Applied Mathematics and Nonlinear Sciences 3(2) (2018) 487–502

Applied Mathematics and Nonlinear Sciences

https://www.sciendo.com

Review of numerical methods for NumILPT with computational accuracy assessment for fractional calculus

Dariusz W. Brzeziński [†]

Institute of Applied Computer Science, Lodz University of Technology, 18/22 Stefanowskiego St., 90-924 Łódź Poland

Submission Info

Communicated by Juan L.G. Guirao
Received 26th August 2018
Accepted 26th November 2018
Available online 1st December 2018

Abstract

In the paper we present results of accuracy evaluation of numerous numerical algorithms for the numerical approximation of the Inverse Laplace Transform. The selected algorithms represent diverse lines of approach to this problem and include methods by Stehfest, Abate and Whitt, Vlach and Singhai, De Hoog, Talbot, Zakian and a one in which the FFT is applied for the Fourier series convergence acceleration. We use C++ and Python languages with arbitrary precision mathematical libraries to address some crucial issues of numerical implementation. The test set includes Laplace transforms considered as difficult to compute as well as some others commonly applied in fractional calculus. Evaluation results enable to conclude that the Talbot method which involves deformed Bromwich contour integration, the De Hoog and the Abate and Whitt methods using Fourier series expansion with accelerated convergence can be assumed as general purpose high-accuracy algorithms. They can be applied to a wide variety of inversion problems.

Keywords: Numerical Approximation of the Inverse Laplace Transform, Fractional order differential Equations, Multi-precision Computing.
AMS 2010 codes: 65D30, 65D32, 65D99.

1 Introduction

Fractional calculus (FC) can be interpreted as an extension of the concept of derivative operator from integer order n to arbitrary order ν, where n is an integer and ν can be a real number. English term *fractional calculus* is misleading because it suggests that differentiation and integration order may assume non-integer orders only. The more appropriate description of this branch of mathematics would be differentiation and integration of any order. In practice, both integration as well as differentiation orders may assume real or complex values. In

[†] Corresponding author.
Email address: dbrzezinski@iis.p.lodz.pl

ISSN 2444-8656 doi:10.2478/AMNS.2018.2.00038

this context derivatives and integrals of integer order can be considered as one of special cases of fractional calculus [1–4].

FC can be applied for finding solutions of differential equations of fractional order (abbreviation FODE). For this reason, the research for a numerical method for finding their accurate and stable solutions has been set for the last 70 years within the scope of many scientific projects as for example [5–10].

The Laplace Transform (abbreviation LPT) is commonly applied for solving linear differential equations of integer order by converting them into algebraic equations, which, in this form, are simpler to solve [11–13]. However, a found solution has to be inverted by applying the Inverse Laplace Transform (abbreviation ILPT). This inversion is commonly done by using tables with so called the Laplace pairs, consisting of a Laplace transform and an corresponding original. If an original is not found, the numerical approximation of ILPT (abbreviation NumILPT) can be applied to find it.

NumILPT is an ill-posed inversion problem due to the inclusion of the multiplication of an exponential function of time in the inversion formula, in which algorithmic and finite precision errors can lead to exponential divergence of numerical solutions [14]. Due to this difficulty, one has been trying to find a method to mitigate its influence on final accuracy of numerically inverted ODE solutions for the last 70 years (as it is stated in [15] - the first recognized algorithm of NumILPT is the algorithm by D.V. Widder dated from 1946 [16]). As a result, currently there can be distinguished not only several main numerical approaches to the inversion problem, but also within a single approach there can be noted numerous different approximation methods to the various parts of the inversion formula.

Main approaches to the NumILPT include the use of: Bromwich contour integration, Fourier series expansion, finite differences and Laguerre and Legendre polynomials.

Bromwich contour can have a fixed shape or, to obtain better results, it can be deformed. The deformation can be done by using cotangent function or any other trigonometric function depending on which deformation is then more accurately integrated by a selected method of integration, e.g. the trapezoidal rule.

The exponential part of the inversion formula can be approximated by applying Taylor series expansion or Padée approximation. For the same purpose there can also be used Fourier series expansion, within which the exponential part is approximated with the use of a Padée approximant.

In this context, the main task within the scope of this paper is present the results of the evaluation of some of these algorithms for the inversion accuracy of Laplace transforms frequently used in FC.

However, it must be stressed that the accuracy evaluation is not the most difficult part of this task, but the right selection of the algorithms according to the aim of the scope of the evaluation (numerical inversion difficulty level) and their efficient programming which must include solutions to the implementation issues associated, among the others, with the limitations of the double precision computer arithmetic.

In the paper [17] we gave an initial evaluation for the inversion accuracy of several methods representing the main approaches to the NumILPT. Testbed included Laplace transforms of FODE solutions we have found in the most cited papers at those times. Currently, we would like to present results of a new evaluation inspired by the lecture of [18] and [19] publishing an impressive list of Laplace transforms used in the very practical applications of FC. The new results were obtained by using improved computer implementations of selected algorithms of NumILPT using C++ and Python programming languages.

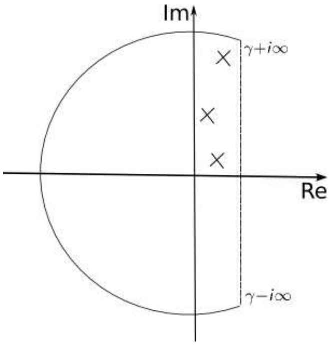
The paper is divided in the following sections. The first section includes short information about the Inverse Laplace Transform, its numerical counterpart - the numerical approximation of the Inverse Laplace Transform. It list all the methods used in the evaluation and gives briefly insights into their mathematical algorithms as well. The third subsection discuses the most important problems of theor programming and presents suggested solutions to the most severe of them. Sections four and five presents Laplace transforms criteria selection and how computational accuracy evaluation was conducted. Section six contained the presentation of the results in form of two plots with each evaluated method for each selected Laplace transform. The paper end with usual conclusions and extended list of references.

2 The Inverse Laplace Transform

The Inverse Laplace Transform $f(t)$ is defined as a contour integral

$$f(t) = \frac{1}{2\pi i}\int_{\gamma-i\infty}^{\gamma+i\infty} e^{st} F(s)\, ds,$$

where all the singularities of $F(s)$ are to the left of the vertical line $\gamma - i\infty$, $\gamma + i\infty$.



The analytic Inverse Laplace Transform involves application of the well known theory of complex variables. For isolated singularities, the Bromwich contour is the standard approach.

2.1 The Selection of Algorithms for Numerical Approximation of the Inverse Laplace Transform

If there is no analytical inversion formula, numerical approach to the problem is required. This is done by using numerical Inverse Laplace Transform, denoted as $\hat{\mathrm{f}}(t)$.

