Available online at www.sciencedirect.com

ScienceDirect

JOURNAL OF COMPUTATIONAL AND APPLIED MATHEMATICS

Journal of Computational and Applied Mathematics 198 (2007) 98–115

www.elsevier.com/locate/cam

Computation of the inverse Laplace transform based on a collocation method which uses only real values

S. Cuomo[a], L. D'Amore[a,*], A. Murli[a], M. Rizzardi[b]

[a] University of Naples, "Federico II", Complesso Universitario M. S. Angelo, via Cintia, 80126 Naples, Italy
[b] University of Naples "Parthenope", Via A. De Gasperi, 80133 Naples, Italy

Received 11 November 2005

Abstract

We develop a numerical algorithm for inverting a Laplace transform (LT), based on Laguerre polynomial series expansion of the inverse function under the assumption that the LT is known on the real axis only. The method belongs to the class of Collocation methods (C-methods), and is applicable when the LT function is regular at infinity. Difficulties associated with these problems are due to their intrinsic ill-posedness. The main contribution of this paper is to provide computable estimates of truncation, discretization, conditioning and roundoff errors introduced by numerical computations. Moreover, we introduce the pseudoaccuracy which will be used by the numerical algorithm in order to provide uniform scaled accuracy of the computed approximation for any x with respect to $e^{\sigma x}$. These estimates are then employed to dynamically truncate the series expansion. In other words, the number of the terms of the series acts like the regularization parameter which provides the trade-off between errors.

With the aim to validate the reliability and usability of the algorithm experiments were carried out on several test functions.

MSC: 65F22; 65G10; 65R32

Keywords: Laplace transform real inversion; Collocation methods; Error analysis; Numerical algorithm

1. Introduction

Given the Laplace transform

$$F(z) = \int_0^\infty e^{-zx} f(x)\,dx, \quad \mathrm{Re}(z) > \sigma_0, \tag{1}$$

where $F(z)$ is computable on the real axis only and σ_0 is the abscissa of convergence of $F(z)$, we focus on the numerical approximation of *f*, a real-valued function on the non-negative real line and regular at zero, for any real value of $x \geqslant 0$.

When $F(z)$ is known for real values of z only, the inverse problem (1) is usually known as the *real inversion problem*.

The difficulties associated with this inverse problem are due to its intrinsic ill-posedness, and the derivation of error estimates is the first step towards the development of a reliable inversion algorithm.

* Corresponding author.
E-mail addresses: salvatore.cuomo@dma.unina.it (S. Cuomo), luisa.damore@dma.unina.it (L. D'Amore), almerico.murli@dma.unina.it (A. Murli), mariarosaria.rizzardi@uniparthenope.it (M. Rizzardi).

0377-0427/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.cam.2005.11.017

Starting from [8], where the authors generalized and formally characterized the Collocation methods (C-methods) for numerical inversion of the Laplace transform, we describe and analyze a numerical algorithm based on the Laguerre expansion of $f(x)$:

$$f(x) = e^{\sigma x} \sum_{k=0}^{\infty} c_k e^{-bx} L_k(2bx), \tag{2}$$

where $L_k(2bx)$ is the Laguerre polynomial of degree k, while σ and b are parameters of the series.

In this paper we first provide a detailed error analysis, then we introduce computable estimates of truncation, discretization and roundoff errors which are employed to determine the number of the terms of the series. This parameter acts like a *regularization parameter* providing the trade-off between errors introduced by the numerical *approximation* of (2).

The paper is organized as follows: in Section 2 we give preliminary definitions and assumption concerning C-methods; then Section 3 reports theoretical analysis of truncation, discretization and roundoff errors; in Section 4 we derive computable error estimates, and finally, conclusions are provided in Section 5 together with numerical experiments on exact data, perturbed data and on an application test case.

2. Outline of C-methods

We give preliminary assumptions concerning the Laplace transform, and review the main steps of a C-method [8].

The domain of applicability of this algorithm is the class S_γ where

Definition 2.1. Let S_γ denotes the class of all functions whose analytic continuation, $F(z)$, can be assumed in the form

$$F(z) = z^{-\gamma} G(z), \tag{3}$$

for some fixed $\gamma > 0$, where G is analytic at infinity.

Definition 2.2. Let

$$z = \frac{2b}{1-w} + \sigma - b \quad (\sigma > \sigma_0,\ b > 0) \tag{4}$$

be the Möbius transformation between the z-plane and the w-plane.

We define a transformation $H : S_\gamma \to S_\gamma$ such that

$$H : h \in S_\gamma \to H(h(w)) = \left(\frac{2 \cdot b}{1-w}\right)^{\gamma} h(w) \in S_\gamma. \tag{5}$$

Applying H to Laplace transform F, we define the function

$$\Phi(w) = H(F(z)) = \left(\frac{2 \cdot b}{1-w}\right)^{\gamma} F\left(\frac{2 \cdot b}{1-w} + \sigma - b\right). \tag{6}$$

We consider a C-method, in which $f(x)$ is approximated by

$$\tilde{f}_N(x) = e^{\sigma x} \sum_{k=0}^{N-1} \tilde{c}_k e^{-bx} L_k^{(\gamma-1)}(2bx), \tag{7}$$

where $L_k^{(\gamma-1)}$ is the generalized Laguerre polynomial of order γ and degree k, obtained by truncating the infinite series expansion and approximating the coefficients c_k by $\tilde{c}_k$ as described at the end of this section in Algorithm 1. In the following we assume $\gamma = 1$.

A C-method [8] consists essentially of the three main steps described in Algorithm 1.

Algorithm 1. *C-method*

Step 1: Evaluation of the function $\Phi(w) = H(F(w))$ at

$$w_k = \cos\left(\frac{2k+1}{N}\frac{\pi}{2}\right), \quad k = 0, \dots, N-1,$$

the N zeros of the Chebyschev polynomial of first kind of degree N;

Step 2: Computation of the coefficients of the polynomial $l_{N-1} \in \Pi_N$ that interpolates $\Phi(w)$ at w_k:

$$l_{N-1}(w) = \sum_{k=0}^{N-1} \tilde{c}_k w^k.$$

Step 3: Computation of the approximation $\tilde{f}_N(x)$:

$$\tilde{f}_N(x) = \mathrm{e}^{\sigma x} \sum_{k=0}^{N-1} \tilde{c}_k \mathrm{e}^{-bx} L_k(2bx)$$

of the original inverse Laplace transform $f(x)$.

3. Error analysis and pseudoaccuracy

The major challenge of the design of numerical algorithms for inverting the Laplace transform is the computation of the inverse Laplace transform with a uniform accuracy for any given value of x. Difficulties are primarily due to the exponential factor $\mathrm{e}^{\sigma x}$ in (2) as well as in (7), which degrades the accuracy of the computed approximation $f_N^{\mathrm{comp}}(x)$ as x grows. In order to address this problem, as in [2,4] in the following we refer to the *pseudoaccuracy* which provides a uniform accuracy scaled with respect to $\mathrm{e}^{\sigma x}$.

Definition 3.1. Let

$$f_N(x) = \mathrm{e}^{\sigma x} \sum_{k=0}^{N-1} c_k \mathrm{e}^{-bx} L_k(2bx). \tag{8}$$

The function

$$\varepsilon_{\mathrm{trunc}}(x, \sigma, b, N) = \frac{1}{\mathrm{e}^{\sigma x}} \left[\mathrm{e}^{\sigma x} \sum_{k=N}^{\infty} \mathrm{e}^{-bx} c_k L_k(2bx) \right] = \sum_{k=N}^{\infty} \mathrm{e}^{-bx} c_k L_k(2bx) \tag{9}$$

is referred to as the *truncation error* with respect to the pseudoaccuracy.

