Manuel D. Ortigueira, António M. Lopes* and José Tenreiro Machado

On the Numerical Computation of the Mittag–Leffler Function

https://doi.org/10.1515/ijnsns-2018-0358
Received November 22, 2018; accepted June 07, 2019

Abstract: The Mittag–Leffler function (MLF) plays an important role in many applications of fractional calculus, establishing a connection between exponential and power law behaviors that characterize integer and fractional order phenomena, respectively. Nevertheless, the numerical computation of the MLF poses problems both of accuracy and convergence. In this paper, we study the calculation of the 2-parameter MLF by using polynomial computation and integral formulas. For the particular cases having Laplace transform (LT) the method relies on the inversion of the LT using the fast Fourier transform. Experiments with two other available methods compare also the computational time and accuracy. The 3-parameter MLF and its calculation are also considered.

Keywords: Mittag–Leffler function, numerical computation, fast Fourier transform

MSC® (2010). Primary 26A33; Secondary 65R10

1 Introduction

The Mittag–Leffler function (MLF) [1–4] is ubiquitous both in the theory of fractional calculus (FC) and its applications [5–16]. In fact, the impulse response of fractional linear systems can be expressed in terms of the MLF, which entails the need of efficient algorithms for its computation [17]. However, this calculation is not trivial, excepting when considering small values of the argument, since the numerical computation of the MLF poses problems both of accuracy and convergence [1, 3, 18–21].

***Corresponding author: António M. Lopes,** Faculty of Engineering, University of Porto, Rua Dr. Roberto Frias, Porto 4200–465, Portugal, E-mail: aml@fe.up.pt
Manuel D. Ortigueira, Center for Technology and Systems – UNINOVA and DEE, NOVA School of Science and Technology of NOVA University of Lisbon, Portugal, E-mail: mdo@fct.unl.pt
José Tenreiro Machado, Dept. of Electrical Engineering, Institute of Engineering, Polytechnic of Porto, Rua Dr. Antonio Bernardino de Almeida, 431, Porto 4249-015, Portugal, E-mail: jtm@isep.ipp.pt
http://orcid.org/0000-0003-4274-4879

In this paper, we tackle this problem. We start from the general integral formula for the 2-parameter MLF [1, 22, 23] and we develop a method for its numerical implementation. We then particularize the general formulation for values on half-straight lines, which are useful in its application to linear systems. In a second step, we introduce a generalization of the MLF and we propose one method based on the efficient computation of the inverse Laplace transform (LT). We adopt the bilinear transformation to follow a $\mathcal{Z}$ transform form and we use the fast Fourier transform (FFT) to obtain a sampled version of the generalized Mittag–Leffler fuction (GMLF). Finally, based on these ideas, we consider also the implementation of the 3-parameter GMLF.

The paper is organized as follows. Section 2 introduces the 1- and 2-parameter MLF and their integral representations. Section 3 addresses distinct methods for computing the MLF, presents several numerical tests, and discusses the main results. Section 4 generalizes the algorithms to the 3-parameters MLF. Finally, Section 5 draws the mains conclusions.

2 The MLF and its integral representation

Definition 1. *The (1-parameter) MLF [6] is given by:*

$$E_\alpha(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha+1)}, \quad z \in \mathbb{C}, \quad \alpha \in \mathbb{R}^+. \tag{1}$$

The exponential series is a particular case obtained when $\alpha = 1$.

A generalization of (1) includes a second parameter, yielding the 2-parameter MLF:

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(k\alpha+\beta)}, \quad z \in \mathbb{C}, \quad \alpha, \beta \in \mathbb{R}^+. \tag{2}$$

When α and β are positive real (case that we will consider here), the series converges for all values of $z \in \mathbb{C}$ [1, 22, 23] and has the integral representation that we will deduce here.

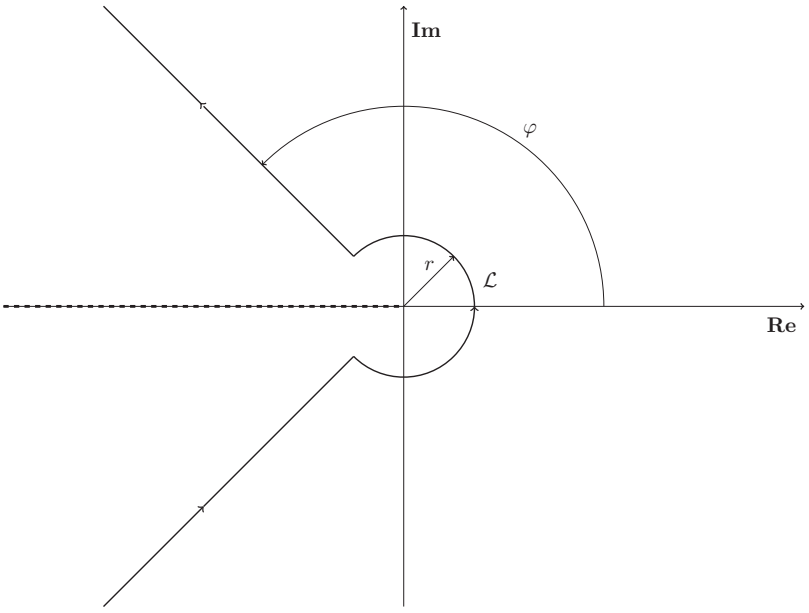

Figure 1: The Integration path, $\mathcal{L}$, for $\frac{1}{\Gamma(z)}$.

Consider the integration path in Figure 1. As shown by Hankel [24], the reciprocal gamma function can be computed by the integral

$$\frac{1}{\Gamma(z)} = \frac{1}{2\pi i}\int_{\mathcal{L}} u^{-z}e^{u}du, \quad z \in \mathbb{C}, \tag{3}$$

provided that:

- u^{-z} is defined by its principal branch; we can assume for branchcut line the negative real half axis, unless a pole exists on this line;
- r may assume any positive real value;
- φ can be any value in the interval $(\pi/2, \pi)$; it can be equal to π, provided that $\arg(u) = \pi$, or $\arg(u) = -\pi$, above, or below, the cut, respectively.

As the MLF is defined by a uniformly convergent series, we can write

$$\begin{aligned} E_{\alpha,\beta}(z) &= \frac{1}{2\pi i}\sum_{k=0}^{\infty} z^k \int_{\mathcal{L}} u^{-k\alpha-\beta}e^{u}du \\ &= \frac{1}{2\pi i}\int_{\mathcal{L}} u^{-\beta}\left[\sum_{k=0}^{\infty}(zu^{-\alpha})^k\right]e^{u}du. \end{aligned}$$

The summation of the inner geometric series is $\sum_{k=0}^{\infty}(zu^{-\alpha})^k = \frac{u^{\alpha}}{u^{\alpha}-z}$, provided that $|zu^{-\alpha}| < 1$. This condition introduces a constraint in the integration loop in such a way that we can formulate the general integral form of the MLF, [1, 19, 22, 25]:

$$E_{\alpha,\beta}(z) = \frac{1}{2\pi i}\int_{\mathcal{L}} \frac{u^{\alpha-\beta}e^{u}}{u^{\alpha}-z}du, \quad z \in \mathbb{C}, \quad \alpha,\beta \in \mathbb{R}^{+}, \tag{4}$$

where now $\mathcal{L}$ is the integration path consisting in a loop that starts at $u = -\infty$, encircles the disk with center $u = 0$ and radius $|z|^{1/\alpha}$, and returns to $u = -\infty$ (in Figure 1, $r > |z|^{1/\alpha}$). This integration path can be deformed to become the Hankel contour, simplifying the computations, under the assumptions:

- The negative real axis is the branchcut line, but we can choose any other in the left half plane; as referred above, we have to follow that strategy if there is a pole on such axis;
- As the MLF of any order $\alpha > 1$ can be written in terms of the MLF with order $\alpha < 1$, we will consider $0 < \alpha \le 1$;
- We address the cases $\alpha - \beta - 1 > 0$. Other alternatives [1], namely $\alpha < 0$, can also be treated, but they are not relevant, and are not considered;
- $z \in \mathbb{C} \setminus 0$.