Numerical inversion of Laplace transforms is not an easy task [20]. There neither exists an universal method of accurate inversion nor an inversion can be always conducted with satisfactory accuracy. Numerical inversion can be done using numerous distinct algorithms and only a few of them engage numerical integration of the contour. In the paper we evaluate for computational accuracy numerous methods representing all major lines of approach to the numerical inversion.

The Widder formula, mentioned in the introduction, required high-order analytic image function derivatives and it was thus impractical for computational purposes. The method of Stehfest is a discrete version of this formula. Another popular method used for numerical inversion is Fourier series expansion with accelerated convergence. There also have to be noted a modification of Stehfest finite differences evaluation method known as Gaver-Stehfest algorithm [21]. Bilinear Inverse Laplace Transform is possible by the use of Laguerre [22] and Legendre polynomials [23–25]. Delta function approximation by a finite linear combination of exponential functions was also used, as well as the approximation of a part of the Inverse Laplace Transform by a rational function. The method which follows closely the analytical approach to the numerical inversion is the use of deformed or fixed Bromwich contour integration using the Trapezoidal rule.

Next we present a briefly describe some of the methods.

2.1.1 Stehfest Method

The Stehfest inversion formula [26] is based on computing a sample of the time function by using finite differences and Salzer summation algorithm. The approximation assumes the following form:

$$\hat{\mathrm{f}}(t) = \left[s \cdot \sum_{n=1}^{N} K_n \cdot F(ns) \right]_{s=\ln(2/t)}$$

where the weight coefficients K_n are given by

$$K_n = (-1)^{n+N/2} \cdot \sum_{k=(n+1)/2}^{min(n,N/2)} \frac{k^{N/2}(2k)!}{(N/2-k)!k!(k-1)!(n-k)!(2k-n)!}.$$

The available parameters are the choice of N (which must be even) and the values of s at which $F(s)$ t-depended must be sampled.

2.1.2 Abate and Whitt Method

An algorithm involves Fourier series was originally proposed by Dubner and Abate [27] and improved by Abate and Whitt [28]. The algorithm employs Euler summation formula [29] (van Wijngaarden algorithm), which is shown to accelerate convergence of a slowly converging truncated Fourier series [30].

It is essentially the Trapezoidal rule applied to Laplace inversion formula. By applying the Trapezoidal rule with step size h and setting $h = \pi/(2t)$ and $a = A/(2t)$, we get

$$\hat{\mathrm{f}}(t) = \frac{e^{A/2}}{2t} Re\left(F\left(\frac{A}{2t}\right)\right) + \frac{e^{A/2}}{t} \sum_{k=1}^{\infty} (-1)^k Re\left(F\left(\frac{A+2k\pi}{2t}\right)\right).$$

The choice of A has to be made in such a way that a falls at the left of the real part of all singularities of the inverted Laplace transform.

2.1.3 De Hoog Method

It is an improved procedure for numerical inversion of Laplace transforms. It is based on accelerating the convergence of the Fourier series obtained from the inversion of integral by using non-linear double acceleration with Padé approximation and an analytic expression for the remainder in the series [31]. Non-linear acceleration techniques drastically reduce the required number of functions evaluations [32].

Speed of algorithm is primarily a function of the order of the Taylor series expansion, which is controlled manually.

2.1.4 FFT Method

The non-accelerated Fourier series inverse algorithm is almost useless because it requires thousands of F(s) evaluations. Practical approaches accelerate the convergence of the sum by using for example well known FFT (Fast Fourier Transform) algorithm [33].

The program inverts numerically a Laplace transform f(s) into f̂(t) using the Fast Fourier Transform (FFT) algorithm for a specific time t, an upper frequency limit ω, a real parameter σ and the number of integration intervals N.

Parameter σ contains singularities coordinates of F(s) and must be input by a user.

2.1.5 Vlach and Singhai Method

The basic inversion formula which works for real time functions assumes the following form [34]

$$\hat{\mathrm{f}}(t) = -\frac{1}{t} \sum_{i=1}^{M} K_i F\left(\frac{z_i}{t}\right),$$

where K_i are coefficients obtained as a result of Padée approximation of exponential part of the Laplace inversion formula.

2.2 Zakian Method

Zakian developed a method [35], in which approximation for $f(t)$ is given by

$$\hat{\mathrm{f}}(t) = \frac{1}{t} \sum_{i=1}^{N} K_i F\left(\frac{\alpha}{t}\right)$$

where sets of $\alpha_i K_i$ can be obtained as solution to

$$\sum_{i=1}^{N} \frac{K_i k!}{(\alpha_i)^{k+1}} = 1, \;\; k = 0, 1, 2, \ldots 2N - 1.$$

The only free parameter of choice is N. Its value should be 10 for best results.

2.2.1 Talbot Method

Talbot developed a method for the numerical inversion of the Laplace transform, in which the inversion is approximated by the Trapezoidal rule along a special deformed contour.

The Inverse Laplace Transform formula can be rewritten as

$$\hat{\mathrm{f}}(t) = \frac{1}{2\pi i} \oint_{\mathbf{B}} F(s) e^{st} ds,$$

where $\mathbf{B}$ is the Bromwich contour from $r - i\infty$ to $r + i\infty$ and $t > 0$, so that $\mathbf{B}$ is to the right of all singularities of $F(s)$.

Talbot provided [36] a recipe for a contour, in which the contour is moved to the left so as to reduce in magnitude the factor e^{st} in the integrand, but the contour must not be moved too close to singularities of $F(s)$, as to do so will result in peaks in the integrand function. This requires to be known the locations of the singularities of $F(s)$ [37].

Weideman [38] proposed an improved formula involving the cotangent function with additional parameters.

3 Some Aspects of Programming

3.1 Numerical Inversion Issues

The numerical Inverse Laplace Transform is an ill-posed problem by inherent sensitivity due to the multiplication by an exponential function of time:

- Finite precision and algorithmic errors can lead to the exponential divergence of numerical solutions,
- High precision variables are required for most Laplace Transform inversion methods: double precision variable can hold the values from the range $10^{-308} - 10^{308}$. Value $10^{308} \sim e^{709}$ is large enough for only modest t.

To counteract possible issues we apply some modern programming tools and techniques briefly described in the next two subsections.

3.2 Modern Programming Languages

C++ and Python are modern languages which have similar capabilities. However, Python achieves a higher abstract level than C++, i.e. an individual programmer can achieve the same results that in a much shorter time and with far fewer lines of code using it. It also has especially clean and straightforward syntax. It can lead to programs' shorter executing time and lower computational complexity.

Still, both languages' most respectful inventory in context of the goals of the following research are the arbitrary precision libraries: GNU GMP/MPFR for C++ and mpmath for Python.

3.3 Advantages of Arbitrary Precision Libraries

The GNU Multiple Precision Floating-Point Reliable Library (MPFR) [39] is an arbitrary precision package for C language and it is based on GNU Multiple-Precision Library (GMP) [40]. MPFR supports arbitrary precision floating point variables. It also provides proper rounding of all implemented operations and mathematical functions.