Definition 3.2. The function

$$\varepsilon_{\mathrm{discr}}(x, \sigma, b, N) = \frac{1}{\mathrm{e}^{\sigma x}} \left[\mathrm{e}^{\sigma x} \sum_{k=0}^{N-1} \mathrm{e}^{-bx} (c_k - \tilde{c}_k) L_k(2bx) \right] = \sum_{k=0}^{N-1} \mathrm{e}^{-bx} (c_k - \tilde{c}_k) L_k(2bx) \tag{10}$$

is referred to as the *discretization error* with respect to the pseudoaccuracy.

In the following the quantity:

$$\varepsilon_{\mathrm{td}}(x, \sigma, b, N) = \varepsilon_{\mathrm{trunc}} + \varepsilon_{\mathrm{discr}}$$

is used to indicate the pseudoaccuracy with respect to the discretization and truncation errors.

Beyond truncation and discretization errors, the numerical computation of $\tilde{f}_N(x)$ involves roundoff errors. Here we give some definitions concerning how the C-method works in a finite precision arithmetic system.

Let

$$\hat{f}_N(x) = \mathrm{e}^{\sigma x} \sum_{k=0}^{N-1} \hat{c}_k \mathrm{e}^{-bx} L_k(2bx), \tag{11}$$

where $\{\hat{c}_k\}_{k=0,\dots,N-1}$ are the computed values of coefficients $\{\tilde{c}_k\}_{k=0,\dots,N-1}$.

Definition 3.3. The function

$$\varepsilon_{\mathrm{cond}}(x, \sigma, b, N) = \frac{1}{\mathrm{e}^{\sigma x}} \mathrm{e}^{\sigma x} \left[\sum_{k=0}^{N-1} \mathrm{e}^{-bx} (\tilde{c}_k - \hat{c}_k) L_k(2bx) \right] = \sum_{k=0}^{N-1} \mathrm{e}^{-bx} (\tilde{c}_k - \hat{c}_k) L_k(2bx) \tag{12}$$

is referred to as the *conditioning error* with respect to the pseudoaccuracy.

Finally, let

$$f_N^{\mathrm{comp}}(x) = fl(\hat{f}_N(x)) = fl\left(\mathrm{e}^{\sigma x} \sum_{k=0}^{N-1} \hat{c}_k \mathrm{e}^{-bx} L_k(2bx) \right), \tag{13}$$

the evaluation of $\hat{f}_N(x)$ in finite precision arithmetic.

Definition 3.4. The function

$$\varepsilon_{\mathrm{round}}(x, \sigma, b, N) = \frac{1}{\mathrm{e}^{\sigma x}} [\hat{f}_N(x) - fl(\hat{f}_N(x))], \tag{14}$$

is referred to as the *roundoff error* with respect to the pseudo accuracy.

Finally, let

$$\varepsilon_{\mathrm{app}}(x, \sigma, b, N) = \varepsilon_{\mathrm{trunc}}(x, \sigma, b, N) + \varepsilon_{\mathrm{discr}}(x, \sigma, b, N) + \varepsilon_{\mathrm{cond}}(x, \sigma, b, N), \tag{15}$$

be the *approximation error* with respect to the pseudoaccuracy. Then

$$f(x) = f_N^{\mathrm{comp}}(x) + \mathrm{e}^{\sigma x} (\varepsilon_{\mathrm{app}}(x, \sigma, b, N) + \varepsilon_{\mathrm{round}}(x, \sigma, b, N)). \tag{16}$$

3.1. Truncation and discretization error estimate

The aim of this section is to give an upper bound of the truncation and discretization errors, which will be used to provide a computable estimate.

Let

$$R_{\mathrm{conv}}(\sigma, b) = \sqrt{b^2 - \sigma^2}$$

and following [7] it is

$$\varepsilon_{\mathrm{td}}(x, \sigma, b, N) = I(x, r, b) \cdot D_N(r, R) \cdot L(x, b),$$

where

$$I(x, r, b) = \int_0^{2\pi} \exp\left(-4rbx \,\mathrm{Re}\left(\frac{\mathrm{e}^{\mathrm{i}\theta}}{(r\mathrm{e}^{\mathrm{i}\theta} - 1)^2} \right) \right) \left| \frac{r\mathrm{e}^{\mathrm{i}\theta} + 1}{r\mathrm{e}^{\mathrm{i}\theta} - 1} \right| \mathrm{d}\theta. \tag{17}$$

The parameter r is related to $R_{\mathrm{conv}}(\sigma, b)$ through $R(\sigma, b)$[1]

$$0 < R < R_{\mathrm{conv}}(\sigma, b), \quad R < r < R + \sqrt{R^2 - 1}.$$

Moreover,

$$D_N(r, R) = \alpha(r, R)\frac{r^N + r^{-N}}{(R + \sqrt{R^2 - 1})^N + (R - \sqrt{R^2 - 1})^N}, \tag{18}$$

where

$$\alpha(r, R) = \frac{1}{\pi}\frac{\mu(\Phi)}{1 - (1/2R)(r + (1/r))}$$

and

$$\mu(\Phi(w)) = \max_{w \in C_R} |\Phi(w)|,$$

where C_R is the disk centered at the origin of complex plane with radius R. Finally,

$$L(x, b) = \mathrm{e}^{-bx} \tag{19}$$

depends on b and x.

The following properties allow us to determine estimates of $I(x, r, b)$ and $D_N(r, R)$.

Proposition 3.5. *For a fixed $x > 0$ and $b > 0$,*

$$\lim_{r \to +\infty} I(x, r, b) = 2\pi. \tag{20}$$

Proof. Let $x > 0$ and $b > 0$ be fixed, and let $\{r_n\}_{n \in N}$ be a sequence of numbers, such that

$$r_n > 1 \quad \forall n \in N \quad \text{and} \quad \lim_{n \to \infty} r_n = +\infty. \tag{21}$$

It is sufficient to demonstrate that

$$\lim_{n \to \infty} \psi_n(\theta) = 1 \quad \text{uniformly for } \theta \in [0, 2\pi]$$

and

$$\forall n \in N, \quad \psi_n(\theta) = \exp\left\{-4xb \cdot \mathrm{Re}\left[\frac{r_n \mathrm{e}^{\mathrm{i}\theta}}{(r_n \mathrm{e}^{\mathrm{i}\theta} - 1)^2}\right]\right\} \left|\frac{r_n \mathrm{e}^{\mathrm{i}\theta} - 1}{r_n \mathrm{e}^{\mathrm{i}\theta} + 1}\right|.$$

It holds that,

$$0 \leqslant \left|\frac{r_n \mathrm{e}^{\mathrm{i}\theta}}{(r_n \mathrm{e}^{\mathrm{i}\theta} - 1)^2}\right| \leqslant \frac{r_n}{r_n^2 \cos(2\theta) - 2r_n \cos(\theta) + 1}. \tag{22}$$

Then using (21) and (22), we have

$$\lim_{n \to \infty} \left|\frac{r_n \mathrm{e}^{\mathrm{i}\theta}}{(r_n \mathrm{e}^{\mathrm{i}\theta} - 1)^2}\right| = 0.$$

Moreover from (21),

$$\lim_{n \to \infty} \frac{r_n \mathrm{e}^{\mathrm{i}\theta} - 1}{r_n \mathrm{e}^{\mathrm{i}\theta} + 1} = 1 \quad \text{uniformly for } \theta \in [0, 2\pi].$$

[1] For simplicity of notation, in the rest of paper we indicate $R(\sigma, b)$ by R.

Thus,

$$\lim_{n\to\infty} \psi_n = 1 \quad \text{uniformly for } [0, 2\pi]. \qquad \square$$

The next Proposition gives a convergence result for $D_N(r, R)$.