Consider the integration path in Figure 2 denoted by $\mathcal{L}_t$. It consists of three components:

- The above described path $\mathcal{L}$;
- The Hankel path $\mathcal{H}$ consisting of two half-straight lines joined by a half circle with a radius as small as we wish;
- Two circular arcs with radius $R > |z|^{1/\alpha}$, to make a closed path; radius R grows to ∞.

With the residue theorem, we can write

$$\oint_{\mathcal{L}_t} \frac{u^{\alpha-\beta}e^{u}}{u^{\alpha}-z}du = \begin{cases} \frac{2\pi i}{\alpha} z^{\frac{1-\beta}{\alpha}} e^{z^{\frac{1}{\alpha}}} & |\arg(z)| \le \pi\alpha \\ 0 & |\arg(z)| > \pi\alpha \end{cases}. \tag{5}$$

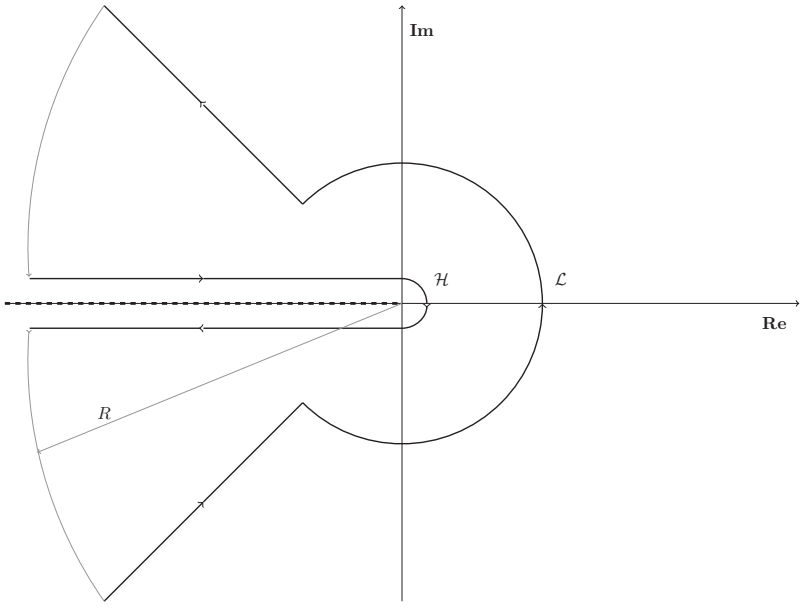

Figure 2: The paths $\mathcal{L}$ and $\mathcal{H}$, and the closing arcs.

Under the above stated conditions, we can deduce that:

- The integrals along the circular arcs tend to zero when the radius goes to infinite (application of the Jordan's lemma);
- The integral along the half-circle in the Hankel path becomes zero when its radius decrease to zero;
- The integral along the upper half-straight line is $\int_{0^+}^{\infty} \frac{u^{\alpha-\beta}e^{i(\alpha-\beta)\pi}}{u^{\alpha}e^{i\alpha\pi}-z}e^{-u}du$;
- The integral over the lower half-straight line is $-\int_{0^+}^{\infty} \frac{u^{\alpha-\beta}e^{-i(\alpha-\beta)\pi}}{u^{\alpha}e^{-i\alpha\pi}-z}e^{-u}du$;
- The MLF is given by

$$E_{\alpha,\beta}(z) = z^{\frac{1-\beta}{\alpha}}\frac{e^{z^{\frac{1}{\alpha}}}}{\alpha} + \frac{1}{\pi}\int_{0^+}^{\infty} \frac{\sin(\beta\pi)u^{2\alpha-\beta} + \sin[(\alpha-\beta)\pi]u^{\alpha-\beta}z}{u^{2\alpha} - 2\cos(\alpha\pi)u^{\alpha}z + z^2}e^{-u}du, \quad (6)$$

if $|\arg(z)| < \pi\alpha$, or by

$$E_{\alpha,\beta}(z) = \frac{1}{\pi}\int_{0^+}^{\infty} \frac{\sin(\beta\pi)u^{2\alpha-\beta} + \sin[(\alpha-\beta)\pi]u^{\alpha-\beta}z}{u^{2\alpha} - 2\cos(\alpha\pi)u^{\alpha}z + z^2}e^{-u}du, \quad (7)$$

if $|\arg(z)| > \pi\alpha$.

Remark 1. *For* $\arg(z) = \pm\alpha\pi$ *we have a singularity in the integral, leading to inaccurate results. We will analyze and solve this problem in Section 3.*

In numerical computations, we need to truncate the integral above some u_0. This originates an error given by:

$$\delta = \frac{1}{\pi}\int_{u_o}^{\infty} \frac{\sin(\beta\pi)u^{2\alpha-\beta} + \sin[(\alpha-\beta)\pi]u^{\alpha-\beta}z}{u^{2\alpha} - 2\cos(\alpha\pi)u^{\alpha}z + z^2}e^{-u}du.$$

It can be shown that the function $\frac{\sin(\beta\pi)u^{2\alpha}+\sin(\alpha\pi)u^{\alpha-\beta}z}{u^{2\alpha}-2\cos(\alpha\pi)u^{\alpha}z+z^2}$ is bounded and approaches $\sin(\beta\pi)$ when u increases without bound. Assume that its absolute value is given by $\sin(\beta\pi)(1+\varepsilon)$, $\varepsilon > 0$. Then

$$|\delta| \le \frac{(1+\varepsilon)}{\pi}\int_{u_o}^{\infty}\frac{1}{u^{\beta}}e^{-u}du < \frac{(1+\varepsilon)}{u_0^{\beta}\pi}\int_{u_o}^{\infty}e^{-u}du = \frac{(1+\varepsilon)}{u_0^{\beta}\pi}e^{-u_0}. \quad (8)$$

Therefore, the error decreases exponentially leading to a fast convergence of the integral.

In FC applications, we need to compute the MLF on half-straight lines, defined by $z = pt^{\alpha}\varepsilon(t)$. From eq. (2) we obtain the expression:

$$\mathcal{E}_{\alpha,\beta}(pt^{\alpha}) = \sum_{k=0}^{\infty} p^k \frac{t^{k\alpha}}{\Gamma(k\alpha+\beta)}\varepsilon(t), \quad p \in \mathbb{C}, \quad t \in \mathbb{R}, \quad (9)$$

where t represents time and $\varepsilon(t)$ denotes the Heaviside unit step function used to state that eq. (9) is identically null for $t < 0$.

To avoid confusion we call expression (9) as "causal MLF" (CMLF), since it is defined on $\mathbb{R}$ and is null on $\mathbb{R}^-$ (it can be considered as the impulse response of a causal system). Furthermore, in order to simplify notation, we will

omit $\varepsilon(t)$. This variable change is inserted in eqs. (6) and (7), where we make $v = u/t$ to get

$$\mathcal{E}_{\alpha,\beta}(pt^\alpha) = p^{\frac{1-\beta}{\alpha}}\frac{e^{p^{\frac{1}{\alpha}}t}}{\alpha} + \frac{t^{1-\beta}}{\pi}\int_{0^+}^{\infty}\frac{\sin(\beta\pi)v^{2\alpha-\beta}+\sin[(\alpha-\beta)\pi]v^{\alpha-\beta}p}{v^{2\alpha}-2\cos(\alpha\pi)pv^\alpha+p^2}e^{-vt}dv, \quad (10)$$

if $|\arg(p)| \le \pi\alpha$, or

$$\mathcal{E}_{\alpha,\beta}(pt^\alpha) = \frac{t^{1-\beta}}{\pi}\int_{0^+}^{\infty}\frac{\sin(\beta\pi)v^{2\alpha-\beta}+\sin[(\alpha-\beta)\pi]v^{\alpha-\beta}p}{v^{2\alpha}-2\cos(\alpha\pi)pv^\alpha+p^2}e^{-vt}dv, \quad (11)$$

if $|\arg(p)| > \pi\alpha$.

