Algorithms for the Fractional Calculus: A Selection of Numerical Methods

K. Diethelm [a,1] N.J. Ford [b] A.D. Freed [c,1,*] Yu. Luchko [d]

[a] *Institut für Angewandte Mathematik, Technische Universität Braunschweig, Pockelsstraße 14, D-38106 Braunschweig, Germany*

[b] *Department of Mathematics, Chester College, Parkgate Road, Chester, CH1 4BJ, United Kingdom*

[c] *Polymers Branch, Materials Division, MS 49-3, NASA's John H. Glenn Research Center at Lewis Field, 21000 Brookpark Road, Brook Park, Ohio 44135, USA*
And: Department of Biomedical Engineering, ND 20, Lerner Research Institute, Cleveland Clinic Foundation, 9500 Euclid Avenue, Cleveland, Ohio 44195, USA

[d] *European University Viadrina, Business Informatics, POB 1768, D-15207 Frankfurt (Oder), Germany*

Abstract

Many recently developed models in areas like viscoelasticity, electrochemistry, diffusion processes, etc. are formulated in terms of derivatives (and integrals) of fractional (non-integer) order. In this paper we present a collection of numerical algorithms for the solution of the various problems arising in this context. We believe that this will give the engineer the necessary tools required to work with fractional models in an efficient way.

Key words: Riemann-Liouville integrals, Caputo derivatives, Caputo differential equations, Mittag-Leffler function.

* Corresponding author.
Email addresses: K.Diethelm@tu-bs.de (K. Diethelm), njford@chester.ac.uk (N.J. Ford), Alan.D.Freed@nasa.gov (A.D. Freed), Loutchko@euv-frankfurt-o.de (Yu. Luchko).

[1] Partially funded by US Army Medical Research and Material Command via Award No. DAMD17-01-1-0673 to the Cleveland Clinic.

1 Introduction

In recent years it has turned out that many phenomena in engineering, physics, chemistry, and other sciences can be described very successfully by models using mathematical tools from fractional calculus, i.e. the theory of derivatives and integrals of fractional (non-integer) order. Some of the most prominent examples are given in a book by Oldham and Spanier [41] (diffusion processes) and the classic papers of Bagley and Torvik [2], Caputo [4], and Caputo and Mainardi [5] (these three papers dealing with the modeling of viscoelastic materials) as well as in the publications of Marks and Hall [34] (signal processing) and Olmstead and Handelsman [42] (also dealing with diffusion problems). More recent results are described, e.g., in the works of Chern [6] (finite element implementation of viscoelastic model), Diethelm and Freed [11] (viscoplastic material model), Gaul, Klein, and Kempfle [19] (description of mechanical systems subject to damping), Glöckle and Nonnenmacher [20] (relaxation and reaction kinetics of polymers), Gorenflo and Rutman [24] (so-called ultraslow processes), Metzler et al. [36] (relaxation in filled polymer networks), Unser and Blu [52,53] (splines and wavelets), Podlubny [44] (control theory), and Podlubny et al. [47] (heat propagation). Surveys with collections of applications can also be found in Gorenflo and Mainardi [23], Mainardi [32], Matignon and Montseny [35], Nonnenmacher and Metzler [39], and Podlubny [45]. Therefore, it seems to be fair to say that there is a significant demand for readily usable tools that numerically handle these mathematical problems. However, to date, such algorithms have been developed only to a rather limited extent. Thus, it is the aim of this paper to identify the most important problems for which specific algorithms are required, to present those schemes that have been developed so far in a way that is directly accessible to the applied scientist, and to fill the remaining gaps with suitable new methods.

The structure of this paper is as follows. In the next three sections we briefly recall the mathematical foundations of the fractional calculus. We then turn to a more detailed description and assessment of our numerical methods for problems described by fractional-order derivatives, integrals, and differential equations. After this we provide an algorithm for calculating the Mittag-Leffler function, which appears in solutions to fractional-order differential equations. Finally, after a discussion of our algorithms, we provide appendices where additional useful results are laid out that are connected to the numerical issues of fractional calculus indirectly (such as, e.g., a table of certain fractional derivatives, and some helpful auxiliary subroutines that can be utilized when working with the fundamental algorithms described in the main part of the paper).