Proposition 3.6. *Let* $R > 1$, *then*

$$\lim_{N\to+\infty} D_N(r, R) = 0. \tag{23}$$

Proof. We have

$$\frac{r^N + r^{-N}}{(R+\sqrt{R^2-1})^N + (R-\sqrt{R^2-1})^N} \leqslant \left(\frac{r}{R+\sqrt{R^2-1}}\right)^N + \left(\frac{1/r}{R+\sqrt{R^2-1}}\right)^N.$$

It follows from (17) that

$$0 \leqslant D_N(r, R) \leqslant \alpha(r, R)\left[\left(\frac{r}{R+\sqrt{R^2-1}}\right)^N + \left(\frac{1/r}{R+\sqrt{R^2-1}}\right)^N\right]. \tag{24}$$

Moreover, since

$$0 < \frac{r}{R+\sqrt{R^2-1}} < 1 \quad \text{and} \quad 0 < \frac{1/r}{R+\sqrt{R^2-1}} < 1, \tag{25}$$

we obtain

$$\lim_{N\to\infty} D_N(r, R) = 0. \qquad \square$$

The next Proposition gives upper and lower bounds for $D_N(r, R)$.

Proposition 3.7. *If* $R > 1$, *then*

$$\frac{1}{2} \leqslant \lim_{N\to+\infty} \frac{D_N(r, R)}{\alpha(r, R)(r/R+\sqrt{R^2-1})^N} \leqslant 1.$$

Proof. It follows that

$$\frac{D_N(r, R)}{\alpha(r, R)(r/R+\sqrt{R^2-1})^N} \leqslant \left[1 + \frac{1}{r^{2N}}\right]. \tag{26}$$

Moreover,

$$\frac{D_N(r, R)}{\alpha(r, R)(r/R+\sqrt{R^2-1})^N} \geqslant \frac{1}{2}\left[1 + \frac{1}{r^{2N}}\right]. \tag{27}$$

Finally, (26) and (27) show that

$$\frac{1}{2} \leqslant \lim_{N\to+\infty} \frac{D_N(r, R)}{\alpha(r, R)(r/R+\sqrt{R^2-1})^N} \leqslant 1. \qquad \square$$

The next theorem gives the estimate of truncation and discretization errors.

Theorem 3.8. *If* $R > 1$, *for* x, σ *and* b *fixed*, *it exists* $\bar{N}$ *such that for all* $N > \bar{N}$ *it holds*

$$|\varepsilon_{\mathrm{td}}(x, \sigma, b, N)| \leqslant 2\pi\alpha(r, R)\left(\frac{r}{R+\sqrt{R^2-1}}\right)^N. \tag{28}$$

Proof. Proposition 3.5 yields

$$|\varepsilon_{\mathrm{td}}(x,\sigma,b,N)| \leqslant 2\pi D_N(r,R).$$

Moreover, by Proposition 3.7, there exists $\bar{N}$ such that for all $N > \bar{N}$, we have

$$|\varepsilon_{\mathrm{td}}(x,\sigma,b,N)| \leqslant 2\pi\alpha(r,R)\left(\frac{r}{R+\sqrt{R^2-1}}\right)^N. \qquad \square$$

Finally, the following theorem provides the upper bound for truncation and discretization errors:

Theorem 3.9. *Given x, σ, and b, we have*

$$|\varepsilon_{\mathrm{td}}(x,\sigma,b,N)| \leqslant \left(K^* \frac{1}{R^N}\right), \tag{29}$$

where $K^* = |\Phi(0)|$ is the absolute value of the first term, $\tilde{c}_0$, of MacLaurin series expansion of $\Phi(w)$ and $R = R_{\mathrm{conv}}$.

3.2. Conditioning and roundoff estimates

The aim of this section is to estimate the conditioning error.

The core of the C-method is step 2 where the coefficients $\hat{c}_k$ are computed by solving the Vandermonde system

$$V_N^T \tilde{c} = \Phi \tag{30}$$

which is strongly illconditioned. Classical norm-wise results (cf. [5]) about roundoff errors introduced in the computation of coefficients $\hat{c}_k$ can give a too pessimistic bound. Indeed, the standard bounds derived using the ∞-norm cannot take into account the magnitude of each coefficient $\tilde{c}_k$.

A componentwise error analysis could be more realistic. To this aim we first observe that we can solve (30) by using the Bjorck–Pereira (BP) algorithm [1] which is composed of two main stages:

```
Stage I: % computation of $\tilde{a}_j = \tilde{a}_j^{(n)}$   $(j = 0, \ldots, n)$
$\tilde{a}_j^{(0)} = \Phi_j$ $(j = 0, \ldots, n)$
    for $k = 0$ to $n - 1$
    $\tilde{a}_j^{(k+1)} = \dfrac{\tilde{a}_j^{(k)} - \tilde{a}_{j-1}^{(k)}}{w_j - w_{j-k+1}}$   $(j = n, n-1, \ldots, k+1)$
    endfor
Stage II: % computation of $\tilde{c}_j = \tilde{c}_j^{(0)}$   $(j = 0, \ldots, n)$
$\tilde{c}_j^{(n)} = \tilde{a}_j^{(n)}$ $(j = 0, \ldots, n)$
    for $k = n - 1$ to 0 step $-1$
    $\tilde{c}_j^{(k)} = \tilde{c}_j^{(k+1)} - w_k \cdot \tilde{c}_{j+1}^{(k+1)}$   $(j = k, \ldots, n-1)$
    endfor
```

and for this algorithm in [9] a specific error analysis is carried out. In particular it holds

Theorem 3.10. *Suppose that the BP algorithm is carried out on a finite floating point arithmetic system, where u is the machine epsilon; the computed solution $\hat{c}_k$ of* (30) *satisfies*

$$|\tilde{c}_k - \hat{c}_k| \leqslant u|M_k^{(0)}||\tilde{c}_k| + \mathrm{o}(u^2), \quad k = 0, \ldots, N-1, \tag{31}$$

where $|M_k^{(0)}|$ is the forward error propagated at the kth step during the computation of the coefficients $\hat{c}_k$.

Theorem (3.10) gives an upper bound of the componentwise error for each $\tilde{c}_k$ by using the forward error analysis applied to BP algorithm. In the sequel we refer to (31) as the *computational bound*.

By using the componentwise upper bound we prove the following result:

Theorem 3.11. *Let u be the machine epsilon. Let σ, x and b be fixed. Then, it holds*

$$|\varepsilon_{\text{cond}}(x,\sigma,b,N)| \leqslant Nu|\Phi(0)| \max_{k=0,\ldots,N-1} |M_k^{(0)}|. \tag{32}$$

Proof. By definition of $\varepsilon_{\text{cond}}$,

$$|\varepsilon_{\text{cond}}| \leqslant \sum_{k=0}^{N-1} |\tilde{c}_k - \hat{c}_k|\,|\mathrm{e}^{-bx}L_k(2bx)|.$$

As in [8],

$$|\mathrm{e}^{-bx}L_k(2bx)| \leqslant 1 \quad x \geqslant 0 \text{ and for all } k. \tag{33}$$

Then by using (31) and neglecting the term $\mathrm{o}(u^2)$

$$|\varepsilon_{\text{cond}}| \leqslant \sum_{k=0}^{N-1} |\tilde{c}_k - \hat{c}_k| \leqslant u \sum_{k=0}^{N-1} |M_k^{(0)}||\tilde{c}_k|.$$

Observing that

$$\tilde{c}_{k+1} < \tilde{c}_k, \quad k = 0,\ldots,N-1,$$

and

$$\tilde{c}_0 = \Phi(0)$$

we obtain

$$|\varepsilon_{\text{cond}}| \leqslant u \sum_{k=0}^{N-1} |M_k^{(0)}||\tilde{c}_k| \leqslant Nu|\Phi(0)| \max_{k=0,\ldots,N-1} |M_k^{(0)}|. \qquad \square$$

Now we give some results about the roundoff error $\varepsilon_{\text{round}}(x,\sigma,b,N)$. In order to estimate $\varepsilon_{\text{round}}$ we write (11) as a scalar product,

$$\hat{f}_N(x) = \langle \hat{c}, g_k(x) \rangle = \sum_{k=0}^{N-1} \hat{c}_k g_k(x), \tag{34}$$

where $g_k(x) = \mathrm{e}^{\sigma x}\mathrm{e}^{-bx}L_k(2bx)$ and $\hat{c} = (\hat{c}_0,\ldots,\hat{c}_{N-1})$.