In practical applications, it is interesting to deal with the LT. If $\beta = 1$ it is a simple task to show that

$$\mathcal{L}\left[\mathcal{E}_{\alpha,1}(pt^\alpha)\right] = \frac{s^{\alpha-1}}{s^\alpha - p}, \quad (12)$$

for $\mathrm{Re}(s) > 0$ and $|ps^{-\alpha}| < 1$ [6]. Given the analytical properties of the LT, these conditions can be expressed as $\mathrm{Re}(s) > \max\left\{\mathrm{Re}(p^{1/\alpha}), 0\right\}$. If $\beta \neq 1$, the function in eq. (9) has a LT with no simple closed form. However, from eqs. (9), (10), and (11) the GMLF

$$c_{\alpha,\beta}(t) = t^{\beta-1}\mathcal{E}_{\alpha,\beta}(pt^\alpha), \quad t \in \mathbb{R}^+, \quad (13)$$

has LT given by:

$$\mathcal{L}\left[c_{\alpha,\beta}(t)\right] = C_{\alpha,\beta}(s) = \frac{s^{\alpha-\beta}}{s^\alpha - p}, \quad \alpha,\beta \in \mathbb{R}^+, \quad (14)$$

for $\mathrm{Re}(s) > \max\left\{\mathrm{Re}(p^{1/\alpha}), 0\right\}$.

Frequently we know the LT in (14) that must be inverted [17]. In agreement with the Bromwich inversion theorem, we compute the integral along a vertical straight line passing at $\mathrm{Re}(s) = \sigma > \max\left\{\mathrm{Re}(p^{1/\alpha}), 0\right\}$. In this line of thought, we apply the LT inverse theorem to obtain a function that will be multiplied by $e^{\sigma t}$; our working domain is a vertical line defined by $s = \sigma + i\omega$, where ω represents the angular frequency in rad/s. With such change of variable we obtain the Fourier transform (FT):

$$C_{\alpha,\beta}(i\omega) = \frac{(\sigma+i\omega)^{\alpha-\beta}}{(\sigma+i\omega)^\alpha - p}. \quad (15)$$

The inverse FT of this function is given by

$$c_{\alpha,\beta}(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty}\frac{(\sigma+i\omega)^{\alpha-\beta}}{(\sigma+i\omega)^\alpha - p}e^{i\omega t}d\omega. \quad (16)$$

The computation of this integral may reveal some difficulties, because the absolute value of the integrand decreases to zero when ω increases, but the decrease is very slow, implying a very high truncation value. To avoid this problem we can use the bilinear transformation:

$$s = \frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}, \quad z \in \mathbb{C}, \quad (17)$$

to obtain, from eq. (14), a function defined in the context of the $\mathcal{Z}$ transform [26]:

$$C_d(z) = \frac{\left[\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right]^{\alpha-\beta}}{\left[\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right]^{\alpha} - p}, \quad |z| > 1, \quad (18)$$

where the subscript d means that we are referring to a transfer function of a discrete-time system.

Remark 2. *The sampling time interval T must be small. In the integer order applications we choose T to guarantee that* $\frac{e^{\frac{2+pT}{2-pT}}}{e^{pT}} \approx 1$ *[26] for any* $p \in \mathbb{C}$. *In practice, the value* $T = 10^{-3}$ *s is adequate. For the fractional case* $0 < \alpha \le 1$ *it is expected to be small, suggesting the value* $T = 10^{-4}$ *s.*

The time-domain function corresponding to eq. (18) is given by:

$$c_d(nT) = \frac{1}{2\pi i}\oint_\lambda C_d(z)z^{n-1}\mathrm{d}z, \quad (19)$$

where λ is a circle with center at the origin and radius greater than 1. With a change of variable $z = e^{\sigma+i\omega} = ae^{i\omega}$, $a = e^\sigma$ we obtain:

$$c_d(nT) = \frac{1}{2\pi i}\int_{-\pi}^{\pi} C_d(e^{i\omega})e^{i\omega n}\mathrm{d}\omega, \quad (20)$$

or, as $H_d(e^{i\omega})$ is a 2π-periodic function, we can write

$$c_d(nT) = a^n\frac{1}{2\pi i}\int_0^{2\pi} C_d(e^{i\omega})e^{i\omega n}\mathrm{d}\omega. \quad (21)$$

Adopting $\omega = 2\pi\nu$ expression (21) yields:

$$c_d(nT) = \int_0^1 C_d(e^{i2\pi\nu})e^{i2\pi\nu n}\mathrm{d}\nu. \quad (22)$$

This integral is suitable for numerical computation by means of the FFT. Let us sample the integrand on a discrete uniform grid with $K = 2^M$ points, $\nu_k = \frac{1}{K}\cdot k$, $k = 0, 1, \ldots, K-1$, to obtain the discrete Fourier transform:

$$c_d(nT) \approx \frac{1}{K}\sum_{k=0}^{K-1} C_d(e^{i\frac{2\pi}{K}k})e^{i\frac{2\pi}{K}kn}. \tag{23}$$

The value of K must be chosen in agreement with desired highest value for the time interval: $t_{mx} = KT$.

To obtain the final estimation values for the CMLF with this approach we must multiply (23) by the exponential $e^{\sigma nT}$. On the other hand, the relation between the continuous-time and the corresponding discrete-time values is obtained by dividing by the sampling interval, T. Therefore, the final CMLF is given by:

$$c_d(nT) = c_{\alpha,\beta}(nT) \approx e^{\sigma nT}\frac{1}{KT}\sum_{k=0}^{K-1} C_d(e^{i\frac{2\pi}{K}k})e^{i\frac{2\pi}{K}kn}. \tag{24}$$

3 The numerical computation of the MLF

Several techniques were proposed for computing the MLF [2, 18–20, 25]. In general, they use the Taylor series approach for small values of the magnitude of the argument, $|z|$, and the properties of integral representations and asymptotic behavior for high values of $|z|$.

The main idea is to avoid the problems involved in the direct computation of the series (2). To do the computation we proceed as follows:

(i) Truncate the series to get a polynomial

$$E_{\alpha,\beta,N}(z) = \sum_{k=0}^{N}\frac{z^k}{\Gamma(k\alpha+\beta)},$$

where N is chosen to assure that $\left|\frac{z^N}{\Gamma(N\alpha+\beta)}\right| < \eta$, and η a small value (we use $\eta = 10^{-10}$).

(ii) Compute the terms $c_k = \frac{1}{\Gamma(k\alpha+\beta)}$.

For fixed values of α and β, when k increases, the same happens to $\Gamma(k\alpha+\beta)$. Although depending on α, we can say that in most cases, for $k > 200$ this function becomes ∞ in the numerical representation. We propose here a simple and robust method for calculating the terms c_k. Therefore, for a given set of values k and $k\alpha+\beta$, we calculate the integer and fractional parts of $k\alpha+\beta$, denoted by P and b, respectively, so that $P = \lfloor k\alpha+\beta \rfloor$ and $b = k\alpha+\beta-P$. Knowing that $\Gamma(x+m) = (x)_m\Gamma(x)$, where $(x)_m$ is the Pochhammer factorial, $x \in \mathbb{R}$, $m \in \mathbb{N}$, we can compute the terms c_k by means of the expression

$$c_k = \frac{1}{\Gamma(b+1)}\prod_{j=0}^{P-2}\left(\frac{1}{b+j}\right). \tag{25}$$

Experiments with different values of α and β reveal that the proposed approach is more robust when compared with the calculation of the MLF directly form expression (2). We used $\Gamma(b+1)$ instead of $\Gamma(b)$ to avoid the case $b = 0$.

(iii) Perform polynomial computation.

To avoid computing large powers of z we use the well-known Horner algorithm [24]. This consists basically in a recursive computation as described in the follow-up.

To simplify, write $X_N(z) = E_{\alpha,\beta,N}(z)$ in the format

$$X_N(z) = (((c_N z + c_{N-1})z + c_{N-2})z\cdots c_1)z + c_0. \tag{26}$$

This representation suggests the following recursive procedure for its computation. In each step, we carry out a product and an add. Let us start the algorithm with $n = N$,

$$X_{-1}(z) = c_N z + c_{N-1}.$$

For $n = N-1, \cdots, 1$, we make

$$X_{N-1-n}(z) = X_{N-2-n}(z)z + c_{n-1}.$$

As seen, we carry N multiplications and N adds, but we do not need to compute the powers of z.