2 Riemann-Liouville Fractional Integral

In the classical calculus of Newton and Leibniz, Cauchy reduced the calculation of an n-fold integration of the function $\mathbf{y}(x)$ into a single convolution integral possessing an Abel (power law) kernel,

$$J^n\mathbf{y}(x) := \int_0^x \int_0^{x_{n-1}} \cdots \int_0^{x_1} \mathbf{y}(x_0)\,\mathrm{d}x_0 \ldots \mathrm{d}x_{n-2}\,\mathrm{d}x_{n-1}$$
$$= \frac{1}{(n-1)!}\int_0^x \frac{1}{(x-x')^{1-n}}\,\mathbf{y}(x')\,\mathrm{d}x', \quad n \in \mathbb{N}, \quad x \in \mathbb{R}_+, \quad (1)$$

where J^n is the n-fold integral operator with $J^0\mathbf{y}(x) = \mathbf{y}(x)$, $\mathbb{N}$ is the set of positive integers, and $\mathbb{R}_+$ is the set of positive reals. Liouville and Riemann [2] analytically continued Cauchy's result, replacing the discrete factorial $(n-1)!$ with Euler's continuous gamma function $\Gamma(n)$, noting that $(n-1)! = \Gamma(n)$, thereby producing [30, ¶5, Eqn. A]

$$J^\alpha\mathbf{y}(x) := \frac{1}{\Gamma(\alpha)}\int_0^x \frac{1}{(x-x')^{1-\alpha}}\,\mathbf{y}(x')\,\mathrm{d}x', \quad \alpha, x \in \mathbb{R}_+, \quad (2)$$

where J^α is the Riemann-Liouville integral operator of order α,·which commutes (i.e. $J^\alpha J^\beta\mathbf{y}(x) = J^\beta J^\alpha\mathbf{y}(x) = J^{\alpha+\beta}\mathbf{y}(x)\ \forall\ \alpha, \beta \in \mathbb{R}_+$). Equation (2) is the cornerstone of the fractional calculus, although it may vary in its assignment of limits of integration. In this document we take the lower limit to be zero and the upper limit to be some positive finite real. Actually, α can be complex [51], but in the applications that are of interest to us α is real.

A brief history of the development of fractional calculus can be found in Ross [50] and Miller and Ross [37, Chp. 1]. A survey of many emerging applications of the fractional calculus in areas of science and engineering can be found in the recent text by Podlubny [45, Chp. 10].

3 Caputo-Type Fractional Derivative

3.1 Fundamental definition

From this single definition for fractional integration, one can construct several definitions for fractional differentiation (cf., e.g., with Refs. [45,51]). The special operator $D_\star^\alpha$ that we choose to use, which requires the dependent variable

[2] Riemann's pioneering work in the field of fractional calculus was done during his student years, but published posthumous—forty-four years after Liouville first published in the field [50].

$\mathbf{y}$ to be continuous and $\lceil\alpha\rceil$-times differentiable in the independent variable x, is defined by

$$D_\star^\alpha \mathbf{y}(x) := J^{\lceil\alpha\rceil-\alpha} D^{\lceil\alpha\rceil}\mathbf{y}(x), \tag{3}$$

such that

$$\lim_{\alpha\to n_-} D_\star^\alpha \mathbf{y}(x) = D^n\mathbf{y}(x) \quad \text{for any} \quad n\in\mathbb{N}, \tag{4}$$

with $D_\star^0\mathbf{y}(x) = \mathbf{y}(x)$, where $\lceil\alpha\rceil$ is the ceiling function giving the smallest integer greater than or equal to α, and where $\alpha\to n_-$ means α approaches n from below. The operator D^n, $n\in\mathbb{N}$, is the classical differential operator. It is accepted practice to call $D_\star^\alpha$ the Caputo differential operator of order α, after Caputo [4] who was the among the first to use this operator in applications and to study some of its properties.[3] Appendix A presents a table of Caputo derivatives for some of the more common mathematical functions.