Remark 3.12. Given x, σ and b, if $f_N^{\text{comp}}(x)$ is the computed value of $\hat{f}_N(x)$, then by using standard results about roundoff error analysis of a scalar product carried out in a finite arithmetic precision system, with u as machine epsilon, it holds

$$|\mathrm{e}_{\text{round}}(x,\sigma,b,N)| = |f_N^{\text{comp}}(x) - \hat{f}_N(x)| \leqslant \frac{Nu}{1-Nu}|\hat{f}_N(x)| \tag{35}$$

that is the roundoff error bound in the computation of (11) grows linearly with N.

Theorem 3.13. *Given x, σ and b,*

$$|\varepsilon_{\text{round}}(x,\sigma,b,N)| \leqslant \frac{Nu}{1-Nu} N|\hat{\Phi}(0)|, \tag{36}$$

where $\hat{\Phi}(0)$ is the computed value of the first term of MacLaurin series expansion of $\Phi(w)$ and u is the machine epsilon.

Proof. By Remark 3.12 we have

$$|\varepsilon_{\text{round}}| \leqslant \frac{Nu}{1-Nu} \left| \sum_{k=0}^{N-1} \hat{c}_k \mathrm{e}^{-bx} L_k(2bx) \right| .$$

It follows, in view of (33) that

$$|\varepsilon_{\text{round}}| \leqslant \frac{Nu}{1-Nu} \sum_{k=0}^{N-1} |\hat{c}_k|.$$

Observe

$$\hat{c}_{k+1} < \hat{c}_k \quad \forall\; k = 0, \ldots, N-1$$

and setting $\hat{c}_0$ be is the computed value of $\tilde{c}_0$, it holds

$$\hat{c}_0 = \hat{\Phi}(0).$$

Finally, we have that

$$|\varepsilon_{\text{round}}(x, \sigma, b, N)| \leqslant \frac{Nu}{1-Nu} N |\hat{\Phi}(0)|. \qquad \square$$

4. The global error estimate

The previous analysis tell us that while the truncation and discretization errors introduced in the Laguerre expansion (2), given x, σ and b, go to zero when N grows, the conditioning and roundoff errors increase. This means that the overall accuracy of the solution depends on a suitable trade-off between truncation, discretization, roundoff and conditioning errors. The parameter N should act like a regularization parameter and help determining a suitable balance between various errors. With the aim of computing the optimal value of N, we first introduce and compute the global error estimate, which gives the asymptotic behavior of the global error as N grows. Moreover, by minimizing the global error estimate, the best attainable accuracy will be determined together with the corresponding value of N.

Definition 4.1. Let x be fixed. Let

$$E_{\text{td}}(\sigma, b, N) = K^* \frac{1}{R^N}$$

be the *computable estimate* of the truncation and discretization errors.

Definition 4.2. Given x, we introduce the *computable estimate* of the conditioning error

$$E_{\text{cond}}(\sigma, b, N) = Nu|\Phi(0)| \max_{k=0,\ldots,N-1} |M_k|.$$

By using Definitions 4.1, 4.2 and Theorems 3.9, it follows:

Definition 4.3. Given x, we define the global error estimate (GEE) function

$$\begin{aligned} \mathrm{GEE}(N) &= K^* \frac{1}{R^N} + Nu|\Phi(0)| \max_{k=0,\ldots,N-1} |M_k| \\ &= E_{\text{td}}(\sigma, b, N) + E_{\text{cond}}(\sigma, b, N), \end{aligned} \tag{37}$$

where $\mathrm{GEE}(N)$ is a *computable estimate* of $|\varepsilon_{\text{app}}(x, \sigma, b, N)|$.

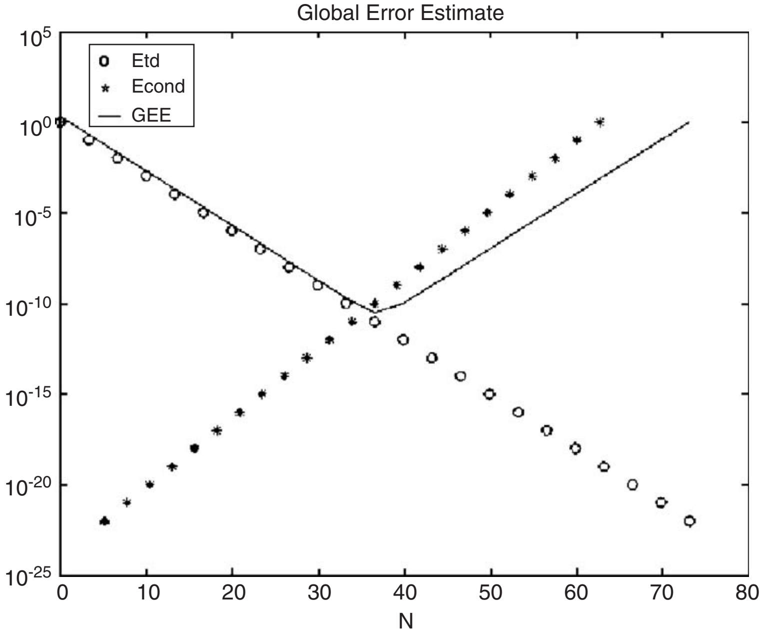

Fig. 1. GEE behavior versus N. The Laplace transform is $F_1(s)=\frac{s}{(s^2+1)^2}$, $K=1$, $R=1.72$, $x=1$, $\sigma=0.7$, $b=1.75$ (figure legend: circles (o) refer to E_{td} points, stars (*) refer to E_{cond} points, lines '-' refers to GEE function).

Fig. 1 shows the GEE behavior: given x, σ and b for a fixed value of N, $\mathrm{GEE}(N)$ gives the estimate of the errors introduced by the algorithm. We observe that there is an optimal value of N, say N^*, such that for $N \leqslant N^*$ the function $\mathrm{GEE}(N)$ decreases and for $N \geqslant N^*$, $\mathrm{GEE}(N)$ increases. Introduce

$$\psi(t)=\mathrm{GEE}(t)=\frac{K^*}{R^t}+t\cdot\Omega_\Phi, \qquad t\in\Re, \tag{38}$$

where $\Omega_\Phi = u|\Phi(0)|\max_k|M_k|$.

Theorem 4.4. *The function $\psi(t)$ is convex and achieves its minimum value at*

$$t^*=\log_R\frac{K^*}{\Omega_\Phi}, \tag{39}$$

where $K^*=|\Phi(0)|$ and $\Omega_\Phi=u|\Phi(0)|\max_k|M_k|$.