The mpmath [41] library is a free Python library for real and complex floating-point arithmetic with arbitrary precision.

The main advantages of the libraries are:

- Almost any calculation can be performed just as well at 10-digit or 1000-digit precision, with either real or complex numbers, and in many cases they implement efficient algorithms that scale well for extremely high precision work,
- Proper rounding in compliance with IEEE 754-2008 standard,
- A high number of special functions, with arbitrary precision and full support for complex numbers,
- Rudimentary support for interval arithmetic.

The libraries enable a user to set the precision of the arbitrary precision variables by specifying the number of bits to use in the mantissa of the floating point number. Due to the design of the libraries it is possible to work with any precision between 2 bits and maximum bits allowed for a computer.

The most common errors in numerical calculations are caused by wrong rounding. The libraries support proper rounding in compliance with the IEEE 754-2008 standard, as well as an additional one not included in it, i.e. round away from 0.

An increased computational time complexity for arbitrary precision calculations, which is their main disadvantage, is caused not only by applying variables of increased precision (more bits for mantissa in the floating point number), but also by applying the corresponding functions and different syntax, which is required. However, if an application is speed-depended - so called *wrapper*, as for example MPFR C++ [42] can be used. It enables using arbitrary precision in connection with standard C++ syntax. This simplifies programming and does not increases computational time complexity. For example, computational time complexity using this wrapper for up to 100-digits precision computing does not increase it more than it is in case of double precision application done on mid-2000 laptop.

4 Laplace Transforms Criteria Selection

We spent a large amount of time studying papers devoted to numerical Inverse Laplace Transform problem and solutions of fractional order differential equations. To be able to select aptly Laplace transforms which are useful in both areas, we consulted numerous noble mathematicians actively involved in FODE about the subject. As a result, we formed a list with Laplace transforms, which are the most frequently used and which represent various kinds of inversion problems. The list also included transforms which represent generally difficult computational problems.

5 Accuracy Criterion

We have considered various accuracy assessment criteria of numerically obtained results which are usually applied in this kind of research. They include: (a) absolute/relative error to obtain a real knowledge how close to an exact inversion an approximation really is, (b) quadratic mean and quadratic mean weighted by the factor e^{-t} and the difference between them which give root-mean-square deviation between the analytical and the numerical calculated values for small and large t and between them respectively, (c) order of convergence which informs how a method performs in respect to increasing number of sampling points, (d) time and memory complexity of computations which shows how a method occupies a computer with increasing amount of input data.

Due to the main goal of the numerical experiment, we selected relative error measure (a) for accuracy evaluation. It is commonly applied in similar to our comparisons. It actually can also be considered as (b), because it informs well how accurate to an exact inversion an approximation at each t is. However, due to the form of results presentation we selected method (a) because advantages of (b) can be utilized for results presented in tables.

The criterion (c) is abandoned due to the irrelevancy to the goals of the experiment. Order of convergence for each method can be found in source papers. They are presented in the references section.

The only drawback of (b) is an increasing value of relative error in case of decreasing values of exact inversion. It gives a false conclusion about an with time deteriorating accuracy. In such situation there is advised comparison of inversion and relative error plots.

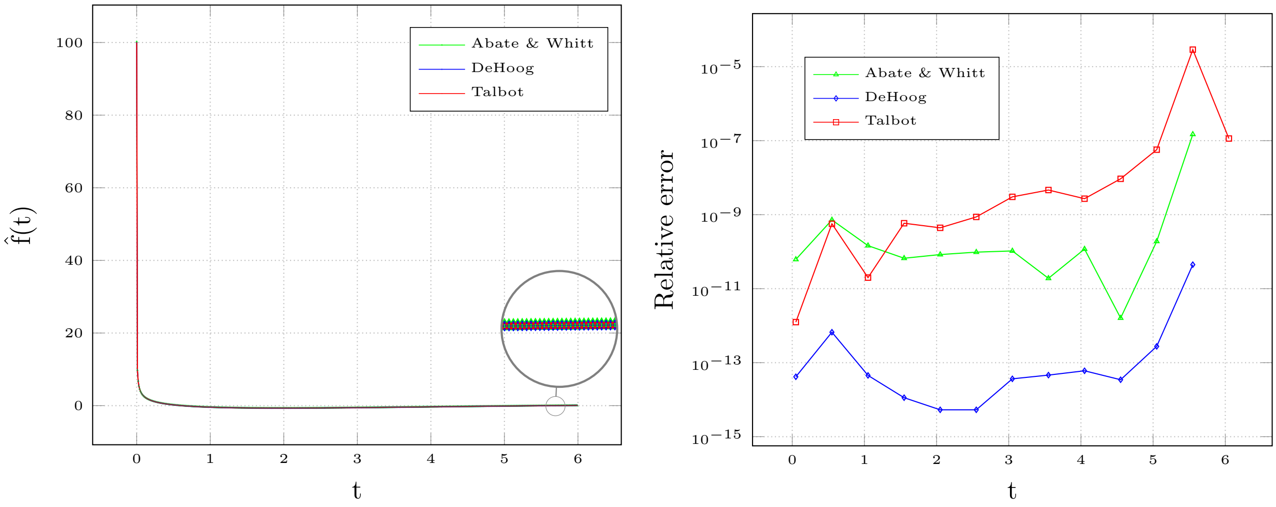
The next section presents accuracy comparison results in form of two plots for each selected Laplace transforms: the first plot of an actual approximation by applying a respective numerical inversion method and the second plot with relative error e_r of an approximation at each t for each of the methods v_c computed in respect to the values of an analytical inversion formula v_e

$$e_r = \left|1 - \frac{v_c}{v_e}\right|.$$

6 Results

- The Laplace transform and its inversion (1) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{\sqrt{\pi}\, e^{-1/s}}{\sqrt{s}}\right\} = \frac{\cos\left(2\sqrt{t}\right)}{\sqrt{t}},\ t \in (0,6\rangle\,. \tag{1}$$