The Caputo differential operator is a linear operator, i.e.

$$D_\star^\alpha\Big(a\mathbf{y}+b\mathbf{z}\Big)(x) = aD_\star^\alpha\mathbf{y}(x) + bD_\star^\alpha\mathbf{z}(x) \tag{5}$$

for arbitrary constants a and b, that commutes, viz.

$$D_\star^\alpha D_\star^\beta\mathbf{y}(x) = D_\star^\beta D_\star^\alpha\mathbf{y}(x) = D_\star^{\alpha+\beta}\mathbf{y}(x) \quad \forall\ \alpha,\beta\in\mathbb{R}_+, \tag{6}$$

if $\mathbf{y}(x)$ is sufficiently smooth, and it possesses the desirable property that

$$D_\star^\alpha c = 0 \quad \text{for any constant } c. \tag{7}$$

The more common Riemann-Liouville fractional derivative D^α, although linear, need not commute [45, pg. 74]; furthermore, $D^\alpha c = D^{\lceil\alpha\rceil}J^{\lceil\alpha\rceil-\alpha}c = cx^{-\alpha}/\Gamma(1-\alpha)$, which depends on x. Ross [50] attributes this startling fact as the main reason why the fractional calculus has historically had a difficult time being embraced by the mathematics and physics communities.

---

[3] Actually, Liouville introduced the operator in his historic first paper on the topic [30, ¶6, Eqn. B]. Still, nothing in Liouville's works suggests that he ever saw any difference between $D_\star^\alpha := J^{\lceil\alpha\rceil-\alpha}D^{\lceil\alpha\rceil}$ and $D^\alpha := D^{\lceil\alpha\rceil}J^{\lceil\alpha\rceil-\alpha}$, with D^α being his accepted definition [30, first formula on pg. 10]—the Riemann-Liouville differential operator of order α. Liouville freely interchanged the order of integration and differentiation, because the class of problems that he was interested in happened to be a class where such an interchange is legal, and he made only a few terse remarks about the general requirements on the class of functions for which his fractional calculus works [31]. The accepted naming of the operator $D_\star^\alpha$ after Caputo therefore seems warrented.
Rabotnov [49, pg. 129] introduced this same differential operator into the Russian viscoelastic literature a year before Caputo's paper was published. Regardless of this fact, operator $D_\star^\alpha$ is commonly named after Caputo in the present-day literature.

The Riemann-Liouville integral operator J^α and the Caputo differential operator $D^\alpha_\star$ are inverse operators in the sense that

$$D^\alpha_\star J^\alpha \mathbf{y}(x) = \mathbf{y}(x) \quad \text{and} \quad J^\alpha D^\alpha_\star \mathbf{y}(x) = \mathbf{y}(x) - \sum_{k=0}^{\lfloor \alpha \rfloor} \frac{x^k}{k!}\, \mathbf{y}^{(k)}_{0^+}, \quad \alpha \in \mathbb{R}_+, \quad (8)$$

with $\mathbf{y}^{(k)}_{0^+} := D^k \mathbf{y}(0^+)$, where $\lfloor \alpha \rfloor$ is the floor function giving the largest integer less than or equal to α. The classic n-fold integral and differential operators of integer order satisfy like formulæ, viz.: $D^n J^n \mathbf{y}(x) = \mathbf{y}(x)$, and $J^n D^n \mathbf{y}(x) = \mathbf{y}(x) - \sum_{k=0}^{n-1} \frac{x^k}{k!}\, \mathbf{y}^{(k)}_{0^+}$, $n \in \mathbb{N}$. In this sense, the Caputo derivative and Riemann-Liouville integral are analytic continuations of the well-known n-fold derivative and integral from the classical calculus.