Proof. Computing the first derivative of $\psi(t)$ it follows that the function ψ' has only one zero at $t^*\in(0,+\infty)$ given by (39). Moreover $\psi''(t)>0$, then ψ is a convex function in $(0,+\infty)$ for $t\geqslant 0$. □

Definition 4.5. Let x, σ and b be fixed. We define the *best attainable accuracy*, ε^*, of a C-method, and the associated value N^*

$$\varepsilon^*=\mathrm{GEE}(N^*)=\min_N \mathrm{GEE}(N). \tag{40}$$

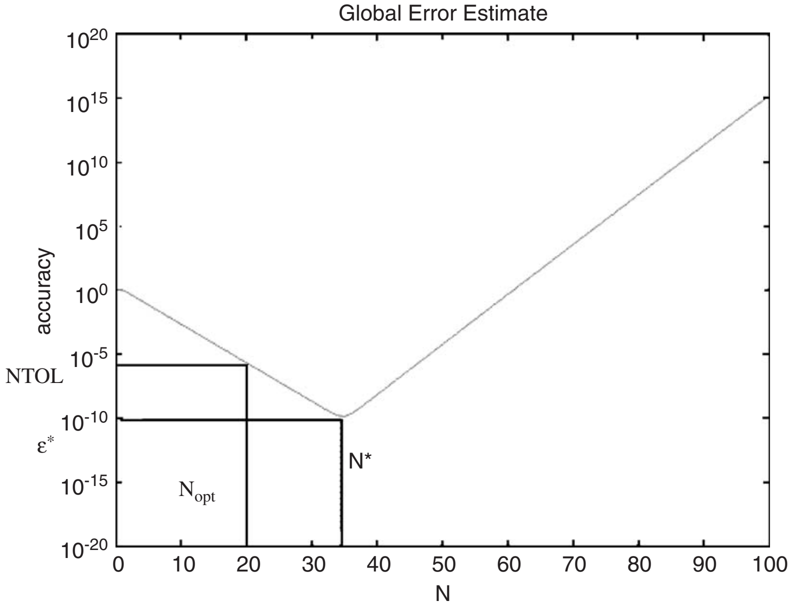

Fig. 2. GEE behavior versus N. The Laplace transform is $F_1(s) = \frac{s}{(s^2+1)^2}$, $K^* = 1$, $R = 1.8$, the minimum value of GEE is obtained at $N^* = 35$ and $\varepsilon^* = \mathrm{GEE}(35) = \mathrm{O}(10^{-11})$. If we set $x = 1$, $\sigma = 0.7$ and $\mathrm{TOL} = 10^{-5}$ the corresponding $N\,\mathrm{TOL}$ is $N\,\mathrm{TOL} = 10^{-6}$. Observe that in this case, $N_{\mathrm{opt}} = 20$.

Our algorithm is designed to find the value $N_{\mathrm{opt}} \leqslant N^*$ using the following scheme:

```
scale the input accuracy requirement tol as $N\,\mathrm{TOL} := \frac{\mathrm{tol}}{\mathrm{e}^{\sigma x}}$
and start with a value of $N$ such that: $\mathrm{GEE}(N-1) \leqslant \mathrm{GEE}(N)$;
$N_0 := N$;
repeat
$N_1 := N_0 + 1$;
evaluate $\mathrm{GEE}(N_0) := E_{\mathrm{td}}(N_0) + E_{\mathrm{cond}}(N_0)$;
evaluate $\mathrm{GEE}(N_1) := E_{\mathrm{td}}(N_1) + E_{\mathrm{cond}}(N_1)$;
if $(\mathrm{GEE}(N_1) \geqslant \mathrm{GEE}(N_0))$
FLAG $:= 3$;
endif
$N_0 := N_1$;
until $(\mathrm{GEE}(N_1) \geqslant N\,\mathrm{TOL}$ or FLAG $\neq 3)$
assume $N_{\mathrm{opt}} := N_1$;
```

In other words, for each value of x, once σ and b have been fixed, the algorithm dynamically selects the optimal value of N, N_{opt}, such that

$$\mathrm{GEE}(N^*) \leqslant \mathrm{GEE}(N_{\mathrm{opt}}) \leqslant \max\{N\,\mathrm{TOL}, \varepsilon^*\}. \tag{41}$$

This means that, once x, σ and b have been fixed, we have

$$\frac{|f(x) - f^{\mathrm{comp}}_{N_{\mathrm{opt}}}(x)|}{\mathrm{e}^{\sigma x}} \leqslant \frac{|f(x) - \hat{f}_{N_{\mathrm{opt}}}(x)|}{\mathrm{e}^{\sigma x}} + \frac{|f^{\mathrm{comp}}_{N_{\mathrm{opt}}}(x) - \hat{f}_{N_{\mathrm{opt}}}(x)|}{\mathrm{e}^{\sigma x}} \tag{42}$$

$$\leqslant \mathrm{GEE}(\sigma, b, N_{\mathrm{opt}}) + (N_{\mathrm{opt}} u |\hat{\Phi}(0)|) \approx \max\{N\,\mathrm{TOL}, \varepsilon^*\}.$$

Eq. (42) gives the global error estimate, on which our algorithm is based, and allows us to compute the inverse Laplace transform $f^{\mathrm{comp}}_N(x)$ within $N\,\mathrm{TOL}$ (Fig. 2).

5. Numerical results

5.1. The role of σ and b parameters

Most numerical methods for inversion of the Laplace transform requires an upper bound of $\sigma > \sigma_0$, the abscissa of convergence of the Laplace transform to be inverted. We observe that

- if $\sigma - \sigma_0$ is chosen too small unnecessary many terms of Laguerre series (2) are needed (slow convergence);
- if $\sigma - \sigma_0$ is chosen too large then we cannot compute an accurate numerical approximation because of the exponential growth factor.

A suitable choice is

$$\sigma = \sigma_0 + 0.7$$

(see [4]). Regarding the parameter b, if z_j is the singularity of the Laplace Transform $F(z)$ closest to σ_0, we have

$$\frac{b}{2} \geqslant \min_{\sigma > \sigma_0} |\sigma - z_j|.$$

We let

$$b = 2.5(\sigma - \sigma_0)$$

(see [4,6]).

5.2. Experiments

There are over 100 algorithms for numerical inversion of the Laplace transform [11], however, only few are devoted to the real inversion and, to our knowledge, none of them is effectively available as mathematical software, i.e. none of them can be used in practice for a numerical comparison. Indeed, usability and reliability of these inversion algorithms strongly depends on the selection of suitable chosen optimal values of some input parameters, which heavily affect numerical results. Often the obtained accuracy can still be improved by manually tuning the parameter values. Essah and Delves [3] state "[. . .] every numerical algorithm which is not designed to be used in an automatic way but requires to choose suitable parameters, cannot be compared [. . .]". On the other hand, one of the main features of the proposed algorithm is that it has been designed to work in an automatic way and to compute at run time suitable values of the incidental parameters. To our knowledge there is no other real inversion algorithm which can be used in such a transparent way to be compared with the proposed one.

Several experiments have been carried out on test functions shown in Table 1, here we report only few of them.

In order to validate the reliability and usability of the proposed algorithm, we assume the following model of degradation on vector data $F = (F(z_1), \ldots, F(z_n))$:

$$\text{Pert}\, F(i) = F(z_i)(1 + \text{noise}(i)), \quad i = 1, \ldots, n,$$

where the vector noise $=$ (noise(1), . . . , noise(n)), is a Gaussian noise, i.e. the coefficients noise(i) are independent random values with a Gaussian distribution with zero mean and variance δ, where

$$\delta = \frac{\|\text{noise}\|_2^2}{\|F\|_2^2}.$$

We provide three types of data test

(i) $\delta = u$, where u equals to the maximum accuracy of double precision (about 16 significant digits);
(ii) $\delta \leqslant d\,\text{tol}$, where $d\,\text{tol} > u$ is a preassigned noise level;
(iii) an application test with noisy data.