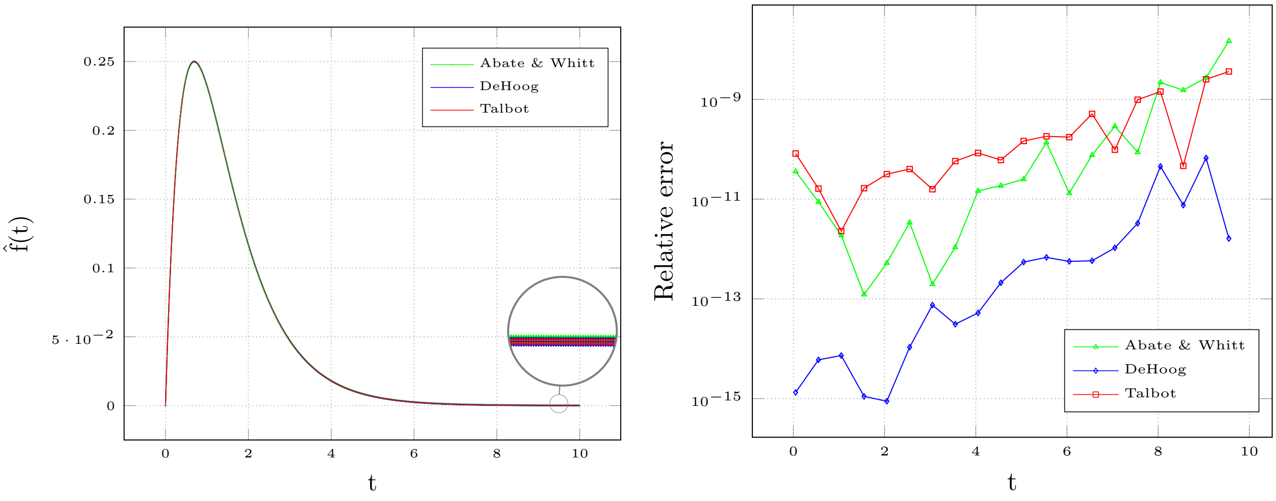
(a)

(b)

Fig. 1 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (1) (a) and their relative errors (b) for applied methods in interval $(0,10\rangle$.

- The Laplace transform and its inversion (2) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{1}{(s+1)(s+2)}\right\} = e^{-t} - e^{-2t},\ t \in (0,10\rangle\,. \tag{2}$$



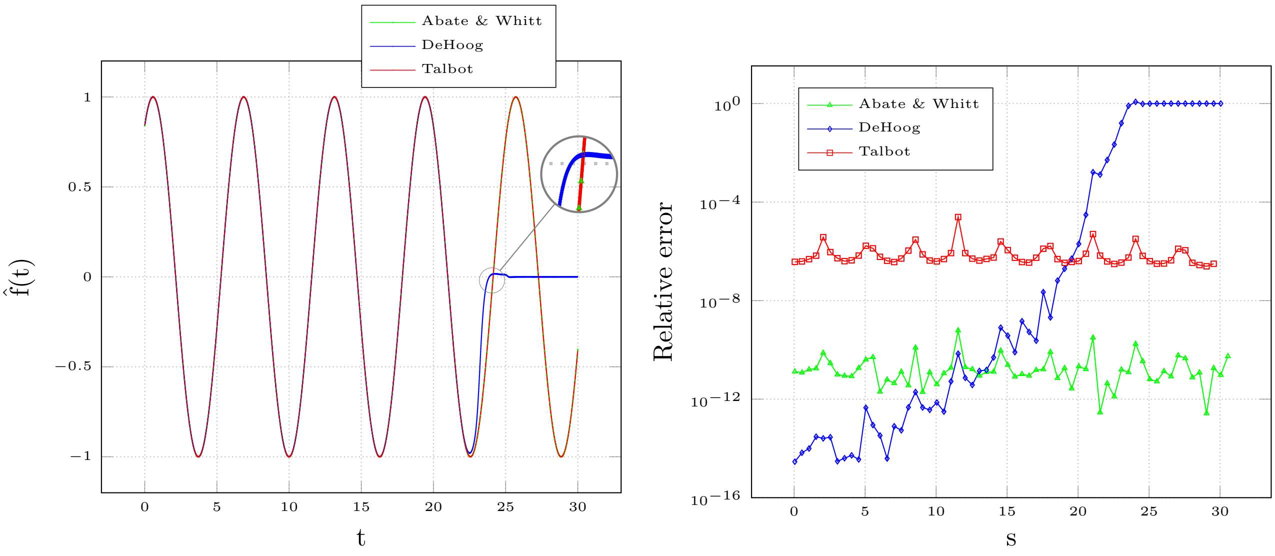
(a)

(b)

Fig. 2 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (2) (a) and their relative errors (b) for applied methods in interval $\langle 0, 30\rangle$.

- The Laplace transform and its inversion (3) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{\sin(b)\cdot s + a\cdot\cos(b)}{(s^2+a^2)}\right\} = \sin(a\cdot t + b)\,,\ a = 1, b = 1,\ t \in \langle 0, 10\rangle\,. \tag{3}$$



(a)

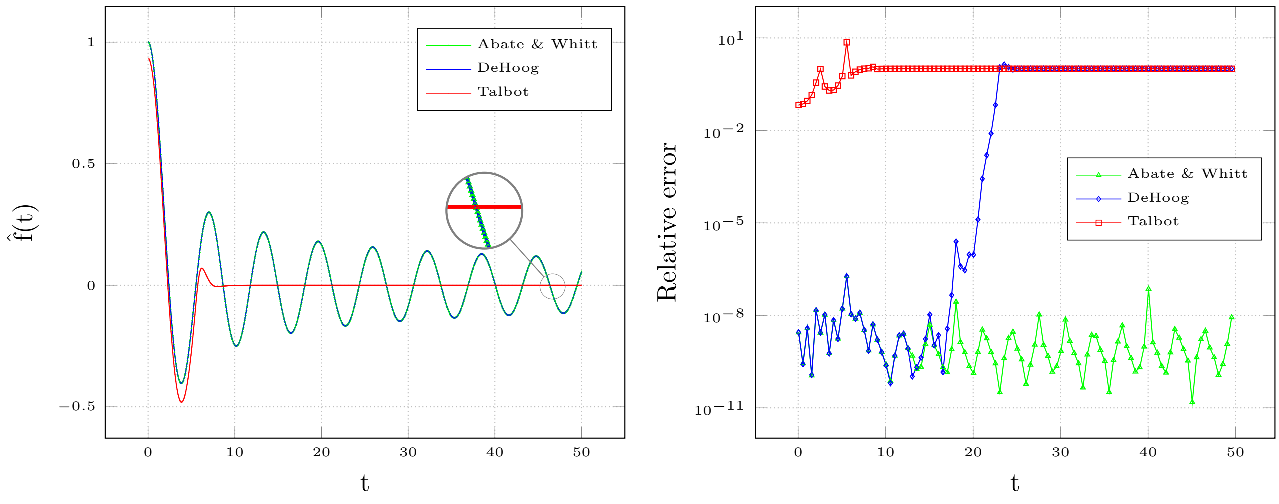
(b)

Fig. 3 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (3) (a) and their relative errors (b) for applied methods in interval $\langle 0, 30\rangle$.

- The Laplace transform and its inversion (4) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{1}{\sqrt{s^2+a^2}}\right\} = \mathscr{J}_0(at)\,,\ a = 1,\ t \in \langle 0, 50\rangle\,, \tag{4}$$

in which $\mathscr{J}_0(t)$ is the Bessel function of the first kind of order 0.



