \Psi_n := G_{n,m_n} - \delta\sqrt{d}\ , \\ \psi_n := (1-q)\sqrt{a_n}\|f\|_{X_{0,b}}. \end{gathered} \tag{2.92}$$

Then estimate (2.90) can be rewritten as

$$\Psi_n \leq q\Psi_{n-1} + \sqrt{q}\psi_{n-1}, \tag{2.93}$$

where the relation $a_n = qa_{n-1}$ was used. Let us prove estimate (2.91) by induction. For $n = 0$ we get

$$\Psi_0 = -\delta\sqrt{d} \leq \frac{(1-q)\sqrt{a_0}\|f\|_{X_{0,b}}}{1-\sqrt{q}}. \tag{2.94}$$

Suppose that estimate (2.91) is true for $0 \leq n \leq k$. Then

$$\begin{aligned} \Psi_{k+1} &\leq q\Psi_k + \sqrt{q}\psi_k \leq \frac{q}{1-\sqrt{q}}\psi_k + \sqrt{q}\psi_k \\ &= \frac{\sqrt{q}}{1-\sqrt{q}}\psi_k = \frac{\sqrt{q}}{1-\sqrt{q}}\frac{\psi_k}{\psi_{k+1}}\psi_{k+1} \\ &= \frac{\sqrt{q}}{1-\sqrt{q}}\frac{\sqrt{a_k}}{\sqrt{a_{k+1}}}\psi_{k+1} = \frac{1}{1-\sqrt{q}}\psi_{k+1}, \end{aligned} \tag{2.95}$$

where the relation $a_{k+1} = qa_k$ was used.

Lemma 2.13 is proved. □

Lemma 2.14. *Suppose*

$$G_{1,m_1} > \delta\sqrt{d}, \tag{2.96}$$

where G_{n,m_n} are defined in (2.87)*. Then there exists a unique integer n_δ, satisfying the stopping rule* (2.83) *with $C > \sqrt{d}$.*

Proof. From Lemma 2.13 we get the estimate

$$G_{n,m_n} \leq \delta\sqrt{d} + \frac{(1-q)\sqrt{a_n}\|f\|_{X_{0,b}}}{1-\sqrt{q}}, \tag{2.97}$$

where a_n are defined in (2.35). Therefore,

$$\limsup_{n\to\infty} G_{n,m_n} \leq \delta\sqrt{d}, \tag{2.98}$$

where the relation $\lim_{n\to\infty} a_n = 0$ was used. This together with condition (2.96) yields the existence of the integer n_δ. The uniqueness of the integer n_δ follows from its definition.

Lemma 2.14 is proved. □

Lemma 2.15. *Suppose that conditions of Lemma 2.14 hold and n_δ is chosen by the rule* (2.83)*, then*

$$\lim_{\delta\to 0}\frac{\delta}{\sqrt{a_{n_\delta}}} = 0. \tag{2.99}$$

Proof. From the stopping rule (2.83) and estimate (2.97) we get

$$C\delta^{\varepsilon} \leq G_{n_\delta-1,m_{n_\delta-1}} \leq \delta\sqrt{d} + \frac{(1-q)\sqrt{a_{n_\delta-1}}\|f\|_{X_{0,b}}}{1-\sqrt{q}}, \tag{2.100}$$

where $C > \sqrt{d}, \varepsilon \in (0,1)$. This implies

$$\frac{\delta\left(C\delta^{\varepsilon-1}-\sqrt{d}\right)}{\sqrt{a_{n_\delta-1}}} \leq \frac{(1-q)\|f\|_{X_{0,b}}}{1-\sqrt{q}}, \tag{2.101}$$

so, for $\varepsilon \in (0,1)$, and $a_{n_\delta} = q a_{n_\delta-1}$, one gets

$$\lim_{\delta\to 0}\frac{\delta}{\sqrt{a_{n_\delta}}} = \lim_{\delta\to 0}\frac{\delta}{\sqrt{q}\sqrt{a_{n_\delta-1}}} \leq \lim_{\delta\to 0}\frac{(1-q)\delta^{1-\varepsilon}\|f\|_{X_{0,b}}}{(\sqrt{q}-q)\left(C-\delta^{1-\varepsilon}\sqrt{d}\right)} = 0. \tag{2.102}$$

Lemma 2.15 is proved. □

Lemma 2.16. *Consider the stopping rule* (2.83)*, where the parameters m_n are chosen by rule* (2.61)*. If n_δ is chosen by the rule* (2.83) *then*

$$\lim_{\delta\to 0} n_\delta = \infty. \tag{2.103}$$

Proof. From the stopping rule (2.83) with the sequence G_n defined in (2.87) one gets

$$\begin{aligned} qC\delta^{\varepsilon} + (1-q)a_{n_\delta}\left\|c^{(m_{n_\delta},\delta)}\right\|_{W^{m_{n_\delta}}} &\leq qG_{n_\delta - 1, m_{n_{\delta-1}}} + (1-q)a_{n_\delta}\left\|c^{(m_{n_\delta},\delta)}\right\|_{W^{m_{n_\delta}}} \\ &= G_{n_\delta, m_{n_\delta}} < C\delta^{\varepsilon}, \end{aligned} \tag{2.104}$$

where $c^{(m,\delta)}$ is obtained by solving linear algebraic system (2.70). This implies

$$0 < a_{n_\delta}\left\|c^{(m_{n_\delta},\delta)}\right\|_{W^{m_{n_\delta}}} \leq C\delta^{\varepsilon}. \tag{2.105}$$

Thus,

$$\lim_{\delta\to 0} a_{n_\delta}\left\|c^{(m_{n_\delta},\delta)}\right\|_{W^{m_{n_\delta}}} = 0. \tag{2.106}$$

If $F^{(m)} \neq 0$, then there exists a $\lambda_0^{(m)} > 0$ such that

$$E^{(m)}_{\lambda_0^{(m)}} F^{(m)} \neq 0, \quad \left\langle E^{(m)}_{\lambda_0} F^{(m)}, F^{(m)}\right\rangle_{W^m} := \xi^{(m)} > 0, \tag{2.107}$$

where $E_s^{(m)}$ is the resolution of the identity corresponding to the operator $Q^{(m)} := \mathcal{L}_m\mathcal{L}_m^*$. Let

$$h_m(\delta,\alpha) := \alpha^2\left\|Q^{-1}_{m,\alpha}F^{(m)}_\delta\right\|^2_{W^m}, \quad Q_{m,a} := aI + Q^{(m)}. \tag{2.108}$$

For a fixed number $a > 0$ we obtain

$$\begin{aligned} h_m(\delta,a) &= a^2\left\|Q^{-1}_{m,a}F^{(m)}_\delta\right\|^2_{W^m} \\ &= \int_0^\infty \frac{a^2}{(a+s)^2} d\left\langle E_s^{(m)}F_\delta^{(m)}, F_\delta^{(m)}\right\rangle_{W^m} \\ &\geq \int_0^{\lambda_0^{(m)}} \frac{a^2}{(a+s)^2} d\left\langle E_s^{(m)}F_\delta^{(m)}, F_\delta^{(m)}\right\rangle_{W^m} \\ &\geq \frac{a^2}{(a+\lambda_0)^2}\int_0^{\lambda_0^{(m)}} d\left\langle E_s^{(m)}F_\delta^{(m)}, F_\delta^{(m)}\right\rangle_{W^m} \\ &= \frac{a^2\left\|E^{(m)}_{\lambda_0^{(m)}}F_\delta^{(m)}\right\|^2_{W^m}}{\left(a+\lambda_0^{(m)}\right)^2}. \end{aligned} \tag{2.109}$$