The algorithm described is effective for small values of $|z|$. On the other hand, the computation of the MLF (2) and the CMLF (9) by means of the integral formulations (6)–(7) and (10)–(11), respectively, yields good results for high values of $|z|$, but reveals difficulties for small values of the argument. Therefore, herein we combine both approaches. The procedure is the following:

(i) Consider 2 positive real values: $x_1 < x_2$. For $|z| \le x_1$ we use the above polynomial-based approximation to compute the MLF. We denote by $f_1(z)$ the obtained function, and we use it to approximate the MLF for $|z| \le x_1$.

(ii) For $|z| \ge x_2$ we use the integral formulations (6)–(7). As above, we denote this function by $f_2(z)$, and use it to approximate the MLF for $|z| \ge x_2$.

(iii) For $x_1 < |z| < x_2$ we use a weighted average of both methods, $f_1(z)$ and $f_2(z)$:

a. Define two real weight functions $w_1(z) = 1 - \frac{|z|-x_1}{x_2-x_1}$, for $x_1 \le |z| \le x_2$ and $w_2(z) = \frac{|z|-x_1}{x_2-x_1}$ for $x_1 \le |z| \le x_2$, such that $w_1(z) + w_2(z) = 1$, for $x_1 \le |z| \le x_2$.
b. In the interval $x_1 < |z| < x_2$, the MLF is approximated by the weighted average:

$$f(z) = w_1(z)f_1(z) + w_2 f_2(z). \tag{27}$$

This is important to avoid jumps.

In the follow-up we compute the MLF (1)–(2), the CMLF (9) and the GMLF (13) for different values of their parameters (Table 1). All of the experiments are performed in Matlab R2015a, under Windows 7, 64-bit, on a Intel(R) Core(TM) i5 CPU @ 3.20 GHz computer. The results obtained with the proposed methods and those generated by the P&K-MLF [27] and Gar-MLF [28] algorithms are compared in terms of accuracy and computational time. For the P&K-MLF approach we use precision 10^{-8}. For the Gar-MLF we use the default settings. As a reference for accuracy we adopt the values calculated with the software Mathematica (Mat-MLF).

In the first experiment, C_1, we calculate the MLF by means of the polynomial and general integral formulation (Int-MLF), given by (6)–(7), (27). The integral is computed through the Matlab function `integral`. The parameters are $(\alpha, \beta) = (0.6, 1)$, and the domain are the circumferences defined by $|z| \in [1, 1.1]$, $\arg(z) \in [-\pi, \pi]$. Figure 3 depicts the results in the complex plane. We verify that the Int-MLF and Gar-MLF lead to accurate results (close to those obtained with the Mat-MLF), while the P&K-MLF reveals difficulties for certain values of z, namely when $\arg(z) = \pm\alpha\pi$.

Remark 3. *For* $\arg(z) = \pm\alpha\pi$ *we have one pole on the branchcut line. In this case, for avoiding inaccurate results with the Int-MLF method, we should replace* u *by* $u \cdot e^{i\theta}$, $|\theta| < \frac{\pi}{2}$, *in the integral equations* (6)–(7), *to move the branchcut from* π *towards* $\pi - \theta$.

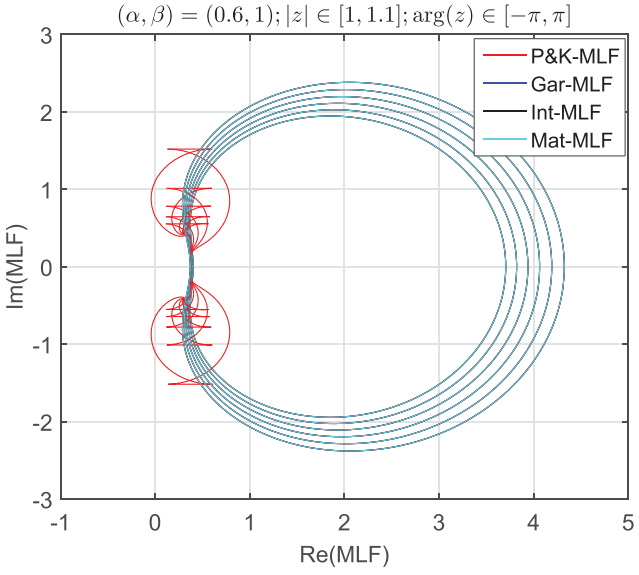

Figure 3: The value of the MLF calculated with the P&K-MLF, Gar-MLF, Int-MLF and Mat-MLF algorithms, for $(\alpha, \beta) = (0.6, 1)$, $|z| \in [1, 1.1]$, and $\arg(z) \in [-\pi, \pi]$.

In the two next experiments $\{C_2, C_3\}$ we assess the methods in terms of accuracy and computational time. The domain is the straightline defined by $|z| \in]0, 100]$, $\arg(z) = \alpha\pi$. For C_2 the parameters are $(\alpha, \beta) = (0.5, 1)$, while for C_3 we have $(\alpha, \beta) = (0.5, 0.4)$. Figure 4 depicts the error $E = |\tilde{E}_{\alpha,\beta} - E_{\alpha,\beta}|$, where $\tilde{E}_{\alpha,\beta}(z)$ and $E_{\alpha,\beta}(z)$ denote the values calculated by the numerical methods and those obtained with Mathematica, respectively. Occasional gaps in the error curve correspond to $E = 0$. Figure 5 shows the computational time, t_C, required for the numerical evaluation of the MLF with the different algorithms.

Table 1: Set of experiments illustrating the methods proposed.

Label	Function	Parameters	Domain	Method
C_1	MLF (1)–(2)	$(\alpha, \beta) = (0.6, 1)$	$\lvert z\rvert = 1$, $\arg(z) \in [1, 1.1]$	Int-MLF (6)–(7), (27)
C_2	MLF (1)–(2)	$(\alpha, \beta) = (0.5, 1)$	$\lvert z\rvert \in]0, 100]$, $\arg(z) = \alpha\pi$	Int-MLF (6)–(7), (27)
C_3	MLF (1)–(2)	$(\alpha, \beta) = (0.5, 0.4)$	$\lvert z\rvert \in]0, 100]$, $\arg(z) = \alpha\pi$	Int-MLF (6)–(7), (27)
C_4	MLF (1)–(2)	$(\alpha, \beta) = (0.7, 1)$	$\lvert z\rvert \in]0, 100]$, $\arg(z) = \alpha\frac{\pi}{2}$	Int-MLF (6)–(7), (27)
C_5	MLF (1)–(2)	$(\alpha, \beta) = (1.2, 0.7)$	$\lvert z\rvert \in]0, 100]$, $\arg(z) = \alpha\frac{\pi}{2}$	Int-MLF (6)–(7), (27)
C_6	CMLF (9)	$(\alpha, \beta) = (0.7, 0.9)$	$t \in]0, 20]$, $p = -1$	Int-CMLF (10)–(11), (27)
C_7	CMLF (9)	$(\alpha, \beta) = (0.7, 0.9)$	$t \in]0, 20]$, $p = -1 + i$	Int-CMLF (10)–(11), (27)
C_8	GMLF (13)	$(\alpha, \beta) = (0.7, 1)$	$t \in]0, 20]$, $p = -1$	FFT-GMLF (24)
C_9	GMLF (13)	$(\alpha, \beta) = (0.7, 1)$	$t \in]0, 20]$, $p = -1 + i$	FFT-GMLF (24)