(a)

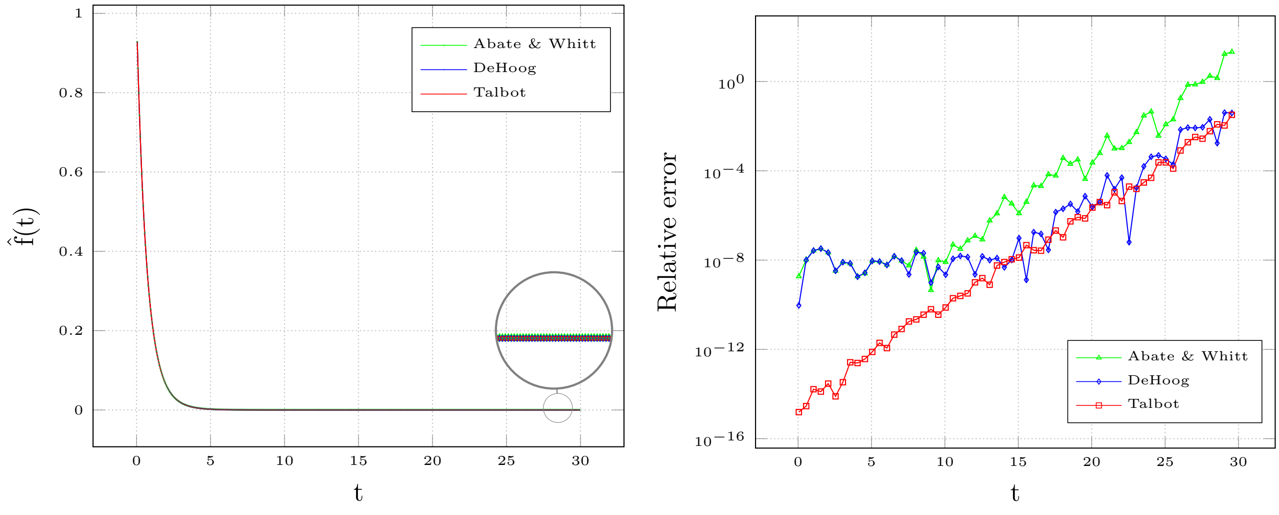
(b)

Fig. 4 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (4) (a) and their relative errors (b) for applied methods in interval $(0,50\rangle$.

- The Laplace transform and its inversion (5) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{1}{\sqrt{s+a}\sqrt{s+b}}\right\} = e^{-(a+b)t/2}\,\mathscr{I}_0\left(\frac{a-b}{2}\,t\right),\ a=2, b=1,\ t\in(0,30\rangle\,, \tag{5}$$

in which $\mathscr{I}_0(t)$ is modified Bessel function of order 0.



(a)

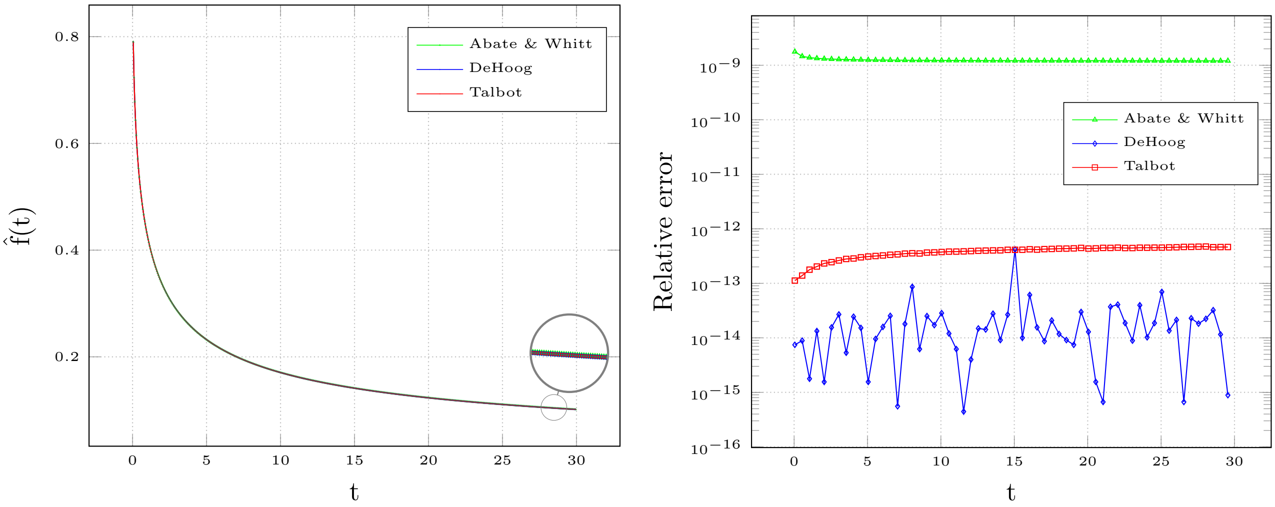
(b)

Fig. 5 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (5) (a) and their relative errors (b) for applied methods in interval $(0,30\rangle$.

- The Laplace transform and its inversion (6) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{1}{\sqrt{s}\cdot\left(\sqrt{s}+a\right)}\right\} = e^{a^2 t}\cdot erfc\left(a\sqrt{t}\right),\ a=1,\ t\in\left(0,30\right\rangle, \tag{6}$$

in which $erfc$ is the complementary error function, $1-erf(x)$.



(a)

(b)

Fig. 6 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (6) (a) and their relative errors (b) for applied methods in interval $\left(0,30\right\rangle$.

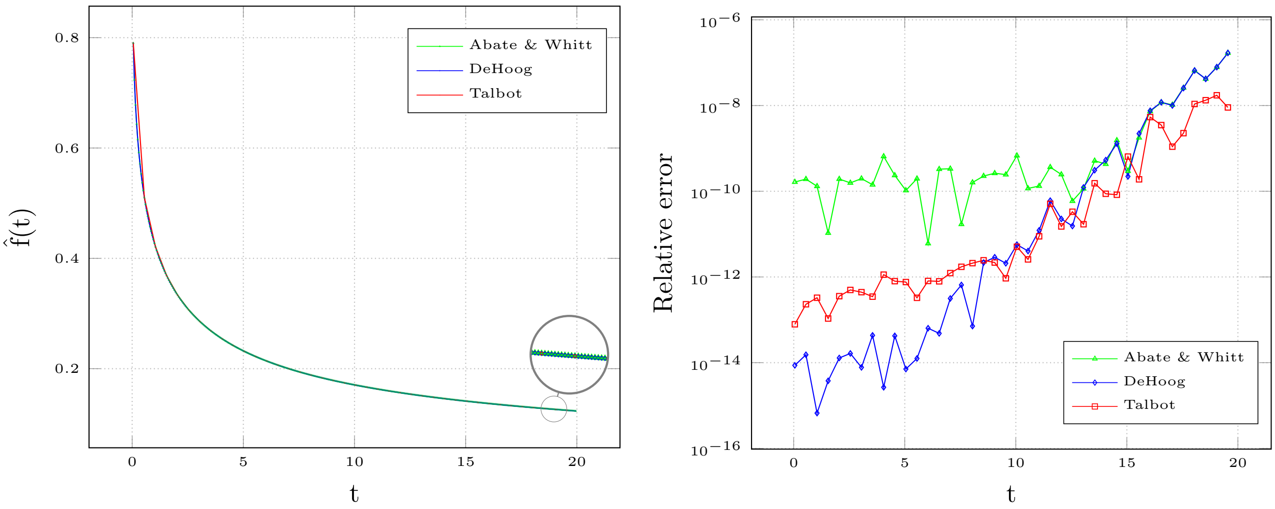
- The Laplace transform and its inversion (7) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{s^{\alpha-1}}{s^{\alpha}+\lambda}\right\} = \mathscr{E}_{\alpha}\left(-\lambda t^{\alpha}\right),\ \alpha=0.5, \lambda=1,\ t\in\left(0,20\right\rangle \tag{7}$$