Since $E_{\lambda_0}^{(m)}$ is a continuous operator, and $\|F^{(m)} - F_\delta^{(m)}\|_{W^m} < \sqrt{d}\delta$, it follows from (2.107) that

$$\lim_{\delta\to 0}\left\langle E_{\lambda_0}^{(m)} F_\delta^{(m)}, F_\delta^{(m)}\right\rangle_{W^m} = \left\langle E_{\lambda_0}^{(m)} F^{(m)}, F^{(m)}\right\rangle_{W^m} > 0. \tag{2.110}$$

Therefore, for the fixed number $a > 0$ we get

$$h_m(\delta, a) \geq c_2 > 0 \tag{2.111}$$

for all sufficiently small $\delta > 0$, where c_2 is a constant which does not depend on δ. Suppose $\lim_{\delta\to 0} a_{n_\delta} \neq 0$. Then there exists a subsequence $\delta_j \to 0$ as $j \to \infty$, such that

$$\begin{gathered} a_{n_{\delta_j}} \geq c_1 > 0, \\ 0 < m_{n_{\delta_j}} = \left[\left[\kappa\left(\frac{a_0}{a_{n_{\delta_j}}}\right)^{1/4}\right]\right] \leq \left[\left[\kappa\left(\frac{a_0}{c_1}\right)^{1/4}\right]\right] := c_3 < \infty, \quad \kappa, a_0 > 0, \end{gathered} \tag{2.112}$$

where rule (2.61) was used to obtain the parameters $m_{n_{\delta_j}}$. This together with (2.107) and (2.111) yields

$$\begin{aligned} \lim_{j\to\infty} h_{m_{n_{\delta_j}}}\left(\delta_j, a_{n_{\delta_j}}\right) &\geq \lim_{j\to\infty} \frac{a_{n_{\delta_j}}^2 \left\| E_{\lambda_0^{(m_{n_{\delta_j}})}}^{(m_{n_{\delta_j}})} F_{\delta_j}^{(m_{n_{\delta_j}})} \right\|_{W^{m_{n_{\delta_j}}}}^2}{\left(a_{n_{\delta_j}} + \lambda_0^{(m_{n_{\delta_j}})}\right)^2} \\ &\geq \liminf_{j\to\infty} \frac{c_1^2 \left\| E_{\lambda_0^{(m_{n_{\delta_j}})}}^{(m_{n_{\delta_j}})} F^{(m_{n_{\delta_j}})} \right\|_{W^{m_{n_{\delta_j}}}}^2}{\left(c_1 + \lambda_0^{(m_{n_{\delta_j}})}\right)^2} > 0. \end{aligned} \tag{2.113}$$

This contradicts relation (2.106). Thus, $\lim_{\delta\to 0} a_{n_\delta} = \lim_{\delta\to 0} a_0 q^{n_\delta} = 0$, that is, $\lim_{\delta\to 0} n_\delta = \infty$.

Lemma 2.16 is proved. □

It follows from Lemmas 2.15 and 2.16 that the stopping rule (2.83) yields the relations (2.79). We have proved the following theorem.

Theorem 2.17. *Suppose all the assumptions of Theorem 2.12 hold, m_n are chosen by rule* (2.61)*, n_δ is chosen by rule* (2.83)*, and $G_{1,m_1} > C\delta$, where G_{n,m_n} are defined in* (2.87)*, then*

$$\lim_{\delta\to 0}\left\| f - u_{n_\delta, m_{n_\delta}}^\delta \right\| = 0. \tag{2.114}$$

2.3. The Algorithm

Let us formulate the algorithm for obtaining the approximate solution f_m^δ:

(1) The data $F_\delta(p)$ on the interval $[0,d]$, $d>0$, the support of the function $f(t)$, and the noise level δ.

(2) Initialization: choose the parameters $\kappa>0$, $a_0>0$, $q\in(0,1)$, $\varepsilon\in(0,1)$, $C>\sqrt{d}$, and set $u^{\delta}_{0,m_0}=0$, $G_0=0$, $n=1$.

(3) Iterate, starting with $n=1$, and stop when condition (2.120) (see below) holds,

(a) $a_n=a_0q^n$,

(b) choose m_n by the rule (2.61),

(c) construct the vector $F^{(m_n)}_\delta$:

$$\left(F^{(m_n)}_\delta\right)_l=F_\delta(p_l),\quad p_l=lh,\ h=\frac{d}{m_n},\ l=0,1,\ldots,m, \tag{2.115}$$

(d) construct the matrices H_{m_n} and D_{m_n}:

$$(H_{m_n})_{ij}:=\int_0^b e^{-(p_i+p_j)t}dt=\frac{1-e^{-b(p_i+p_j)}}{p_i+p_j},\quad i,j=1,2,3,\ldots,m_n,$$
$$(D_{m_n})_{ij}=\begin{cases}w^{(m_n)}_i, & i=j,\\ 0, & \text{otherwise},\end{cases} \tag{2.116}$$

where $w^{(m)}_j$ are defined in (2.5),

(e) solve the following linear algebraic system:

$$(a_nI+H_{m_n}D_{m_n})c^{(m_n,\delta)}=F^{(m_n)}_\delta, \tag{2.117}$$

where $(c^{(m_n,\delta)})_i=c^{(m_n,\delta)}_i$,

(f) update the coefficient $c^{(m_n,\delta)}_j$ of the approximate solution $u^{\delta}_{n,m_n}(t)$ defined in (2.69) by the iterative formula:

$$u^{\delta}_{n,m_n}(t)=qu^{\delta}_{n-1,m_{n-1}}(t)+(1-q)\sum_{j=1}^{m_n}c^{(m_n,\delta)}w^{(m_n)}_j e^{-p_jt}, \tag{2.118}$$

where

$$u^{\delta}_{0,m_0}(t)=0. \tag{2.119}$$

Stop when for the first time the inequality

$$G_{n,m_n}=qG_{n-1,m_{n-1}}+a_n\left\|c^{(m_n,\delta)}\right\|_{W^{m_n}}\le C\delta^{\varepsilon} \tag{2.120}$$

holds, and get the approximation $f^{\delta}(t)=u^{\delta}_{n_\delta,m_{n_\delta}}(t)$ of the function $f(t)$ by formula (2.118).