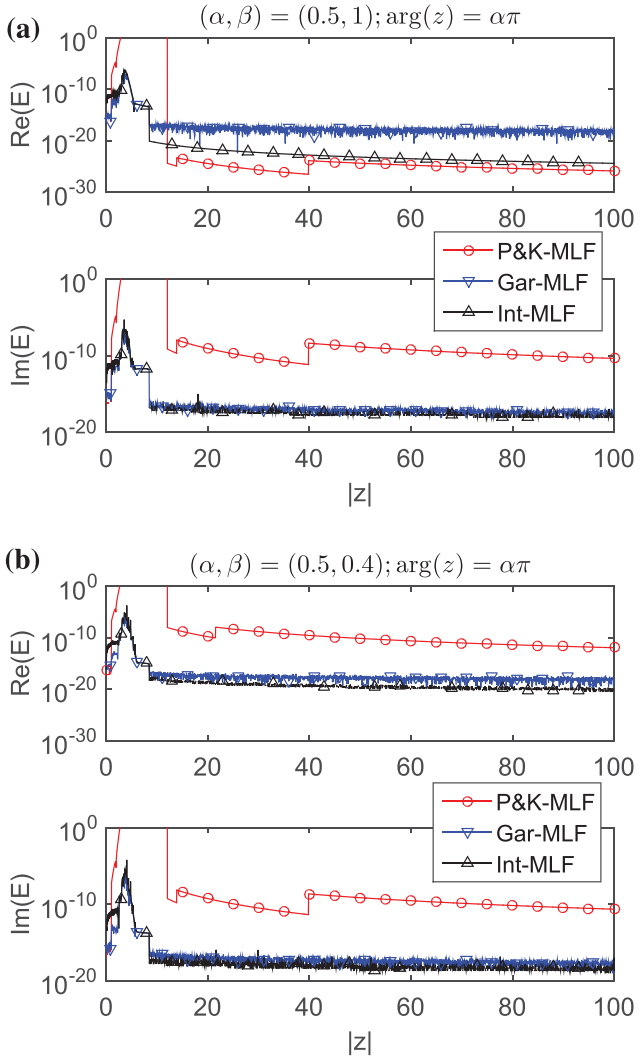

Figure 4: The error, E, in the numerical evaluation of the MLF for the cases (a) C_2: $(\alpha,\beta)=(0.5,1)$, $|z|\in]0,100]$, $\arg(z)=\alpha\pi$; (b) C_3: $(\alpha,\beta)=(0.5,0.4)$, $|z|\in]0,100]$, $\arg(z)=\alpha\pi$.

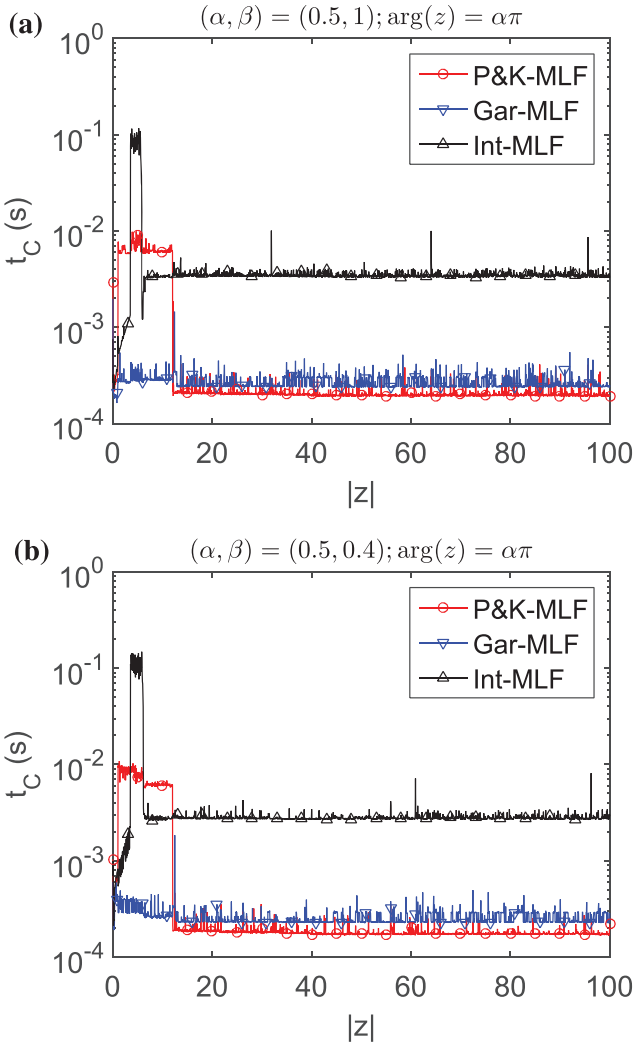

Figure 5: The computational time, t_C, to evaluate the MLF for the cases (a) C_2: $(\alpha,\beta)=(0.5,1)$, $|z|\in]0,100]$, $\arg(z)=\alpha\pi$; (b) C_3: $(\alpha,\beta)=(0.5,0.4)$, $|z|\in]0,100]$, $\arg(z)=\alpha\pi$.

We observe that the Int-MLF leads to more accurate results than the Gar-MLF and P&K-MLF algorithms, particularly for larger values of $|z|$. For $|z|$ small, the P&K-MLF diverges. In terms of computational time, we verify that t_C is smaller, and remains nearly constant, for the Gar-MLF algorithm. Both for the Int-MLF and P&K-MLF, the computational time is non-uniform, varying more than one order of magnitude. In general, the Gar-MLF method requires more time than its counterparts for evaluating the MLF.

Experiments $\{C_4, C_5\}$ assess the methods in another demanding domain, defined by $|z| \in]0,100]$, $\arg(z) = \alpha\frac{\pi}{2}$. The parameters are $(\alpha,\beta) = (0.7,1)$, and $(\alpha,\beta) = (1.2,0.7)$, for C_4 and C_5, respectively. Figures 6 and 7 depict the error, E, and the computational time, t_C, obtained with the different algorithms. We observe that the Int-MLF and the Gar-MLF yield similar accuracy and are better than the P&K-MLF algorithm. Moreover, the Gar-MLF is more efficient in terms of computational time.

In the experiments $\{C_6, C_7\}$ we calculate the CMLF. We use the integral formulation (Int-CMLF) given in the formulas (10)–(11). For both experiments the parameters are $(\alpha,\beta) = (0.7,0.9)$, and $t \in]0,20]$. In C_6 we use $p = -1$, while in C_5 we adopt $p = -1+i$. Figure 8 shows the error, $E = |\tilde{\mathcal{E}}_{\alpha,\beta} - \mathcal{E}_{\alpha,\beta}|$, with $\tilde{\mathcal{E}}_{\alpha,\beta}$ and $\mathcal{E}_{\alpha,\beta}$ denoting the values calculated numerically and obtained with Mathematica, respectively. Figure 9 depicts the computational time, t_C.

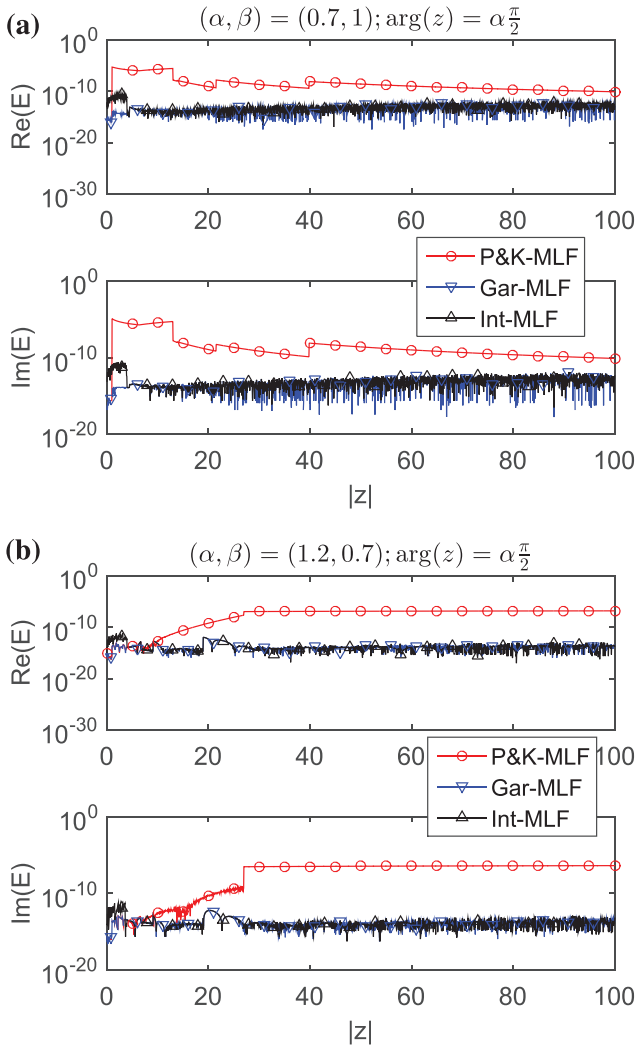

Figure 6: The error, E, in the numerical evaluation of the MLF for the cases (a) C_4: $(\alpha,\beta)=(0.7,1)$, $|z|\in]0,100]$, $\arg(z)=\alpha\frac{\pi}{2}$; (b) C_5: $(\alpha,\beta)=(1.2,0.7)$, $|z|\in]0,100]$, $\arg(z)=\alpha\frac{\pi}{2}$.