Remark 1 *Fractional derivatives do not satisfy the Leibniz product rule as it is known in integer-order calculus. For example, whenever the Caputo derivative is restricted so that* $0 < \alpha < 1$, *its Leibniz product rule is given by*

$$D^\alpha_\star\big(\mathbf{y} \times \mathbf{z}\big)(x) = \frac{\mathbf{y}(0^+)}{\Gamma(1-\alpha)} \times \frac{\mathbf{z}(x) - \mathbf{z}(0^+)}{x^\alpha} + \big(D^\alpha_\star \mathbf{y}\big)(x) \times \mathbf{z}(x) + \sum_{k=1}^{\infty} \binom{\alpha}{k} \big(J^{k-\alpha}\mathbf{y}\big)(x) \times \big(D^k \mathbf{z}\big)(x), \quad (9)$$

wherein, unlike the Leibniz product rule for integer-order derivatives, the binomial coefficients $\binom{\alpha}{k} = \alpha(\alpha-1)(\alpha-2)\cdots(\alpha-k+1)/k!$ *(with* $\binom{\alpha}{0} = 1$, $\alpha \in \mathbb{R}_+$, *and* $k \in \mathbb{N}$*) do not become zero whenever* $k > \alpha$ *because* $\alpha \notin \mathbb{N}$ *(i.e., the binomial sum is now of infinite extent). A similar infinite sum exists for the Leibniz product rule of the Riemann-Liouville fractional derivative (cf. Podlubny [45, pp. 91–97]).*

3.2 Alternative integral expressions

The Caputo derivative defined in Eq. (3) can be expressed in a more explicit notation as the integral

$$D^\alpha_\star \mathbf{y}(x) = \frac{1}{\Gamma(\lceil \alpha \rceil - \alpha)} \int_0^x \frac{1}{(x-x')^{1+\alpha-\lceil \alpha \rceil}} \big(D^{\lceil \alpha \rceil}\mathbf{y}\big)(x')\, \mathrm{d}x', \quad \alpha, x \in \mathbb{R}_+, \quad (10)$$

where the weak singularity caused by the Abel kernel of the integral operator is readily observed. This singularity can be removed through an integration

by parts

$$D^{\alpha}_{\star}\mathbf{y}(x) = \frac{1}{\Gamma(1+\lceil\alpha\rceil-\alpha)}\Bigg(x^{\lceil\alpha\rceil-\alpha}\,\mathbf{y}^{(\lceil\alpha\rceil)}_{0+} + \int_0^x \big(x-x'\big)^{\lceil\alpha\rceil-\alpha}\big(D^{1+\lceil\alpha\rceil}\mathbf{y}\big)(x')\,\mathrm{d}x'\Bigg), \quad (11)$$

provided that the dependent variable $\mathbf{y}$ is continuous and $(1+\lceil\alpha\rceil)$-times differentiable in the independent variable x over the interval of differentiation (integration) $[0,x]$. In Eq. (11) the power-law kernel is bounded over the entire interval of integration; whereas, in Eq. (10) the kernel is singular at the upper limit of integration.

The two representations in formulæ (10) and (11) are quite useful for pen-and-paper calculations, but in order to obtain a numerical scheme for the approximation of such fractional derivatives, we found it even more helpful to look at yet another representation that seems to have been introduced into this context by Elliott [14]; namely,

$$D^{\alpha}_{\star}\mathbf{y}(x) = \frac{1}{\Gamma(-\alpha)}\int_0^x \frac{1}{(x-x')^{\alpha+1}}\,\mathbf{y}(x')\,\mathrm{d}x', \quad \alpha, x \in \mathbb{R}_+. \quad (12)$$

This representation can also be obtained from Eq. (10) using the method of integration by parts, but with the roles of the two factors interchanged. The advantage here is that the function $\mathbf{y}$ itself appears in the integrand instead of its derivative. The disadvantage is that the singularity of the kernel is now strong rather than weak, and thus we have to interpret this integral as a Hadamard-type finite-part integral. This is cumbersome in pen-and-paper calculations but, as we shall see below, it is not a problem to devise an algorithm that makes the computer do this job. We provide a brief description of such an algorithm in the following pages. For more details, the interested reader is referred to Refs. [7,14] and the references cited therein.