3. Numerical Experiments

3.1. The Parameters κ, a_0, d

From definition (2.35) and the rule (2.61) we conclude that $m_n \to \infty$ as $a_n \to 0$. Therefore, one needs to control the value of the parameter m_n so that it will not grow too fast as a_n decreases. The role of the parameter κ in (2.61) is to control the value of the parameter m_n so that the value of the parameter m_n will not be too large. Since for sufficiently small noise level δ, namely, $\delta \in (10^{-16}, 10^{-6}]$, the regularization parameter a_{n_δ}, obtained by the stopping rule (2.83), is at most $O(10^{-9})$, we suggest to choose κ in the interval $(0, 1]$. For the noise level $\delta \in (10^{-6}, 10^{-2}]$ one can choose $\kappa \in (1, 3]$. To reduce the number of iterations we suggest to choose the geometric sequence $a_n = a_0\delta^{\alpha n}$, where $a_0 \in [0.1, 0.2]$ and $\alpha \in [0.5, 0.9]$. One may assume without loss of generality that $b = 1$, because a scaling transformation reduces the integral over $(0, b)$ to the integral over $(0, 1)$. We have assumed that the data $F(p)$ are defined on the interval $J := [0, d]$. In the case the interval $J = [d_1, d]$, $0 < d_1 < d$, the constant d in estimates (2.59), (2.74), (2.75), (2.78), (2.90), (2.91), and (2.97) are replaced with the constant $d - d_1$. If $b = 1$, that is, $f(t) = 0$ for $t > 1$, then one has to take d not too large. Indeed, if $f(t) = 0$ for $t > 1$, then an integration by parts yields $F(p) = [f(0) - e^{-p}f(1)]/p + O(1/p^2)$, $p \to \infty$. If the data are noisy, and the noise level is δ, then the data becomes indistinguishable from noise for $p = O(1/\delta)$. Therefore it is useless to keep the data $F_\delta(p)$ for $d > O(1/\delta)$. In practice one may get a satisfactory accuracy of inversion by the method, proposed in this paper, when one uses the data with $d \in [1, 20]$ when $\delta \leq 10^{-2}$. In all the numerical examples we have used $d = 5$. Given the interval $[0, d]$, the proposed method generates automatically the discrete data $F_\delta(p_j)$, $j = 0, 1, 2, \ldots, m$, over the interval $[0, d]$ which are needed to get the approximation of the function $f(t)$.

3.2. Experiments

To test the proposed method we consider some examples proposed in [5–7, 9, 12, 13, 15, 16, 19, 25]. To illustrate the numerical stability of the proposed method with respect to the noise, we use the noisy data $F_\delta(p)$ with various noise levels $\delta = 10^{-2}$, $\delta = 10^{-4}$, and $\delta = 10^{-6}$. The random quantities δ_j in (2.73) are obtained from the uniform probability density function over the interval $[-\delta, \delta]$. In Examples 3.1–3.12 we choose the value of the parameters as follows: $a_n = 0.1q^n$, $q = \delta^{1/2}$, and $d = 5$. The parameter $\kappa = 1$ is used for the noise levels $\delta = 10^{-2}$ and $\delta = 10^{-4}$. When $\delta = 10^{-6}$ we choose $\kappa = 0.3$ so that the value of the parameters m_n are not very large, namely, $m_n \leq 300$. Therefore, the computation time for solving linear algebraic system (2.117) can be reduced significantly. We assume that the support of the function $f(t)$ is in the interval $[0, b]$ with $b = 10$. In the stopping rule (2.83) the following parameters are used: $C = \sqrt{d} + 0.01$, $\varepsilon = 0.99$. In Example 3.13 the function $f(t) = e^{-t}$ is used to test the applicability of the proposed method to functions without compact support. The results are given in Table 13 and Figure 13.

For a comparison with the exact solutions we use the mean absolute error

$$\text{MAE} := \left[\frac{\sum_{j=1}^{100} \left(f(t_i) - f^{\delta}_{m_{n_\delta}}(t_i) \right)^2}{100} \right]^{1/2}, \quad t_j = 0.01 + 0.1(j-1),\ j = 1, \ldots, 100, \tag{3.1}$$

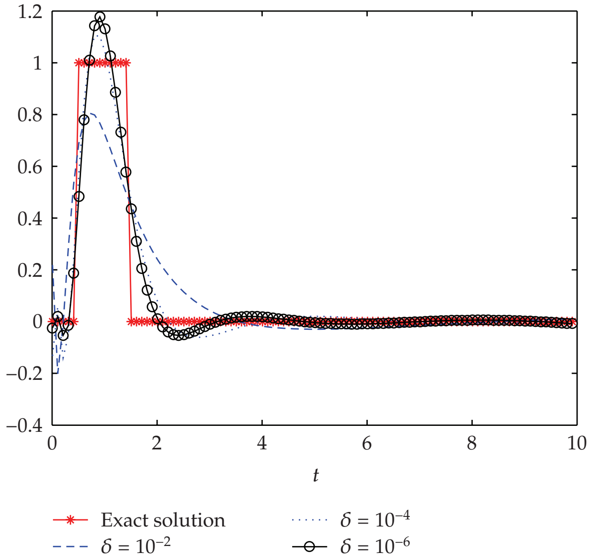

Figure 1: Example 3.1: the stability of the approximate solution.

Table 1: Example 3.1.

δ	MAE	m_{n_δ}	Iter.	CPU time (second)	a_{n_δ}
1.00×10^{-2}	9.62×10^{-2}	30	3	3.13×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	5.99×10^{-2}	32	4	6.25×10^{-2}	2.00×10^{-7}
1.00×10^{-6}	4.74×10^{-2}	54	5	3.28×10^{-1}	2.00×10^{-10}

where $f(t)$ is the exact solution and $f^{\delta}_{m_{n_\delta}}(t)$ is the approximate solution. The computation time (CPU time) for obtaining the approximation of $f(t)$, the number of iterations (Iter.), and the parameters m_{n_δ} and a_{n_δ} generated by the proposed method is given in each experiment (see Tables 1–12). All the calculations are done in double precision generated by MATLAB.

Example 3.1 (see [15]). We have

$$f_1(t) = \begin{cases} 1, & \dfrac{1}{2} \le t \le \dfrac{3}{2}, \\ 0, & \text{otherwise}, \end{cases}$$
$$F_1(p) = \begin{cases} 1, & p = 0, \\ \dfrac{e^{-p/2} - e^{-3p/2}}{p}, & p > 0. \end{cases} \tag{3.2}$$

The reconstruction of the exact solution for different values of the noise level δ is shown in Figure 1. When the noise level $\delta = 10^{-6}$, our result is comparable with the double precision results shown in [15]. The proposed method is stable with respect to the noise δ as shown in Table 1.

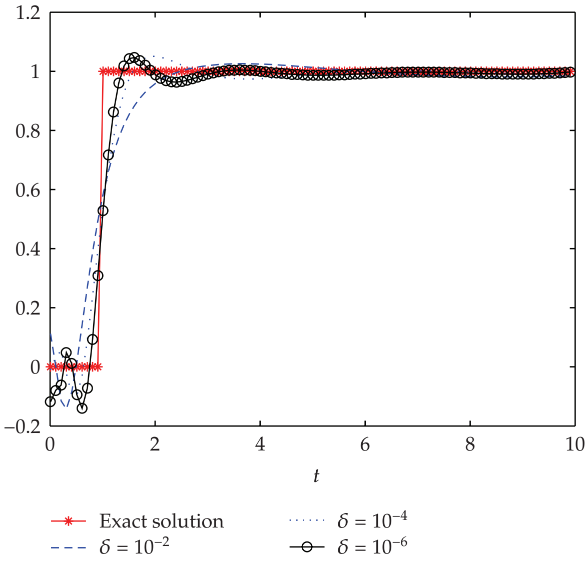

Figure 2: Example 3.2: the stability of the approximate solution.