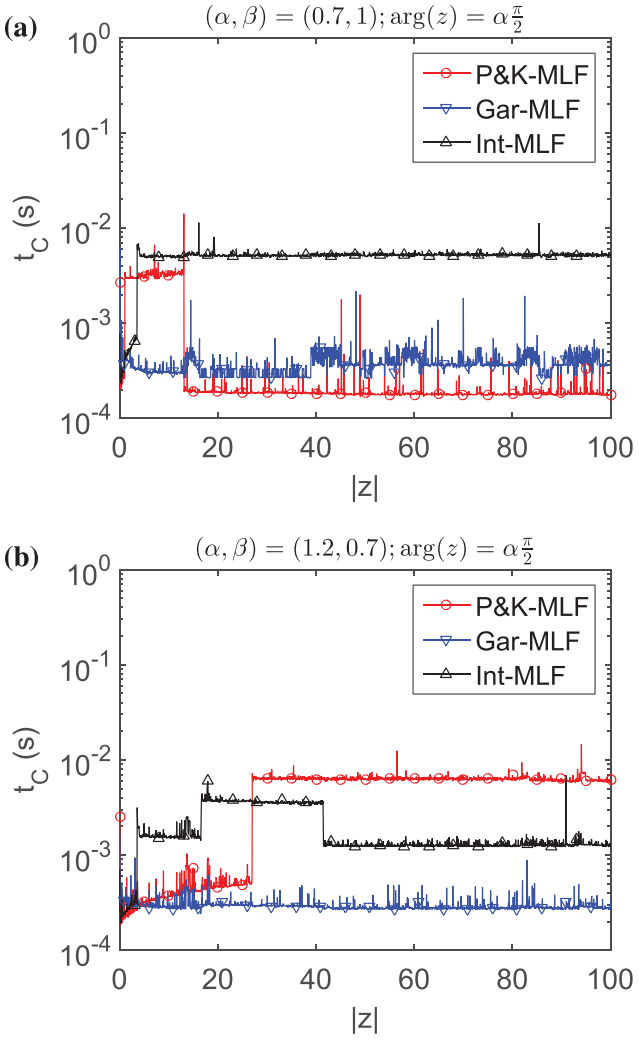

Figure 7: The computational time, t_C, to evaluate the MLF for the cases (a) C_4: $(\alpha,\beta)=(0.7,1)$, $|z|\in]0,100]$, $\arg(z)=\alpha\frac{\pi}{2}$; (b) C_5: $(\alpha,\beta)=(1.2,0.7)$, $|z|\in]0,100]$, $\arg(z)=\alpha\frac{\pi}{2}$.

The Int-CMLF and Gar-MLF lead to identical values of E, while for the P&K-MLF the error is considerable larger. The computational time, t_C, is smaller for the Gar-MLF algorithm.

In the examples $\{C_8, C_9\}$, we calculate the GMLF for $t\in]0,20]$ and parameters $(\alpha,\beta)=(0.7,1)$. Such value for β allows comparing our method based on the inversion of the LT using the FFT (FFT-GMLF), given by (24), with the Int-CMLF, the P&K-MLF and the Gar-MLF algorithms. For the experiments C_8 and C_9 we adopt $p=-1$ and $p=-1+i$, respectively. Figures 10 and 11 depict the error, E, and the computational time, t_C. Regarding accuracy, the FFT-GMLF algorithm performs slightly better than the P&K-MLF, and worse than the Int-CMLF and Gar-MLF. With respect to computational time, we observe that FFT-GMLF is considerably slower than the alternative methods.

In conclusion, we verify that for evaluating the MLF and the CMLF, the proposed methods, namely the Int-MLF and the Int-CMLF, yield results identical to those obtained with the Gar-MLF, and perform better than the P&K-MLF algorithm. In terms of computational time, the Gar-MLF algorithm superior to the other methods. For calculating the GMLF, the FFT-GMLF is more accurate than the P&K-MLF, requiring however a larger computational effort. Therefore, the proposed methods represent alternatives to those already available, and can be implemented easily.

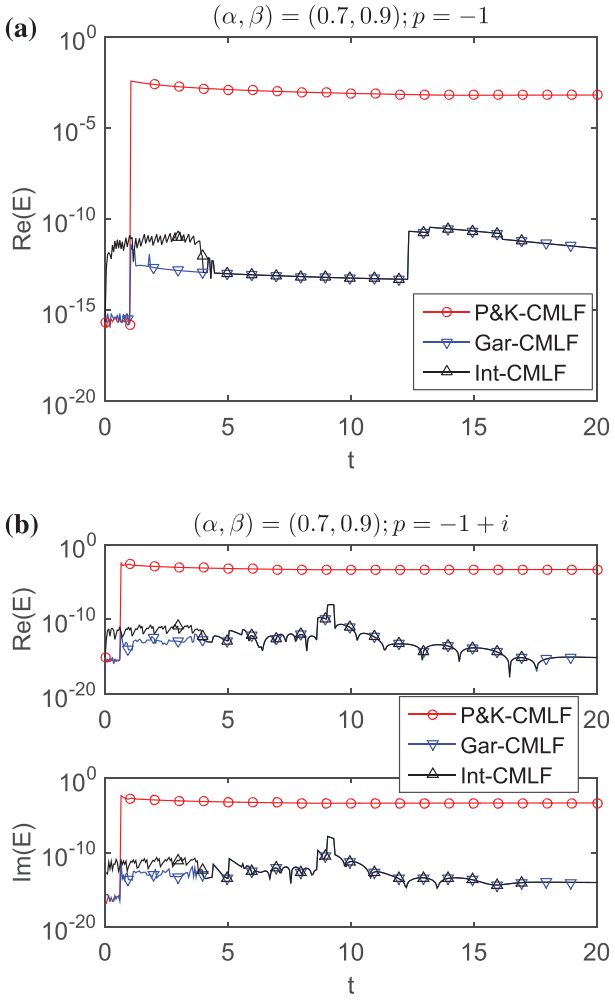

Figure 8: The error, E, in the numerical evaluation of the CMLF for the cases (a) C_6: $(\alpha,\beta)=(0.7,0.9)$, $t\in]0,20]$, $p=-1$; (b) C_7: $(\alpha,\beta)=(0.7,0.9)$, $t\in]0,20]$, $p=-1+i$.

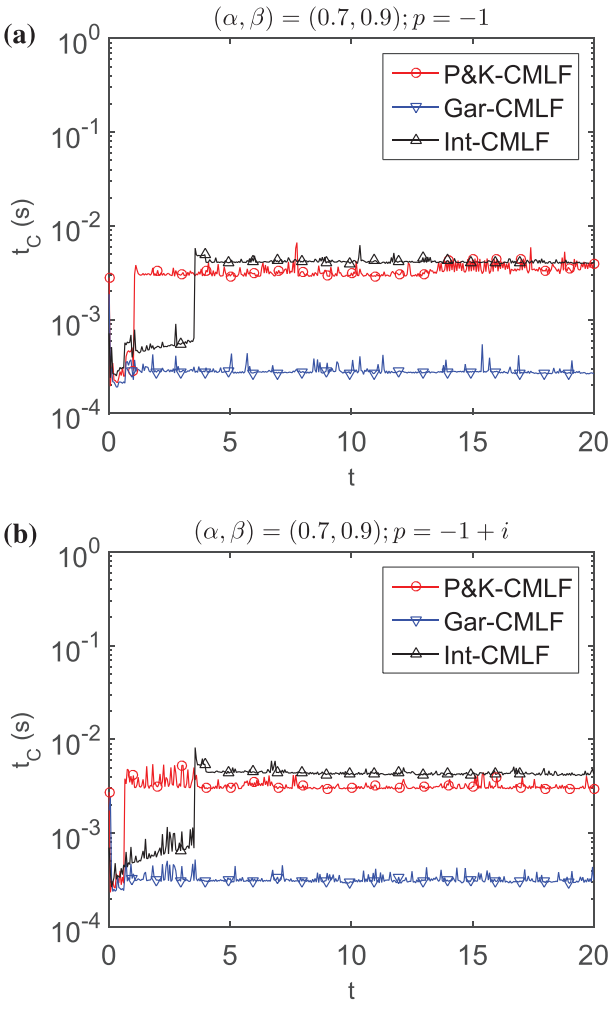

Figure 9: The computational time, t_C, to evaluate the CMLF for the cases (a) C_6: $(\alpha,\beta)=(0.7,0.9)$, $t\in]0,20]$, $p=-1$; (b) C_7: $(\alpha,\beta)=(0.7,0.9)$, $t\in]0,20]$, $p=-1+i$.