in which

$$\mathscr{E}_{\alpha}(z) = \sum_{k=0}^{\infty}\frac{z^k}{\Gamma(\alpha k+1)}, \mathscr{R}(\alpha)>0, z\in\mathscr{C}$$

is the Mittag-Leffler function with one parameter [43].



(a)

(b)

Fig. 7 Plots of numerical inversions $\hat{f}(t)$ of the Laplace transform (7) (a) and their relative errors (b) for applied methods in interval $(0,30\rangle$.

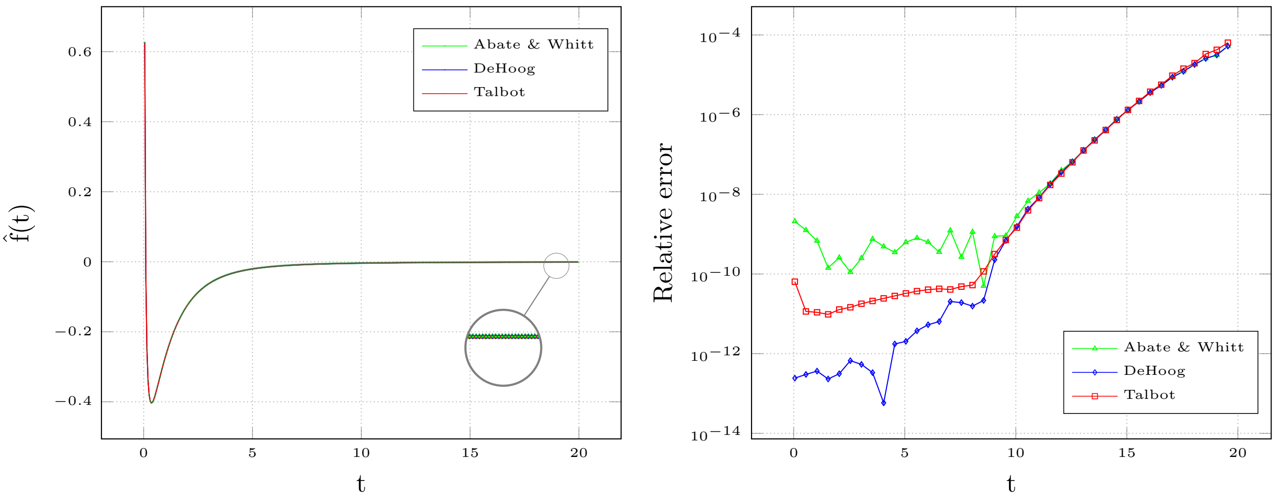
- The Laplace transform and its inversion (8) is

$$\hat{f}(t)=\mathscr{L}^{-1}\left\{\frac{s^{\alpha-\beta}}{s^{\alpha}+\lambda}\right\}=t^{\beta-1}\mathscr{E}_{\alpha,\beta}\left(-\lambda t^{\alpha}\right),\ \alpha=0.9,\beta=0.1,\lambda=1,\ t\in(0,20\rangle\,, \tag{8}$$

in which

$$\mathscr{E}_{\alpha,\beta}\left(z\right)=\sum_{k=0}^{\infty}\frac{z^{k}}{\Gamma\left(\beta+\alpha k\right)},\mathscr{R}\left(\alpha\right)>0,\mathscr{R}\left(\beta\right)>0,z\in\mathscr{C}$$

is generalized the Mittag-Leffler function with two parameters [44].



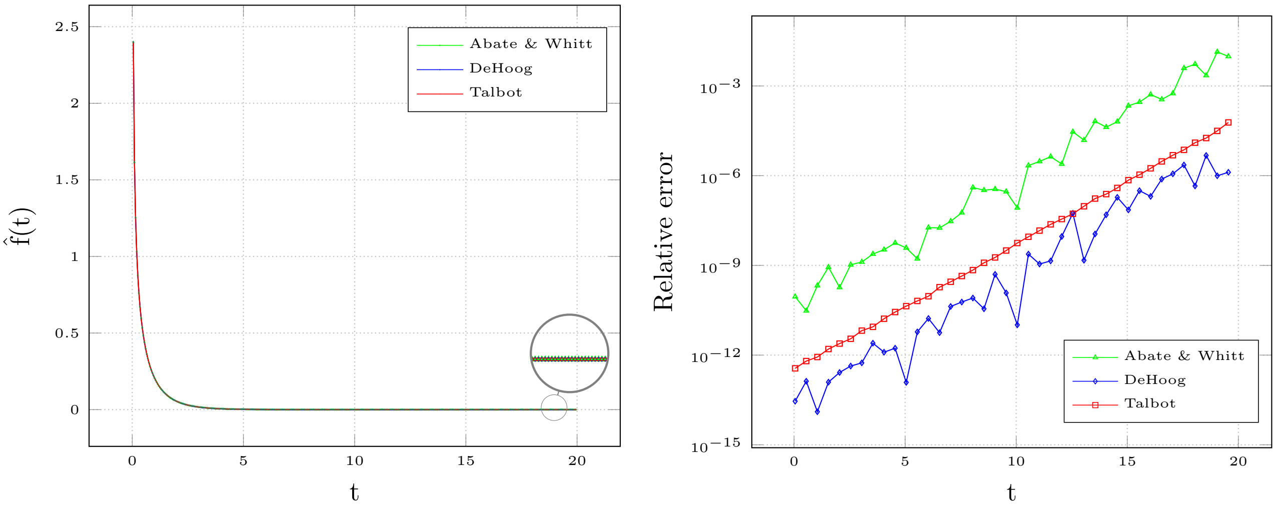
(a)

(b)

Fig. 8 Plots of numerical inversions $\hat{f}(t)$ of the Laplace transform (8) (a) and their relative errors (b) for applied methods in interval $(0,30\rangle$.