Data test (i) shows numerical stability of the algorithm and its reliability in terms of the errors (conditioning, truncation, discretization, and roundoff) estimates. Data test (ii) refers to the usability of the algorithm in presence of

Table 1
Database test functions

Test number	Laplace transform	Inverse Laplace function
1	$\frac{z^2-1}{(z^2+1)^2}$	$x\cos(x)$
2	$\frac{1}{(z+1)^2}$	$x\mathrm{e}^{-x}$
3	$\frac{1}{z^5}$	x^4
4	$\frac{1}{z^4}$	$\frac{x^3}{6}$
5	$\frac{z}{(z^2+1)^2}$	$0.5x\sin(x)$
6	$\frac{z}{(z+1)^2}$	$(1-4x)\mathrm{e}^{-4x}$
7	$\frac{z}{((z^2+4)(z^2+1))}$	$\frac{\cos(x)-\cos(3x)}{3}$
8	$\frac{1}{\sqrt{z^2+1}}$	$J_0(x)$ Bessel function
9	$\frac{2}{(z+2.5)^2-4}$	$\mathrm{e}^{-2.5x}\ \sinh(2x)$
10	$\frac{z*\sin(3)+2\ \cos(3)}{z^2+4}$	$\sin(2x+3)$
11	$\log(\frac{z^2+a^2}{z^2})$	$\frac{2}{x}(1-\cos(ax))$
12	$\frac{(\sqrt{(z^2+1)}-z)^2}{\sqrt{z^2+1}}$	$J_2(x)$ Bessel function
13	$\mathrm{atan}((2z)/(z^2-3))$	$\frac{2\ \sin(x)\ \cos(2x)}{x}$
14	$\frac{1}{(1+0.5z)^3}$	$4x^2\mathrm{e}^{-2x}$

perturbed data, and finally, the application test (iii) has been chosen in order to illustrate the practical applicability of our algorithm.

In order to give both a quantitative and a qualitative description of the numerical results, for each test we provide

- a table, containing information about x, N_{opt}, N TOL, FLAG, and errors;
- plot of errors, behavior versus N TOL;
- plot of the exact inverse function and the computed values $f(x_i)$.

Concerning data test (i) we consider the Laplace transform F_1. In this case the noise level is that of roundoff errors. Results are shown in Table 2. Fig. 3 shows the inverse function f_{exact} and the computed approximation f_{comp}; Fig. 4 reports the errors versus N TOL.

For data test (ii) we consider test function F_5 with noise level $\delta = 10^{-4}$. Results are shown in Table 3. Fig. 5 shows the inverse function f_{exact} and the computed approximation f_{comp}; Fig. 6 shows the errors versus N TOL. Both for data test (i) and (ii), the inverse function is computed first at $x \in [0, 5]$, then at $x \in [6, 16]$.

Finally, regarding the application test (iii), we consider the X-ray diffractometry, which is a type of non-destructive testing [10]. In X-ray diffractometry, we search for the stress tensor $\sigma(x_i)$ of a specimen, corresponding to a set of scanning points x_i, $i = 1, \ldots, m$, with the help of X-ray measurements $g(\theta_i)$ where θ_i denote the set of m tip-angles of the laboratory system. X-ray measurements are connected to the stress tensor by means of the Laplace Transform integral operator, then we need to invert the data vector $g(\theta_i)$. We assume that data g are measured in a laboratory with tip angles θ_i, $i = 1, \ldots, m$ giving to m scanning points $x_i = \mathrm{const}/\cos(\theta_i)$. The scanning points x_i were chosen non-uniformly in the interval $[a, b] = [1, 3.5]$, which is typical for data from X-ray, as

$$x_i = a/(\cos(\theta_i)), \quad \theta_i = \frac{(i-1)}{(m-1)}\arccos(a/b), \quad i = 1, \ldots, m$$

for $m = 11$.

Table 2
Data test (i)—Function F_1, $\sigma = 0.7$, $b = 1.75$, TOL $= 1.0\text{e} - 6$

N_{opt}	x	N TOL	trueabserr	estabserr	truerelerr	estrelerr	FLAG
31	0.0	1.0e−006	1.1e−018	1.3e−008	1.1e−018	1.3e−008	1
31	0.5	1.4e−006	1.3e−012	1.4e−007	2.9e−012	3.2e−007	1
32	1.0	2.0e−006	1.1e−011	2.9e−007	2.0e−011	5.3e−007	1
33	1.5	2.9e−006	1.1e−010	5.8e−007	1.0e−009	5.5e−006	1
33	2.0	4.1e−006	1.7e−010	1.2e−006	2.1e−010	1.4e−006	1
34	2.5	5.8e−006	8.2e−009	2.4e−006	4.1e−009	1.2e−006	1
35	3.0	8.2e−006	5.2e−007	9.8e−006	1.7e−007	3.3e−006	1
36	3.5	1.2e−005	9.5e−007	2.0e−005	2.9e−007	6.1e−006	1
36	4.0	1.6e−005	2.6e−006	4.0e−005	1.0e−006	1.5e−005	1
37	4.5	2.3e−005	1.8e−005	3.9e−005	1.8e−005	4.1e−005	1
37	5.0	3.3e−005	1.5e−005	7.9e−005	1.1e−005	5.6e−005	1
37	5.5	4.7e−005	1.3e−004	1.6e−004	3.4e−005	4.1e−005	1
37	6.0	6.7e−005	4.2e−005	3.2e−004	7.3e−006	5.6e−005	1
37	6.5	9.5e−005	6.1e−004	6.4e−004	9.6e−005	1.0e−005	1
37	7.0	1.3e−004	2.0e−004	1.3e−003	3.9e−005	2.5e−004	1
37	7.5	1.9e−004	2.4e−003	2.6e−003	9.1e−004	1.0e−004	1
37	8.0	2.7e−004	5.0e−004	5.3e−004	4.3e−004	4.5e−003	2

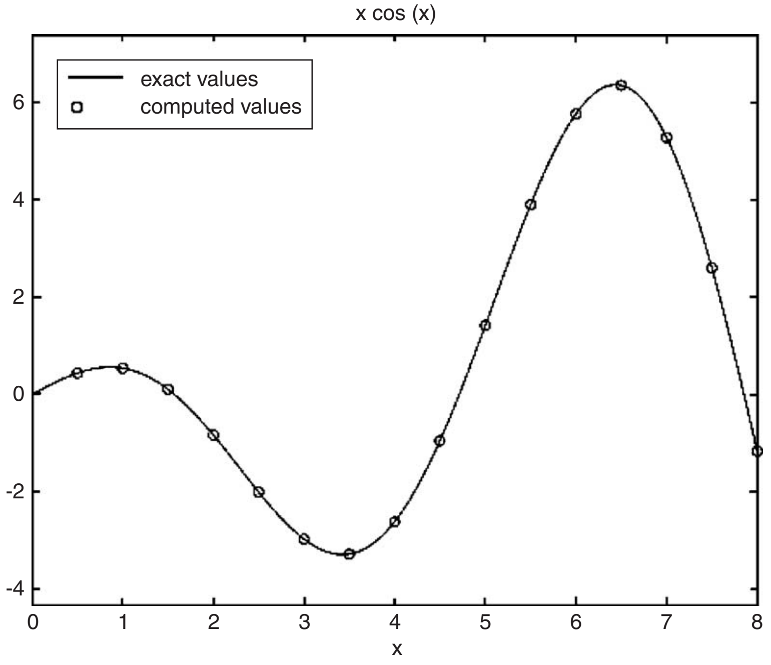

Fig. 3. Data test (i) on Function F_1. Inverse function $f(x) = x * \cos(x)$.