4 Computation of the 3-parameter CMLF

The 3-parameter GMLF is defined by:

$$c_{\alpha,\beta,\gamma}(t)=t^{\beta-1}\cdot\sum_{k=0}^{\infty}\frac{(-\gamma)_k}{k!}\frac{t^{k\alpha}}{\Gamma(k\alpha+\beta)}\varepsilon(t),\quad z\in\mathbb{C},\quad \alpha,\beta,\gamma,t\in\mathbb{R}. \tag{28}$$

Its LT is given by:

$$C_{\alpha,\beta,\gamma}(s)=\frac{s^{\alpha\gamma-\beta}}{(s^\alpha-p)^\gamma},\ \gamma\notin\mathbf{Z}, \tag{29}$$

for $\mathrm{Re}(s)=\sigma>\max\left\{\mathrm{Re}(p^{1/\alpha}),0\right\}$. If $\gamma\in\mathbf{Z}^+$, then we obtain a combination of MLF. If $\gamma\in\mathbf{Z}^-$, then we have a sum of powers of the type t^α.

For computing $c_{\alpha,\beta,\gamma}(t)$ we need to adapt procedures presented in Section 3 as follows:

(i) Computation based on the integral formulation

In this case the LT (29) does not have any pole, meaning that its inverse assumes the form [24]:

$$c_{\alpha,\beta,\gamma}(t)=\frac{1}{2\pi i}\cdot\int_{0^+}^{\infty}\frac{e^{i\beta\pi}(u^\alpha-pe^{-i\alpha\pi})^\gamma-e^{-i\beta\pi}(u^\alpha-pe^{i\alpha\pi})^\gamma}{[u^{2\alpha}-2p\cos(\alpha\pi)u^\alpha+p^2]^\gamma\cdot u^{\beta-\alpha\gamma}}e^{-ut}du\cdot\varepsilon(t). \tag{30}$$

(ii) Computation based on the inversion of the LT using the FFT

In this case, the approach is similar to that used in (18), yielding:

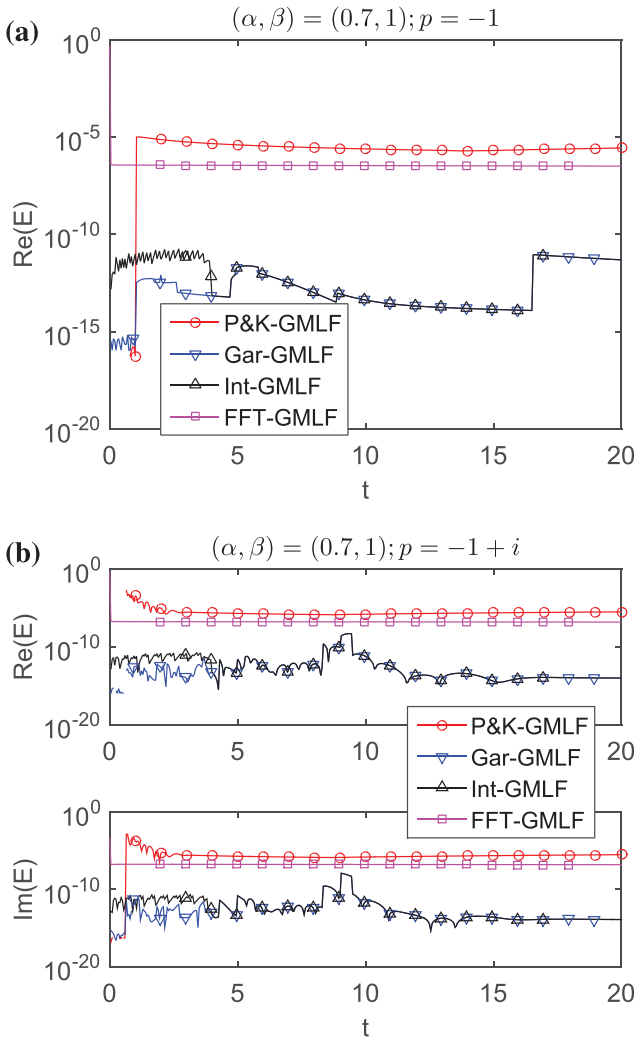

Figure 10: The error, E, in the numerical evaluation of the GMLF for the cases (a) C_8: $(\alpha,\beta)=(0.7,1)$, $t\in]0,20]$, $p=-1$; (b) C_9: $(\alpha,\beta)=(0.7,1)$, $t\in]0,20]$, $p=-1+i$.

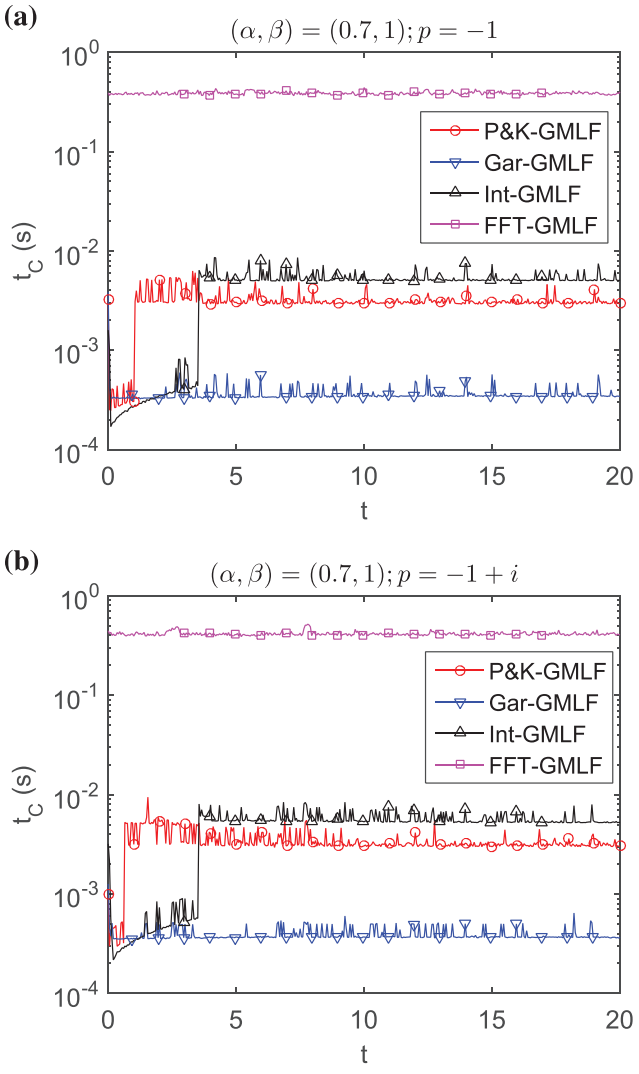

Figure 11: The computational time, t_C, to evaluate the GMLF for the cases (a) C_8: $(\alpha,\beta)=(0.7,1)$, $t\in]0,20]$, $p=-1$; (b) C_9: $(\alpha,\beta)=(0.7,1)$, $t\in]0,20]$, $p=-1+i$.

$$C_d(z)=\frac{\left[\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right]^{\alpha\cdot\gamma-\beta}}{\left\{\left[\frac{2}{T}\frac{1-z^{-1}}{1+z^{-1}}\right]^{\alpha}-p\right\}^{\gamma}}. \tag{31}$$

We compare the performance of the FFT-GMLF and Gar-MLF methods for the parameters $(\alpha,\beta,\gamma)=(0.7,1,0.8)$, and argument $t\in]0,20]$, $p=-1+i$. Figure 12 depicts the value of GMLF and the computational time. We verify again the good performance of the proposed method in terms of accuracy, while t_C is considerably larger.