Table 2: Example 3.2.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	1.09×10^{-1}	30	2	3.13×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	8.47×10^{-2}	32	3	6.25×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	7.41×10^{-2}	54	5	4.38×10^{-1}	2.00×10^{-12}

Example 3.2 (see [13, 15]). We have

$$f_2(t) = \begin{cases} \dfrac{1}{2}, & t = 1, \\ 1, & 1 < t < 10, \\ 0, & \text{elsewhere}, \end{cases} \qquad F_2(p) = \begin{cases} 9, & p = 0, \\ \dfrac{e^{-p} - e^{-10p}}{p}, & p > 0. \end{cases} \tag{3.3}$$

The reconstruction of the function $f_2(t)$ is plotted in Figure 2. In [15] a high accuracy result is given by means of the multiple precision. But, as reported in [15], to get such high accuracy results, it takes 7 hours. From Table 2 and Figure 2 we can see that the proposed method yields stable solution with respect to the noise level δ. The reconstruction of the exact solution obtained by the proposed method is better than the reconstruction shown in [13]. The result is comparable with the double precision results given in [15]. For $\delta = 10^{-6}$ and $\kappa = 0.3$ the value of the parameter m_{n_δ} is bounded by the constant 54.

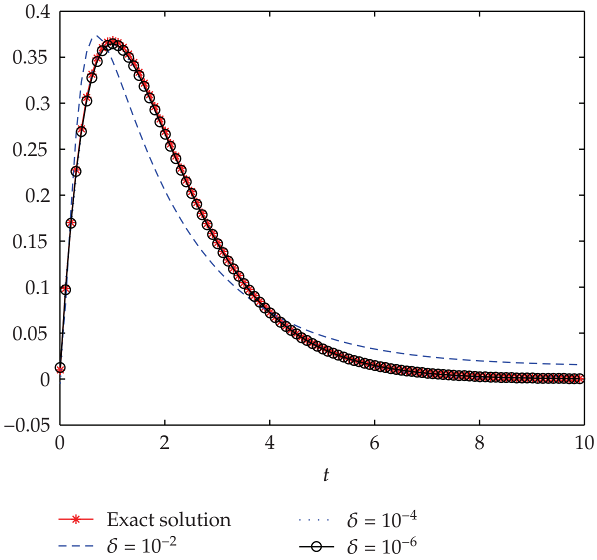

Figure 3: Example 3.3: the stability of the approximate solution.

Table 3: Example 3.3.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	2.42×10^{-2}	30	2	3.13×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	1.08×10^{-3}	30	3	3.13×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	4.02×10^{-4}	30	4	4.69×10^{-2}	2.00×10^{-9}

Example 3.3 (see [6, 12, 13, 16, 19]). We have

$$f_3(t) = \begin{cases} te^{-t}, & 0 \leq t < 10, \\ 0, & \text{otherwise}, \end{cases}$$

$$F_3(p) = \frac{1 - e^{-(p+1)10}}{(p+1)^2} - \frac{10e^{-(p+1)10}}{p+1}. \tag{3.4}$$

We get an excellent agreement between the approximate solution and the exact solution when the noise level $\delta = 10^{-4}$ and 10^{-6} as shown in Figure 3. The results obtained by the proposed method are better than the results given in [13]. The mean absolute error MAE decreases as the noise level decreases which shows the stability of the proposed method. Our results are more stable with respect to the noise δ than the results presented in [19]. The value of the parameter m_{n_δ} is bounded by the constant 30 when the noise level $\delta = 10^{-6}$ and $\kappa = 0.3$.

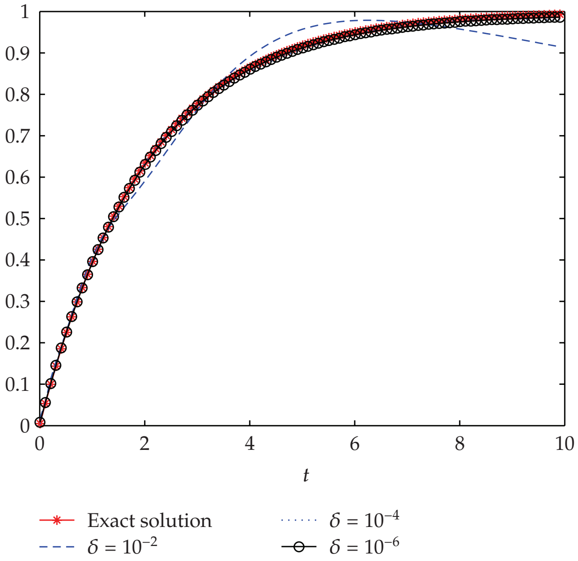

Figure 4: Example 3.4: the stability of the approximate solution.

Table 4: Example 3.4.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	1.59×10^{-2}	30	2	3.13×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	8.26×10^{-4}	30	3	9.400×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	1.24×10^{-4}	30	4	1.250×10^{-1}	2.00×10^{-9}

Example 3.4 (see [13, 15]). We have

$$f_4(t) = \begin{cases} 1 - e^{-0.5t}, & 0 \leq t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$
$$F_4(p) = \begin{cases} 8 + 2e^{-5}, & p = 0, \\ \dfrac{1 - e^{-10p}}{p} - \dfrac{1 - e^{-(p+1/2)10}}{p + 0.5}, & p > 0. \end{cases} \tag{3.5}$$

As in our Example 3.3 when the noise $\delta = 10^{-4}$ and 10^{-6} are used, we get a satisfactory agreement between the approximate solution and the exact solution. Table 4 gives the results of the stability of the proposed method with respect to the noise level δ. Moreover, the reconstruction of the function $f_4(t)$ obtained by the proposed method is better than the reconstruction of $f_4(t)$ shown in [13], and is comparable with the double precision reconstruction obtained in [15].

In this example when $\delta = 10^{-6}$ and $\kappa = 0.3$ the value of the parameter m_{n_δ} is bounded by the constant 109 as shown in Table 4.

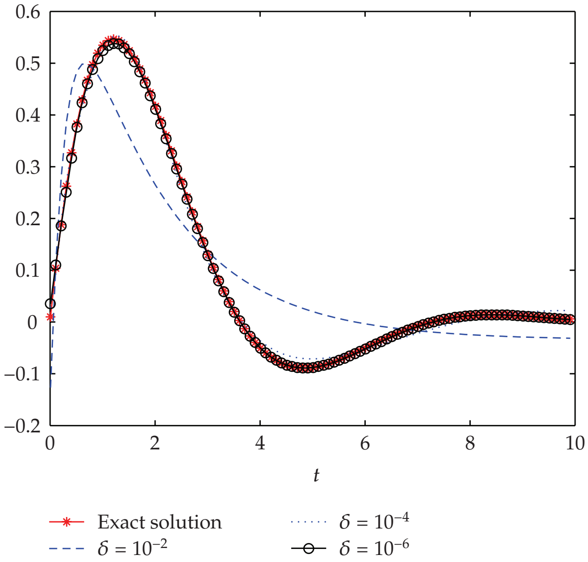

Figure 5: Example 3.5: the stability of the approximate solution.