- The Laplace transform and its inversion (9) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{s^{\alpha-1}}{s^{\alpha}+\lambda}\right\} = \frac{t^{\alpha-1}}{\Gamma(\alpha)}e^{-\lambda t},\ \alpha = 0.5, \lambda = 1,\ t \in (0,20\rangle\,. \tag{9}$$



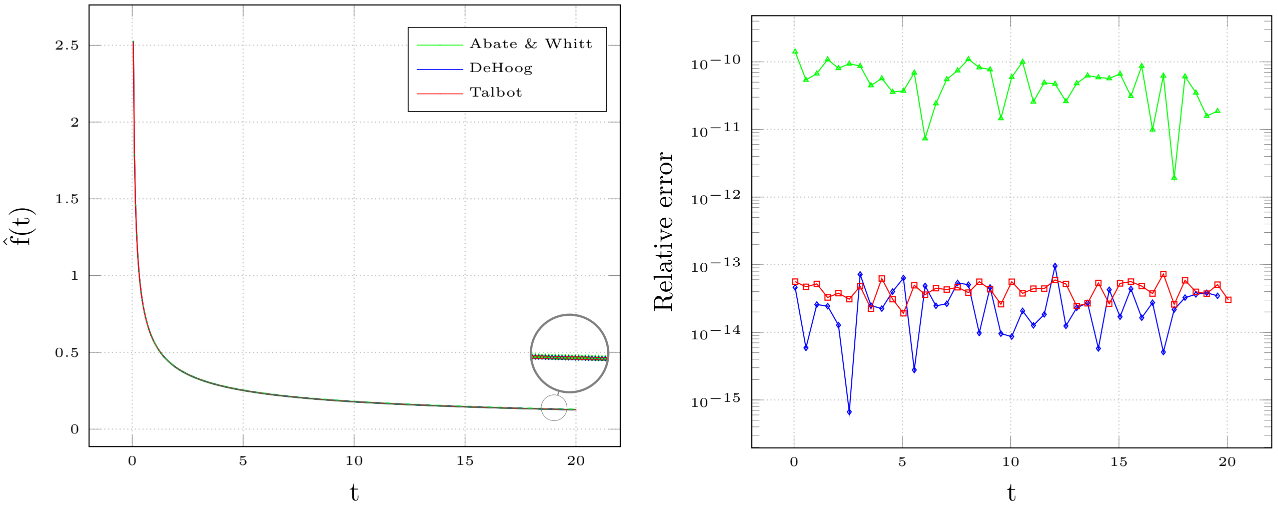
(a)

(b)

Fig. 9 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (9) (a) and their relative errors (b) for applied methods in interval $(0,30\rangle$.

- The Laplace transform and its inversion (10) is

$$\hat{\mathrm{f}}(t) = \mathscr{L}^{-1}\left\{\frac{1}{s^{\alpha}}\right\} = \frac{t^{\alpha-1}}{\Gamma(\alpha)},\ \alpha = 0.5,\ t \in (0,20\rangle\,. \tag{10}$$



(a)

(b)

Fig. 10 Plots of numerical inversions $\hat{\mathrm{f}}(t)$ of the Laplace transform (10) (a) and their relative errors (b) for applied methods in interval $(0,20\rangle$.

7 Conclusions

FC is an extension of classical calculus of integer order. This fact enables finding solutions of a problem within the available numerical methods.

The solution of the problem selected in the paper focuses on the selection and the programming of seven algorithms of NumILPT. The methods represent four main approaches to the numerical approximation of the Inverse Laplace Transform. They were evaluated against a test set of inversion problems frequently used in FC and representing difficult computational issues.

Based on our previous experience and presently obtained results, we have selected three numerical algorithms of the numerical approximation of the Inverse Laplace Transform for the actual comparison: the Abate & Whitt, the DeHoog and the Talbot method for their versatility and high-accuracy inversions. Due to the general application requirement and clarity of presentation for the present evaluation, remaining methods were left aside. Their performance is either mediocre or they calculated satisfactory accurate inversions in individual cases only. It suggests their application only on an particular inversion problem basis.

NumILPT can be applied as a high-accuracy method for obtaining solutions of FODE. During the evaluation, several test problems have correctly been inverted by applying more than one algorithm. Here, the Talbot method, which uses the Trapezoidal Rule for the integration of the cotangent function deformed Bromwich contour, is the most accurate and versatile. However, it requires precise input of Laplace transform poles coordinates for high-accuracy inversions, which can be difficult and can not be automated. For the best results, manual calculation and input of the coordinates is advised.

By applying the DeHoog algorithm, which uses Fourier series expansion integration, one can obtain equally accurate inversions. Wherein, the poles coordinates input is not required.

The Abate and Whitt, another algorithm, which employs the use of Fourier series expansion approximation as well, is advised for inverting Laplace transforms, whose originals are the periodic functions. It is characterized by high correctness of inverted solutions even for $t > 50$.

Difficult implementation issues were successfully solved by the use of arbitrary precision mathematical libraries and concurrent of two programming languages: C++ and Python for programming.

The commonly used standard double precision computer arithmetic has many flaws and limitations, which include the limitations of the values, the double precision variables can hold, no influence of a user on the method of mathematical operations rounding or precision of the intermediate variables. The decision in these cases is made on an operating systems or compiler level.

The state of the art technology called *the infinite precision* computing [45] enables eliminating most of these problems by defining of arbitrary precision variables, allowing a user to access the intermediate mathematical operations rounding modes by breaking them down into basic ones and the capability of a user to explicitly choose one of five rounding modes for each mathematical operation.

For speed-depended applications, the so called wrapper MPFR C++ has been used. It enables the use of arbitrary precision variables for programming in connection with the standard C++ syntax. Therefore, its application has all of the advantages associated with the infinite precision computing and no speed punishment for the computations involving the precision up to 100 digits.

Acknowledgement

The work was created as a result of the research project no. 2016/23/D/ST6/01709 financed from the funds of the National Science Center, Poland.

References

[1] K. Oldham and J. Spanier. *The Fractional Calculus. Theory and Applications of Differentiation and Integration to Arbitrary Order*. Academic Press, INC, San Diego Ca, 1974.

[2] K. S. Miller and B. Ross. *An Introduction to the Fractional Calculus and Fractional Differential Equations*. John Wiley & Sons, Inc., New York, NY, 1993.

[3] S. Samko, A. Kilbas, and O. Marichev. *Fractional Integrals and Derivatives: Theory and Applications*. Gordon and Breach, London, 1993.

[4] I. Podlubny. *Fractional Differential Equations*. Academic Press, INC, San Diego Ca, 1999.