5 Conclusions

In this paper, we proposed algorithms for efficiently computing the MLF and its particular version, the CMLF, based on polynomial calculations and integral formulas. For the GMLF another method, relying on the inversion of the LT using the binomial transformation and the FFT, was proposed. Simulation results comparing the new and two well-known methods, regarding the computational time and accuracy, showed the effectiveness of the proposed approach.

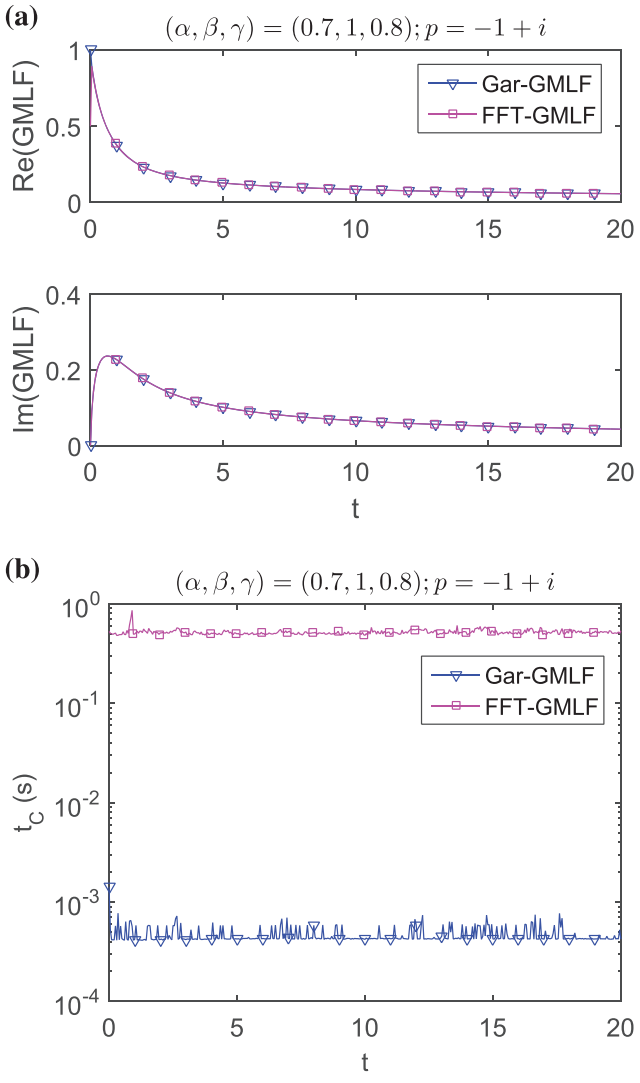

Figure 12: The GMLF for parameters $(\alpha, \beta, \gamma) = (0.7, 1, 0.8)$, and argument $t \in]0, 20]$, $p = -1 + i$: (a) GMLF value; (b) computational time, t_C.

Acknowledgment: This work was funded by Portuguese National Funds through the FCT - Foundation for Science and Technology under the project PEst-UID/EEA/00066/2013.

References

[1] R. Gorenflo, J. Loutchko and Y. Luchko, Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z)$ and its derivative, Fract. Calc. Appl. Anal. **5** (2002), 491–518.

[2] R. Gorenflo, A. A. Kilbas, F. Mainardi and S. V. Rogosin, Mittag-Leffler functions, related topics and applications, Springer, Berlin Heidelberg New York, 2014.

[3] F. Mainardi and R. Gorenflo, On Mittag-Leffler-type functions in fractional evolution processes, J. Comput. Appl. Math. **118** (2000), 283–299.

[4] G. M. Mittag-Leffler, Sur la nouvelle fonction $E_\alpha(x)$, CR Acad. Sci. Paris **137** (1903), 554–558.

[5] T. E. Huillet, On Mittag-Leffler distributions and related stochastic processes, J. Comput. Appl. Math. **296** (2016), 181–211.

[6] A. A. Kilbas, H. M. Srivastava and J. J. Trujillo, Theory and applications of fractional differential equations, vol. 204, North-Holland Mathematics Studies, Elsevier, Amsterdam, 2006.

[7] V. S. Kiryakova, Multiple (multiindex) Mittag-Leffler functions and relations to generalized fractional calculus, J. Comput. Appl. Math. **118** (2000), 241–259.

[8] N. Kopteva and M. Stynes, Analysis and numerical solution of a Riemann-Liouville fractional derivative two-point boundary value problem, Adv. Comput. Math. **43** (2017), 77–99.

[9] M. D. Ortigueira, C. M. Ionescu, J. T. Machado and J. J. Trujillo, Fractional signal processing and applications, Signal Process. **107** (2015), 197–197.

[10] M. D. Ortigueira, J. A. T. Machado, J. J. Trujillo and B. M. Vinagre, Fractional signals and systems, Signal Process. **91** (2011), 349–349.

[11] M. D. Ortigueira, Fractional calculus for scientists and engineers, Lecture Notes in Electrical Engineering, Springer, Berlin Heidelberg, 2011.

[12] M. D. Ortigueira and J. A. T. Machado, Fractional signal processing and applications, Signal Process. **83** (2003), 2285–2286.

[13] M. D. Ortigueira and J. A. T. Machado, Fractional calculus applications in signals and systems, Signal Process. **86** (2006), 2503–2504.

[14] I. Podlubny, Fractional differential equations: an introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications, vol. 198, Academic press, San Diego, 1998.

[15] X. J. Yang and D. Baleanu, Fractal heat conduction problem solved by local fractional variation iteration method, Therm. Sci. **17** (2013), 625–628.

[16] X. J. Yang, H. M. Srivastava, J. H. He and D. Baleanu, Cantor-type cylindrical-coordinate method for differential equations with local fractional derivatives, Phys. Lett. A **377** (2013), 1696–1700.

[17] R. Magin, M. D Ortigueira, I. Podlubny and J. Trujillo, On the fractional signals and systems, Signal Process. **91** (2011), 350–371.

[18] R. Garrappa and M. Popolizio, Evaluation of generalized Mittag-Leffler functions on the real line, Adv. Comput. Math. **39** (2013), 205–225.

[19] H. Seybold and R. Hilfer, Numerical algorithm for calculating the generalized Mittag-Leffler function, SIAM J. Numer. Anal. **47** (2008), 69–88.

[20] D. Valério and J. T. Machado, On the numerical computation of the Mittag-Leffler function, Commun. Nonlinear Sci. Numer. Simul. **19** (2014), 3419–3424.

[21] D. Valério, J. J. Trujillo, M. Rivero, J. T. Machado and D. Baleanu, Fractional calculus: a survey of useful formulas, The Eur. Phys. J. Spec. Top. **222** (2013), 1827–1846.

[22] F. Oberhettinger, A. Erdélyi, W. Magnus and F. G. Tricomi, Higher transcendental functions, vol. 3, McGraw-Hill, 1955.

[23] C. Lavault, Fractional calculus and generalized Mittag-Leffler type functions, arXiv:1703.01912v2 (2017).

[24] P. Henrici, Applied and computational complex analysis, vol. 1, Wiley-Interscience, New York, 1974.
[25] R. Hilfer and H. J. Seybold, Computation of the generalized Mittag-Leffler function and its inverse in the complex plane, Integr. Transform. Spec. Funct. **17** (2006), 637–652.
[26] M. J. Roberts, Signals and systems: Analysis using transform methods and Matlab, McGraw-Hill, New York, 2003.
[27] I. Podlubny and M. Kacenak, MLF - Mittag-Leffler function, September 2012.
[28] R. Garrapa, The Mittag-Leffler function, December 2015.