Following [10] we consider the test function F_{14} with noise level $\delta = 10^{-3}$. Results are shown in Table 4. Fig. 7 shows the inverse function f_{exact}, and the computed approximation f_{comp}.

As input, besides the x-values, the algorithm only requires the parameter TOL. We run the algorithm by giving TOL according to the noise level on vector data. In particular, for data set (i), we set TOL $= 1.0\text{e} - 06$. Looking at Table 2 and at the error behavior shown in Fig. 4, the computed error estimates agree well with the scaled pseudo accuracy NTOL. For data set (ii), we run the algorithm using as input accuracy TOL $= 1.0\text{e} - 04$. In this case, by looking at

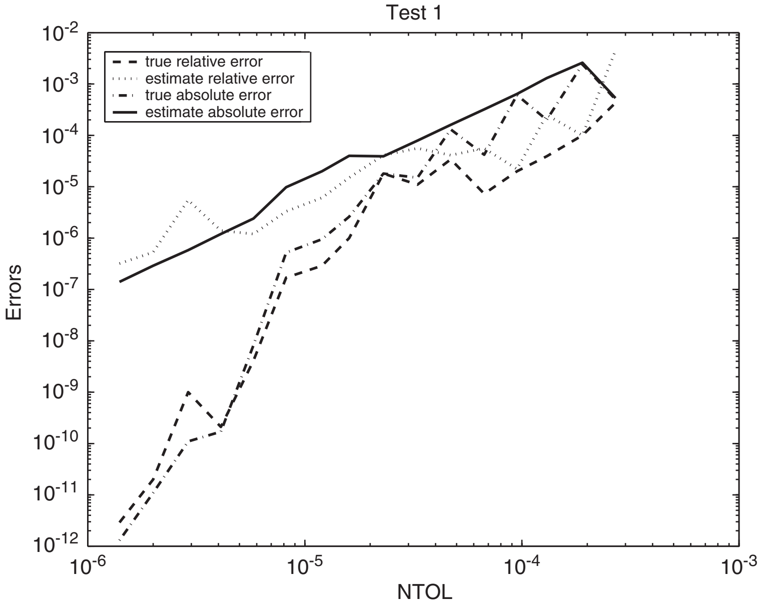

Fig. 4. Data test (i) on Function F_1. Errors versus N TOL plot.

Table 3
Data test (ii)—Function F_5, $\sigma = 0.7$, $b = 1.75$, TOL $= 1.0\text{e} - 4$

N_{opt}	x	N TOL	trueabserr	estabserr	truerelerr	estrelerr	FLAG
26	0.5	1.4e−004	1.8e−005	4.5e−005	1.5e−004	3.8e−004	1
27	1.0	2.0e−004	1.2e−004	6.5e−004	2.8e−004	3.5e−004	1
27	1.5	2.9e−004	3.0e−004	9.2e−004	4.1e−004	4.2e−004	1
28	2.0	4.1e−004	5.1e−004	5.3e−004	5.6e−004	6.5e−004	1
29	2.5	5.8e−004	6.4e−004	6.9e−004	7.5e−004	8.5e−004	1
30	3.0	8.2e−004	5.7e−004	6.7e−004	2.7e−003	3.3e−003	2
30	3.5	1.2e−003	2.5e−004	3.9e−004	4.1e−004	6.3e−004	1
31	4.0	1.6e−003	2.8e−004	5.5e−004	1.9e−004	3.6e−004	1
32	4.5	2.3e−003	6.1e−004	7.9e−004	4.2e−004	4.6e−004	1
33	5.0	3.3e−003	1.5e−003	1.8e−003	6.1e−004	7.7e−004	1
33	5.5	4.7e−003	1.7e−003	1.8e−003	8.9e−004	6.2e−004	1
34	6.0	6.7e−003	1.6e−003	2.3e−003	1.9e−003	2.7e−003	1
35	6.5	9.5e−003	2.5e−004	3.3e−003	1.4e−003	4.7e−003	1
36	7.0	1.3e−002	1.5e−004	4.6e−003	6.4e−005	7.0e−003	1
36	7.5	1.9e−002	1.7e−003	6.6e−003	4.7e−004	5.9e−003	1
37	8.0	2.7e−002	1.4e−003	7.5e−002	3.6e−004	1.9e−002	1

Table 3 and Fig. 6, it appears that as the data noise level δ increases, the accuracy on the computed solution decreases in a quite similar manner. Finally, for the application data set (iii), we set the input accuracy TOL $= 1.\text{e} - 03$. The algorithm always yields at least three correct significant digits.

Observe that as x grows, N TOL grows too; when N TOL becomes greater or equal 1, the algorithm stops. In this case it could help to try to run the algorithm with a smaller value of TOL (see in the Legend the FLAG indicator equals to 4).

The following legend is for the tables for the results of numerical experiments. All experiments were carried out using double precision (16 significant digits).

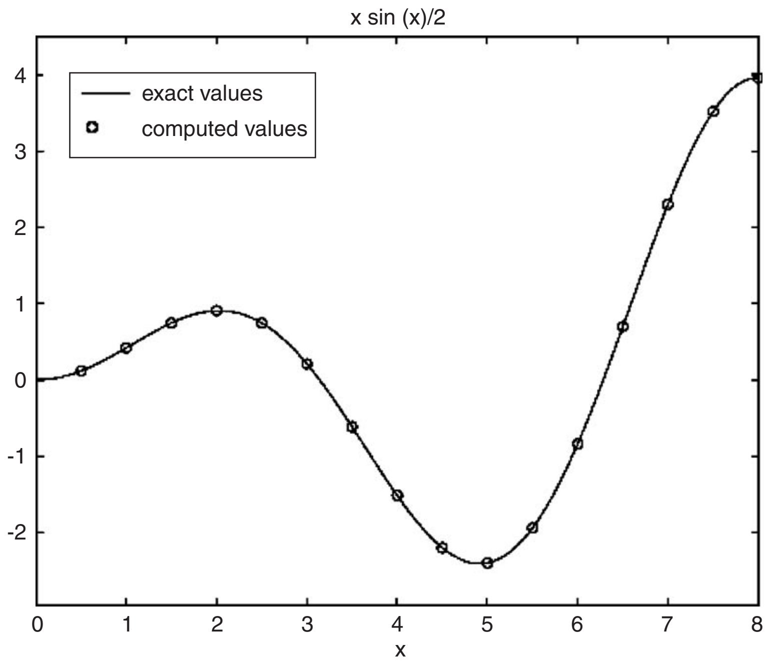

Fig. 5. Data test (ii) on Function F_5. Inverse function $f(x) = \frac{x*\sin(x)}{2}$.

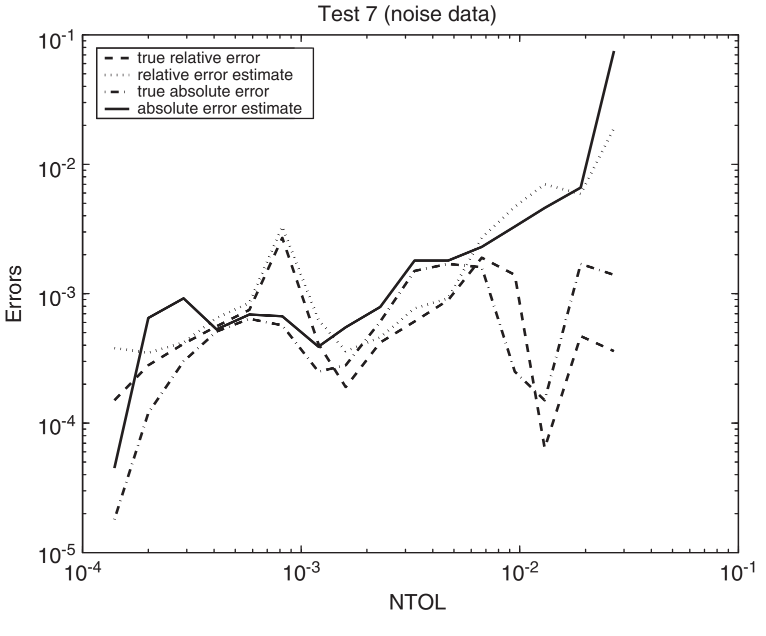

Fig. 6. Data test (ii) on Function F_5. Errors versus N TOL.