Table 5: Example 3.5.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	4.26×10^{-2}	30	3	6.300×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	1.25×10^{-2}	30	3	9.38×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	1.86×10^{-3}	54	4	3.13×10^{-2}	2.00×10^{-9}

Example 3.5 (see [5, 7, 13]). We have

$$f_5(t) = \frac{2}{\sqrt{3}e^{-t/2}\sin\left(t\sqrt{3}/2\right)}$$

$$F_5(p) = \frac{1-\cos\left(10\sqrt{3}/2\right)e^{-10(p+0.5)}}{\left[(p+0.5)^2+3/4\right]} - \frac{2(p+0.5)e^{-10(p+0.5)}\sin\left(10\sqrt{3}/2\right)}{\sqrt{3}\left[(p+0.5)^2+3/4\right]}. \tag{3.6}$$

This is an example of the damped sine function. In [5, 7] the knowledge of the exact data $F(p)$ in the complex plane is required to get the approximate solution. Here we only use the knowledge of the discrete perturbed data $F_\delta(p_j)$, $j = 0, 1, 2, \dots, m$, and get a satisfactory result which is comparable with the results given in [5, 7] when the level noise $\delta = 10^{-6}$. The reconstruction of the exact solution $f_5(t)$ obtained by our method is better than this of the method given in [13]. Moreover, our method yields stable solution with respect to the noise level δ as shown in Figure 5 and Table 5. In this example when $\kappa = 0.3$ the value of the parameter m_{n_δ} is bounded by 54 for the noise level $\delta = 10^{-6}$ (see Table 5).

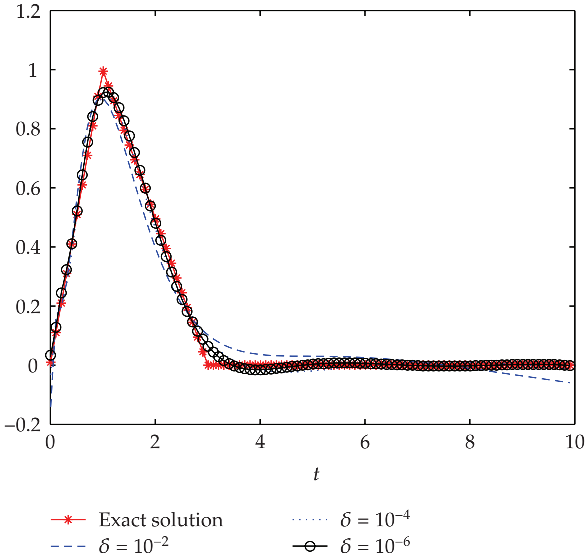

Figure 6: Example 3.6: the stability of the approximate solution.

Table 6: Example 3.6.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	4.19×10^{-2}	30	2	4.700×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	1.64×10^{-2}	32	3	9.38×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	1.22×10^{-2}	54	4	3.13×10^{-2}	2.00×10^{-9}

Example 3.6 (see [15]). We have

$$f_6(t) = \begin{cases} t, & 0 \le t < 1, \\ \dfrac{3}{2} - \dfrac{t}{2}, & 1 \le t < 3, \\ 0, & \text{elsewhere}, \end{cases}$$

$$F_6(p) = \begin{cases} \dfrac{3}{2}, & p = 0, \\ \dfrac{1 - e^{-p}(1+p)}{p^2} + \dfrac{e^{-3p} + e^{2p}(2p-1)}{2p^2}, & p > 0. \end{cases} \tag{3.7}$$

Example 3.6 represents a class of piecewise continuous functions. From Figure 6 the value of the exact solution at the points where the function is not differentiable cannot be well approximated for the given levels of noise by the proposed method. When the noise level $\delta = 10^{-6}$, our result is comparable with the results given in [15]. Table 6 reports the stability of the proposed method with respect to the noise δ. It is shown in Table 6 that the value of the parameter m generated by the proposed adaptive stopping rule is bounded by the constant 54 for the noise level $\delta = 10^{-6}$ and $\kappa = 0.3$ which gives a relatively small computation time.

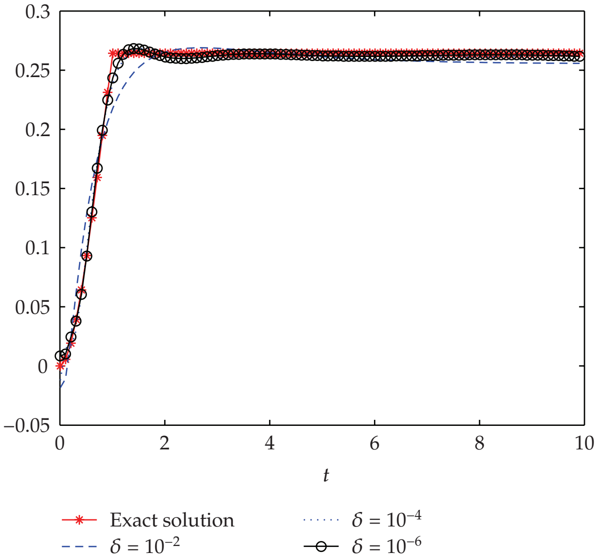

Figure 7: Example 3.7: the stability of the approximate solution.

Table 7: Example 3.7.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	1.52×10^{-2}	30	2	4.600×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	2.60×10^{-3}	30	3	9.38×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	2.02×10^{-3}	30	4	3.13×10^{-2}	2.00×10^{-9}

Example 3.7 (see [15]). We have

$$f_7(t) = \begin{cases} -te^{-t} - e^{-t} + 1, & 0 \le t < 1, \\ 1 - 2e^{-1}, & 1 \le t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$
$$F_7(p) = \begin{cases} \dfrac{3}{e} - 1 + 9\left(1 - \dfrac{2}{e}\right), & p = 0, \\ e^{-1-p(e^{1+p}-e(1+p)^2+p(3+2p)/p(p+1)^2)} + (e-2)e^{-1-p-10p((e^{10p}-e^{p})/p)}, & p > 0. \end{cases} \tag{3.8}$$

When the noise level $\delta = 10^{-4}$ and $\delta = 10^{-6}$, we get numerical results which are comparable with the double precision results given in [15]. Figure 7 and Table 7 show the stability of the proposed method for decreasing δ.