[5] K. Diethlem. *The Analysis of Fractional Differential Equations. An Application-Oriented Exposition Using Differential Operators of Caputo Type*. Springer-Verlag, Berlin, Heidelberg, 2010.

[6] D. Baleanu, K. Diethlem, E. Scalas, and J.J. Trujillo. *Fractional Calculus. Models and Numerical Methods*. World Scientific, Singapore, 2012.

[7] A. A. Kilbas, M. Srivastava H, and J.J. Trujillo. *Theory and Applications of Fractional Differential Equations*. Elsevier Science, 2006.

[8] F. Mainardi. *Fractional Calculus and Waves in Linear Viscoelasticity*. Imperial Collage Press, London, 2010.

[9] Y. Povstenko. *Fractional Thermoelasticity*. Springer International Publishing, Cham, Heidelberg, New York, Dodrecht, London, 2015.

[10] Y. Povstenko. *Linear Fractional Diffusion-Wave Equation for Scientists and Engineers*. Birkhäuser, Springer, Cham, Heidelberg, New York, Dodrecht, London, 2015.

[11] M. O'Flynn and E. Moriarty. *Time Domain and Transform Analysis*. Wiley & Sons, Inc., 1987.

[12] A. D. Poularikas and S. Seely. *Laplace Transforms in The Transforms and Applications Handbook: Second Edition*. Boca Raton: CRC Press LLC, 2000.

[13] J. Abate, G.L. Choudhurry, and W.Whitt. An introduction to numerical inversion and its application to probability models. In W. Grassman, editor, *Computational Probability*, pages 257–323. Kluwer, Boston, 1999.

[14] S. I. Kabanikhin. Definition and examples of inverse and ill-posed problems. a survey paper. *J.Inv.Ill-Posed Problems*, 16:317–357, 2008.

[15] R. Piessens. A bibliography on numerical inversion of the laplace transform and applications. *Journal of Computational and Applied Mathematics*, 1(2):115–128, 1975.

[16] D. V. Widder. *The Laplace Transform*. Princeton Unviersity Press, 1946.

[17] D. W. Brzeziński and P. Ostalczyk. Numerical calculations accuracy comparison of the inverse laplace transform algorithms for solutions of fractional order differential equations. *Nonlinear Dynamics*, 81(1):65–77, 2016.

[18] D. Valerio, J. J. Trujillo, M. Rivero, J. A. T. Machado, and D. Baleanu. Fractional calculus: A survey of useful formulas. *The European Physical Journal Special Topics*, 222:1827–1846, 2013.

[19] Y. Q. Chen, I. Petras, and B. Vinagre. A list of laplace and inverse laplace transforms related to fractional order calculus, 2010. http://www.tuke.sk/petras/foc_laplace.pdf.

[20] Ch. L. Epstein and J. Schotland. The bad truth about laplace's transform. *SIAM Review*, 3:504–520, 2008.

[21] H. Villinger. Solving cylindrical geothermal problems using gaver-stehfest inverse laplace transform. *Geophysics*, 50(10):47–49, 1985.

[22] W. Weeks. Numerical inversion of laplace transforms using laguerre functions. *Journal of the ACM*, 13(3):419–429, 1966.

[23] R. Piessens. New quadrature formulas for the numerical inversion of the laplace transform. *BIT*, 9:351–361, 1969.

[24] P. D. Iseger. Numerical transform inversion using gaussian quadrature. *Probability in the Engineering and Informational Sciences*, 20:1–44, 2006.

[25] B. Davies. Integral transforms inversion using gaussian quadratures. *Journal of Computational Physics*, 33(1):1–32, 1979.

[26] H. Stehfest. Algorithm 368: Numerical inversion of laplace transforms. *Communications of the ACM*, 13(1):47–49, 1970.

[27] H. Dubner and J. Abate. Numerical inversion of laplace transforms by relating them to the finite fourier cosine transform. *Journal of the ACM*, 15:115–123, 1968.

[28] J. Abate and W. Whitt. The fourier-series method for inverting transforms of probability distributions. *Queueing Systems*, 10(5):5–87, 1999.

[29] C. A. O'Cinneide. Euler summation for fourier series and laplace transform inversion. *Stochastic Models*, 13(2):315–337, 1997.

[30] F. Durbin. Numerical inversion of laplace transforms: an efficient improvment to dubner and abate's method. *The Computer Journal*, 17(4):371–376, 1973.

[31] F. R. DeHoog, J. H. Knight, and A. N. Stokes. An improved method for numerical inversion of laplace transforms. *SIAM J. Sci. Stat. Comput.*, 3(3):357–366, 1982.

[32] K. L. Kuhlman. Review of inverse laplace transform algorithms for laplace-space approaches. *Numer Algor*, pages

1–19, 2012.

[33] A. M. Cohen. *Numerical Methods for Laplace Transform Inversion*. Springer-Verlag, Berlin, Heidelberg, 2007.

[34] J. Vlach and K. Singhai. *Computer Methods for Circuit Analysis and Design*. Van Nostrand Rheinhold Company, 1983.

[35] V. Zakian. Solution of homogeneous ordinary linear differential systems by numerical inversion of laplace transforms. *Electronic Letters*, 7:546–548, 1971.

[36] A. Talbot. The accurate numerical inversion of laplace transforms. *IMA Journal of Applied Mathematics*, 23(1):97–112, 1979.

[37] A. Murli and M. Rizzardi. Algorithm 682: Talbot's method for the laplace inversion. *ACM Transactions on Mathematical Software*, 16(2):158–168, 1990.

[38] J. A. C. Weideman. Optimizing talbot's contours for the inversion of the laplace transform. *SIAM J. Anal.*, 44(6):2342–2362, 2006.

[39] Guillaume Hanrot et al. *mpfr: The MPFR library for multiple-precision floating-point computations with correct rounding. (version 3.13)*, 2015. `http://www.mpfr.org/`.

[40] Torbjörn Granlund et al. *gmp: GMP is a free library for arbitrary precision arithmetic (version 6.0.0a)*, 2015. `https://gmplib.org/`.

[41] Fredrik Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.19)*, June 2014. `http://mpmath.org/`.

[42] Pavel Holoborodko. *High-performance C++ interface for MPFR library (version 3.6.2)*, 2015. `http://www.holoborodko.com/pavel/mpfr/`.

[43] P. Humbert and R. P. Agarwal. Sur la fonction de mittag-leffler et quelques-unes de ses géneéralisations. *Bull. Sci. Math. Ser. II*, 77:180–185, 1953.

[44] K. M. Kowankar and A. D. Gangal. Fractional differentability of nowhere differentable functions and dimensions. *CHAOS*, 6(4):180–185, 1996.

[45] K. R. Ghazi, V. Lefevre, P. Theveny, and P. Zimmermann. Why and how to use arbitrary precision. *IEEE Computer Society*, 12(3):1–5, 2001.