Legend

N_{opt}	number of terms computed by the algorithm
x	evaluation points
N TOL	pseudo-tolerance defined as TOL*exp(sigma $*$ x)

Table 4
Data test (iii)—Function F_{14}, $\sigma = 0.7$, $b = 1.75$, TOL $= 1.0\text{e} - 3$

N_{opt}	x	N TOL	trueabserr	estabserr	truerelerr	estrelerr	FLAG
27	1.	2.0e−003	2.3e−004	7.2e−004	4.3e−004	1.3e−004	1
27	1.0083	2.0e−003	2.4e−004	7.3e−004	4.3e−004	1.3e−004	1
27	1.0337	2.1e−003	2.4e−004	7.5e−004	4.5e−004	1.4e−004	1
27	1.0787	2.1e−003	2.6e−004	8.0e−004	4.7e−004	1.5e−004	1
27	1.1474	2.2e−003	2.7e−004	8.8e−004	5.1e−004	1.7e−004	1
27	1.2472	2.4e−003	3.0e−004	1.0e−004	5.7e−004	2.0e−004	1
28	1.3911	2.6e−003	3.2e−004	1.3e−004	6.7e−004	2.6e−004	1
28	1.6021	3.1e−003	3.4e−004	1.7e−004	8.2e−004	4.1e−004	1
28	1.9259	3.9e−003	3.4e−004	2.7e−004	1.1e−003	8.5e−003	1
29	2.4643	5.6e−003	2.9e−004	5.8e−004	1.7e−003	3.3e−003	1
31	3.5	1.2e−002	1.5e−004	2.5e−003	3.4e−003	5.6e−002	1

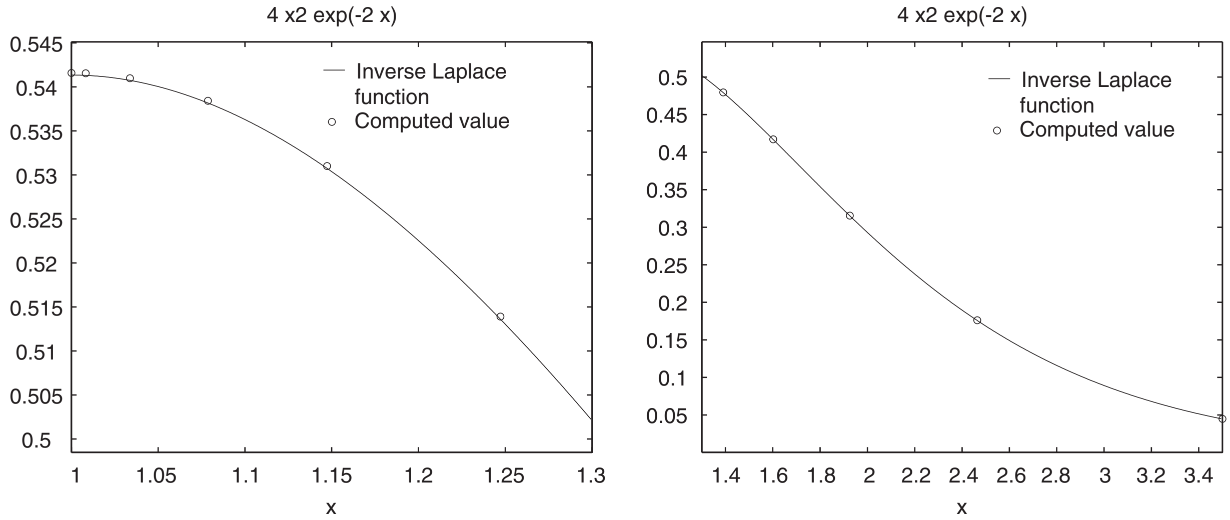

Fig. 7. Data test (iii) on Function F_{14}. Inverse function computed at $x_i \in [1, 1.3]$ on the LEFT and at $x_i \in [1.3, 3.5]$ on the RIGHT.

trueabserr absolute error of the computed solution defined as
$|f_{\text{exact}} - f_{\text{comp}}|$
estabserr computed estimate of trueabserr.
truerelerr relative error of the computed solution defined as
$|f_{\text{exact}} - f_{\text{comp}}|/|f_{\text{exact}}|$
if $f_{\text{exact}} = 0$ then truerelerr equals to trueabserr.
estrelerr computed estimate of truerelerr
FLAG on exit, a diagnostic
=1 the relative error estimate is given according to N TOL.
=2 the absolute error estimate is given according to N TOL.
=3 the tolerance seems to be too small and the algorithm
appears to be not convergent. Try with a greater value of TOL.
=4 Bad accuracy request. N TOL is greater than 1.

6. Concluding remarks

We describe a numerical method for computation of the inverse Laplace transform $f(x)$ in the real case based on a C-method. A complete error analysis of the computational steps needed to determine an approximate solution is presented. We also provide practical error estimates. These estimates are used for computing *the best attainable accuracy* on $f(x)$, defined by the minimum value of (the estimate of) the global error introduced on $f(x)$ by the numerical computations, together with the corresponding value of *N*. This allows us to develop a reliable, efficient and automatic stopping rule for computing the number of terms of the Laguerre expansion. In order to validate the reliability and usability of the proposed algorithm experiments were carried out on several test functions.

References

[1] A. Bjorck, V. Pereira, Solution of Vandermonde Systems of Equation, Math. Comput. 112 (24) (1970) 893–903.

[2] L. D'Amore, G. Laccetti, A. Murli, An implementation of a Fourier series method for the numerical inversion of the Laplace transform, ACM Trans. Math. Software 25 (3) (1999) 279–305.

[3] W.A. Essah, L.M. Delves, On the numerical inversion of Laplace transform, Inverse Problems (4) (1988) 705–724.

[4] B.S. Garbow, G. Giunta, J. Lyness, A. Murli, Software implementation of Week's method for the inverse Laplace transform problem, ACM Trans. Math. Software 15 (1988) 163–170.

[5] W. Gautschi, How (un)stable are Vandermonde systems?, Lecture Notes in Pure and Appl. Math. 124 (1990) 193–210.

[6] G. Giunta, G. Laccetti, M.R. Rizzardi, More on the Weeks method for the numerical inversion of the Laplace transform, Numer. Math. 54 (2) (1988) 192–200.

[7] G. Giunta, A. Murli, G. Smith, Error analysis of Rjabov algorithm for inverting a Laplace transform, Ricerche Mat—Naples XLIV (1) (1995) 207–219.

[8] G. Giunta, A. Murli, G. Smith, An analysis of bilinear transform-polynomial methods of inversion of Laplace transforms, Numer. Math. 69 (3) (1995) 269–282.

[9] N. Higham, Stability of algorithms for solving confluent Vandermonde-like systems, SIAM J. Matrix Anal. Appl. 11 (1) (1990) 23–41.

[10] T. Schuster, A stable inversion scheme for the Laplace transform using arbitrarily distributed data scanning points, J. Inverse and Ill-Posed Problems 11 (3) (2003) 263–287.

[11] P. Valko, J. Abate, Numerical Laplace inversion in rheological characterization, J. Non-Newtonian Fluid Mech. 116 (1) (2004) 395–406.