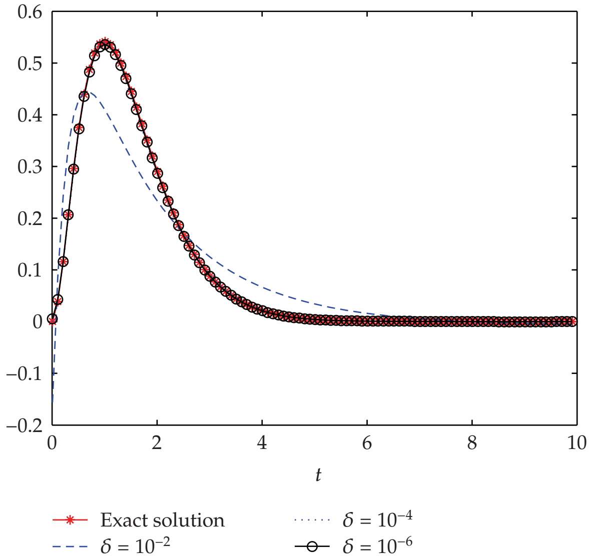

Figure 8: Example 3.8: the stability of the approximate solution.

Table 8: Example 3.8.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	2.74×10^{-2}	30	2	1.100×10^{-2}	2.00×10^{-3}
1.00×10^{-4}	3.58×10^{-3}	30	3	3.13×10^{-2}	2.00×10^{-6}
1.00×10^{-6}	5.04×10^{-4}	30	4	4.69×10^{-2}	2.00×10^{-9}

Example 3.8 (see [13, 25]). We have

$$f_8(t) = \begin{cases} 4t^2 e^{-2t}, & 0 \le t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$

$$F_8(p) = \frac{8 + 4e^{-10(2+p)}\left[-2 - 20(2+p) - 100(2-p)^2\right]}{(2+p)^3}. \tag{3.9}$$

The results of this example are similar to the results of Example 3.3. The exact solution can be well reconstructed by the approximate solution obtained by our method at the levels noise $\delta = 10^{-4}$ and $\delta = 10^{-6}$ (see Figure 8). Table 8 shows that the MAE decreases as the noise level decreases which shows the stability of the proposed method with respect to the noise. In all the levels of noise δ the computation time of the proposed method in obtaining the approximate solution is relatively small. We get better reconstruction results than the results shown in [13]. Our results are comparable with the results given in [25].

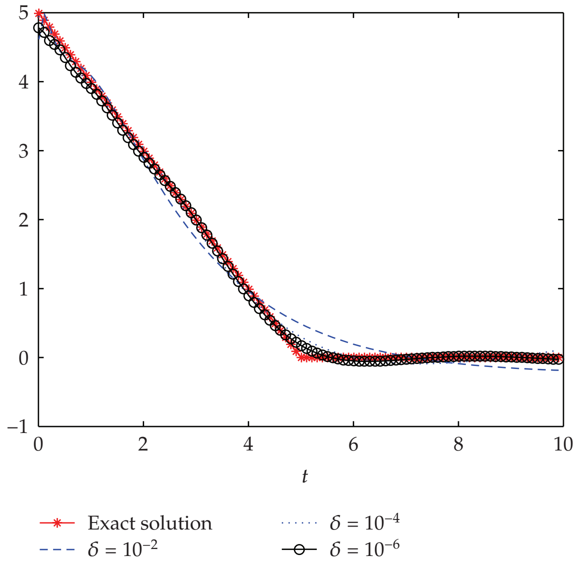

Figure 9: Example 3.9: the stability of the approximate solution.

Table 9: Example 3.9.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	2.07×10^{-1}	30	3	6.25×10^{-2}	2.00×10^{-6}
1.00×10^{-4}	7.14×10^{-2}	32	4	3.44×10^{-1}	2.00×10^{-9}
1.00×10^{-6}	2.56×10^{-2}	54	5	3.75×10^{-1}	2.00×10^{-12}

Example 3.9 (see [16]). We have

$$f_9(t) = \begin{cases} 5 - t, & 0 \leq t < 5, \\ 0, & \text{elsewhere}, \end{cases}$$
$$F_9(p) = \begin{cases} \dfrac{25}{2}, & p = 0, \\ \dfrac{e^{-5p} + 5p - 1}{p^2}, & p > 0. \end{cases} \tag{3.10}$$

As in Example 3.6 the error of the approximate solution at the point where the function is not differentiable dominates the error of the approximation. The reconstruction of the exact solution can be seen in Figure 9. The detailed results are presented in Table 9. When the double precision is used, we get comparable results with the results shown in [16].

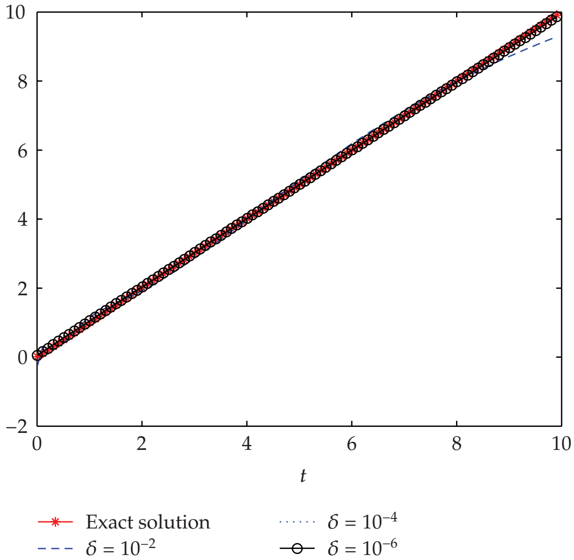

Figure 10: Example 3.10: the stability of the approximate solution.

Table 10: Example 3.10.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	2.09×10^{-1}	30	3	3.13×10^{-2}	2.00×10^{-6}
1.00×10^{-4}	1.35×10^{-2}	32	4	9.38×10^{-2}	2.00×10^{-9}
1.00×10^{-6}	3.00×10^{-3}	54	4	2.66×10^{-1}	2.00×10^{-9}

Example 3.10 (see [6]). We have

$$f_{10}(t) = \begin{cases} t, & 0 \le t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$
$$F_{10}(p) = \begin{cases} 50, & p = 0, \\ \dfrac{1 - e^{-10p}}{p^2} - \dfrac{10e^{-10p}}{p}, & p > 0. \end{cases} \tag{3.11}$$

Table 10 shows the stability of the solution obtained by our method with respect to the noise level δ. We get an excellent agreement between the exact solution and the approximate solution for all the noise levels δ as shown in Figure 10.

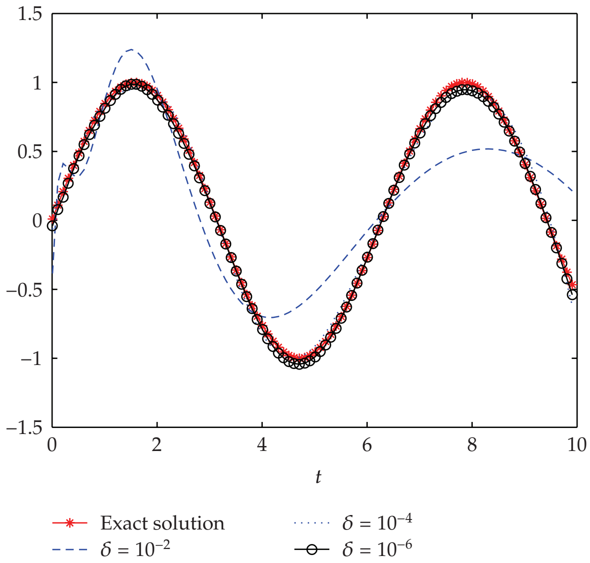

Figure 11: Example 3.11: the stability of the approximate solution.

Table 11: Example 3.11.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	2.47×10^{-1}	30	3	9.38×10^{-2}	2.00×10^{-6}
1.00×10^{-4}	4.91×10^{-2}	32	4	2.50×10^{-1}	2.00×10^{-9}
1.00×10^{-6}	2.46×10^{-2}	54	5	4.38×10^{-1}	2.00×10^{-12}

Example 3.11 (see [6, 9]). We have

$$f_{11}(t) = \begin{cases} \sin(t), & 0 \leq t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$
$$F_{11}(p) = \frac{1 - e^{-10p}(p\sin(10) + \cos(10))}{1 + p^2}. \tag{3.12}$$

Here the function $f_{11}(t)$ represents the class of periodic functions. It is mentioned in [9] that oscillating function can be found with acceptable accuracy only for relatively small values of t. In this example the best approximation is obtained when the noise level $\delta = 10^{-6}$ which is comparable with the results given in [6, 9]. The reconstruction of the function $f_{11}(t)$ for various levels of the noise δ are given in Figure 11. The stability of the proposed method with respect to the noise δ is shown in Table 11. In this example the parameter m_{n_δ} is bounded by the constant 54 when the noise level $\delta = 10^{-6}$ and $\kappa = 0.3$.

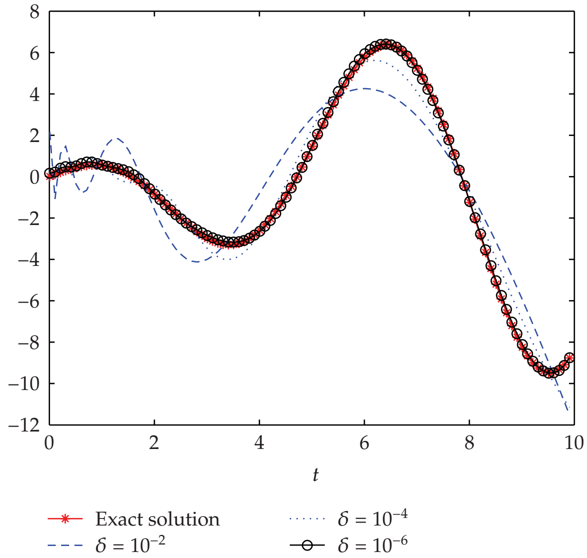

Figure 12: Example 3.12: the stability of the approximate solution.

Table 12: Example 3.12.

δ	MAE	m_{n_δ}	Iter.	CPU time (seconds)	a_{n_δ}
1.00×10^{-2}	1.37×10^{0}	96	3	9.38×10^{-2}	2.00×10^{-6}
1.00×10^{-4}	5.98×10^{-1}	100	4	2.66×10^{-1}	2.00×10^{-9}
1.00×10^{-6}	2.24×10^{-1}	300	5	3.44×10^{-1}	2.00×10^{-12}

Example 3.12 (see [6, 25]). We have

$$f_{12}(t) = \begin{cases} t\cos(t), & 0 \le t < 10, \\ 0, & \text{elsewhere}, \end{cases}$$

$$F_{12}(p) = \frac{(p^2 - 1) - e^{-10p}(-1 + p^2 + 10p + 10p^3)\cos(10)}{(1+p^2)^2} + \frac{e^{-10p}(2p + 10 + 10p^2)\sin(10)}{(1+p^2)^2}. \tag{3.13}$$

Here we take an increasing function which oscillates as the variable t increases over the interval $[0, 10)$. A poor approximation is obtained when the noise level $\delta = 10^{-2}$. Figure 12 shows that the exact solution can be approximated very well when the noise level $\delta = 10^{-6}$. The results of our method are comparable with these of the methods given in [6, 25]. The stability of our method with respect to the noise level is shown in Table 12.

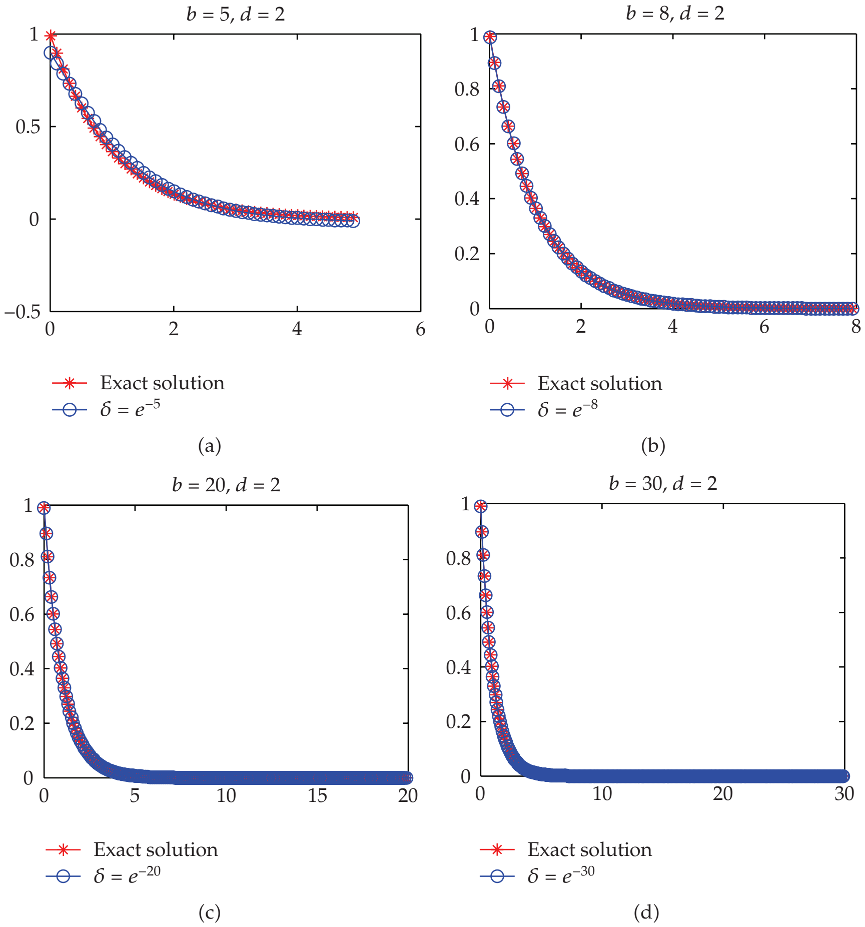

Figure 13: Example 3.13: the stability of the approximate solution.

Example 3.13. We have

$$f_{13}(t) = e^{-t}, \qquad F_{13}(p) = \frac{1}{1+p}. \tag{3.14}$$

Here the support of $f_{13}(t)$ is not compact. From the Laplace transform formula one gets

$$\begin{aligned} F_{13}(p) &= \int_0^\infty e^{-t}e^{-pt}dt = \int_0^b e^{-(1+p)t}dt + \int_b^\infty e^{-(1+p)t}dt \\ &= \int_0^b f_{13}(t)e^{-pt}dt + \frac{e^{-(1+p)b}}{1+p} := I_1 + I_2, \end{aligned} \tag{3.15}$$

Table 13: Example 3.13.

b	MAE	m_δ	Iter.	CPU time (seconds)
5	1.487×10^{-2}	2	4	3.125×10^{-2}
8	2.183×10^{-4}	2	4	3.125×10^{-2}
20	4.517×10^{-9}	2	4	3.125×10^{-2}
30	1.205×10^{-13}	2	4	3.125×10^{-2}

where $\delta(b) := e^{-b}$. Therefore, I_2 can be considered as noise of the data $F_{13}(p)$, that is,

$$F_{13}^{\delta}(p) := F_{13}(p) - \delta(b), \tag{3.16}$$

where $\delta(b) := e^{-b}$. In this example the following parameters are used: $d = 2$, $\kappa = 10^{-1}$ for $\delta = e^{-5}$ and $\kappa = 10^{-5}$ for $\delta = 10^{-8}, 10^{-20}$ and 10^{-30}. Table 13 shows that the error decreases as the parameter b increases. The approximate solution obtained by the proposed method converges to the function $f_{13}(t)$ as b increases (see Figure 13).

4. Conclusion

We have tested the proposed algorithm on the wide class of examples considered in the literature. Using the rule (2.61) and the stopping rule (2.83), the number of terms in representation (2.69), the discrete data $F_\delta(p_j)$, $j = 0, 1, 2, \ldots, m$, and regularization parameter a_{n_δ}, which are used in computing the approximation $f_m^\delta(t)$ (see (2.69)) of the unknown function $f(t)$, are obtained automatically. Our numerical experiments show that the computation time (CPU time) for approximating the function $f(t)$ is small, namely, CPU time ≤ 1 second, and the proposed iterative scheme and the proposed adaptive stopping rule yield stable solution given the noisy data with the noise level δ. The proposed method also works for f with non-compact support as shown in Example 3.13. Moreover, in the proposed method we only use a simple representation (2.69) which is based on the kernel of the Laplace transform integral, so it can be easily implemented numerically.

References

[1] V. A. Morozov, *Methods for Solving Incorrectly Posed Problems*, Springer, New York, NY, USA, 1984.

[2] A. G. Ramm, *Inverse Problems*, Springer, New York, NY, USA, 2005.

[3] S. W. Indratno and A. G. Ramm, "Dynamical systems method for solving ill-conditioned linear algebraic systems," *International Journal of Mathematics and Computer Science*, vol. 2, no. 4, pp. 308–333, 2009.

[4] A. G. Ramm, *Dynamical Systems Method for Solving Operator Equations*, vol. 208 of *Mathematics in Science and Engineering*, Elsevier B. V., Amsterdam, The Netherlands, 2007.

[5] K. S. Crump, "Numerical inversion of Laplace transforms using a Fourier series approximation," *Journal of the Association for Computing Machinery*, vol. 23, no. 1, pp. 89–96, 1976.

[6] B. Davies and B. Martin, "Numerical inversion of the Laplace transform: a survey and comparison of methods," *Journal of Computational Physics*, vol. 33, no. 1, pp. 1–32, 1979.

[7] H. Dubner and J. Abate, "Numerical inversion of Laplace transforms by relating them to the finite Fourier cosine transform," *Journal of the Association for Computing Machinery*, vol. 15, pp. 115–123, 1968.

[8] P. den Iseger, "Numerical transform inversion using Gaussian quadrature," *Probability in the Engineering and Informational Sciences*, vol. 20, no. 1, pp. 1–44, 2006.

[9] V. V. Kryzhniy, “Numerical inversion of the Laplace transform: analysis via regularized analytic continuation,” *Inverse Problems*, vol. 22, no. 2, pp. 579–597, 2006.

[10] P. J. Davis and P. Rabinowitz, *Methods of Numerical Integration*, Computer Science and Applied Mathematics, Academic Press, Orlando, Fla, USA, 2nd edition, 1984.

[11] V. I. Krylov and N. Skoblya, *Reference Book on Numerical Inversion of the Laplace Transform*, Nauka i Technika, Minsk, Russia, 1968.

[12] R. G. Airapetyan and A. G. Ramm, “Numerical inversion of the Laplace transform from the real axis,” *Journal of Mathematical Analysis and Applications*, vol. 248, no. 2, pp. 572–587, 2000.

[13] L. D’Amore and A. Murli, “Regularization of a Fourier series method for the Laplace transform inversion with real data,” *Inverse Problems*, vol. 18, no. 4, pp. 1185–1205, 2002.

[14] C. W. Dong, “A regularization method for the numerical inverse of the Laplace transform,” *SIAM Journal on Numerical Analysis*, vol. 30, no. 3, pp. 759–773, 1993.

[15] H. Fujiwara, T. Matsura, S. Saitoh, and Y. Sawano, “Numerical real inversion of the Laplace transform by using a high-accurate numerical method,” private communication.

[16] A. Murli, S. Cuomo, L. D’Amore, and A. Galletti, “Numerical regularization of a real inversion formula based on the Laplace transform’s eigenfunction expansion of the inverse function,” *Inverse Problems*, vol. 23, no. 2, pp. 713–731, 2007.

[17] A. G. Ramm, “Inversion of the Laplace transform,” *Inverse Problems*, vol. 2, no. 4, pp. L55–L59, 1986.

[18] J. M. Varah, “Pitfalls in the numerical solution of linear ill-posed problems,” *SIAM Journal on Scientific Computing*, vol. 4, no. 2, pp. 164–176, 1983.

[19] J. G. McWhirter and E. R. Pike, “On the numerical inversion of the Laplace transform and similar Fredholm integral equations of the first kind,” *Journal of Physics A*, vol. 11, no. 9, pp. 1729–1745, 1978.

[20] H. Fujiwara, “exflib, a multiple precision arithmetic software,” http://www-an.acs.i.kyoto-u.ac.jp/~fujiwara/exflib/.

[21] K. M. Howell, “Multiple precision arithmetic techniques,” *The Computer Journal*, vol. 9, no. 4, pp. 383–387, 1967.

[22] S. W. Indratno and A. G. Ramm, “An iterative method for solving Fredholm integra equations of the first kind,” *International Journal of Mathematics and Computer Science*, vol. 2, no. 4, pp. 354–379, 2009.

[23] A. G. Ramm, “Discrepancy principle for DSM I,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 10, no. 1, pp. 95–101, 2005.

[24] A. G. Ramm, “Discrepancy principle for DSM II,” *Communications in Nonlinear Science and Numerical Simulation*, vol. 13, no. 7, pp. 1256–1263, 2008.

[25] S. Cuomo, L. D’Amore, A. Murli, and M. Rizzardi, “Computation of the inverse Laplace transform based on a collocation method which uses only real values,” *Journal of Computational and Applied Mathematics*, vol. 198, no. 1, pp. 98–115, 2007.