4 Caputo-Type FDE's

Fractional material models, a topic of interest to one of the authors (ADF), are an important application where systems of fractional-order differential equations (FDE's) arise that need to be solved in accordance with appropriate initial and boundary conditions. A number of other examples are well known; we refer the reader to the book by Podlubny [45, Chap. 10] and the survey of Mainardi [32] for more information.

A FDE of the Caputo type has the form

$$D_{\star}^{\alpha}\mathbf{y}(x) = \mathbf{f}\big(x, \mathbf{y}(x)\big), \quad \alpha, x \in \mathbb{R}_{+}, \tag{13}$$

satisfying a set of (possibly inhomogeneous) initial conditions

$$D^{k}\mathbf{y}(0^{+}) = \mathbf{y}_{0^{+}}^{(k)}, \quad k = 0, 1, \ldots, \lfloor \alpha \rfloor, \tag{14}$$

whose solution is sought over an interval $[0, X]$, say, where $X \in \mathbb{R}_{+}$. It turns out that under some very weak conditions placed on the function $\mathbf{f}$ of the right-hand side, a unique solution to Eqs. (13 & 14) does exist [8].

A typical feature of differential equations (both classical and fractional) is the need to specify additional conditions in order to produce a unique solution. For the case of Caputo FDE's, these additional conditions are just the static initial conditions listed in (14), which are akin to those of classical ODE's, and are therefore familiar to us. In contrast, for Riemann-Liouville FDE's, these additional conditions constitute certain fractional derivatives (and/or integrals) of the unknown solution at the initial point $x = 0$ [27], which are functions of x. These initial conditions are not physical; furthermore, it is not clear how such quantities are to be measured from experiment, say, so that they can be appropriately assigned in an analysis. [4] If for no other reason, the need to solve FDE's is justification enough for choosing Caputo's definition (i.e., $D_{\star}^{\alpha} := J^{\lceil\alpha\rceil-\alpha}D^{\lceil\alpha\rceil}$) for fractional differentiation over the more commonly used (at least in mathematical analysis) definition of Liouville and Riemann (viz., $D^{\alpha} := D^{\lceil\alpha\rceil}J^{\lceil\alpha\rceil-\alpha}$).

5 Numerical Approximation of Caputo-type Derivatives

5.1 Fundamental ideas

Unlike ordinary derivatives, which are point functionals, fractional derivatives are hereditary functionals possessing a total memory of past states. A

[4] We explicitly note, however, the very recent paper by Podlubny [46] who attempts to give highly interesting geometrical and physical interpretations for fractional derivatives of both the Riemann-Liouville and Caputo types. These interpretations are deeply related to the questions: What precisely is time? Is it absolute or not? And can it be measured correctly and accurately, and if so, how? Thus, we are still a long way from a full understanding of the geometric and physical nature of a fractional derivative, let alone from an idea of how we can measure it in an experiment, but our mental image of what fractional derivatives and integrals 'look like' continues to improve.

numerical algorithm for computing Caputo derivatives has been derived by Diethelm [7][5] and is listed in Alg. 1. Validity of its Richardson extrapolation scheme for $1 < \alpha < 2$, or one similar to it, has to date not been proven, or disproven. Here $\mathbf{y}_n$ denotes $\mathbf{y}(x_n)$, while $\mathbf{y}_N$ represents $\mathbf{y}(X)$ where $[0, X]$ is the interval of integration (fractional differentiation) with $0 \le x_n \le X$. This algorithm was arrived at by approximating the integral in Eq. (12) with a product trapezoidal method, thereby restricting $0 < \alpha < 2$. Similar algorithms applicable to larger ranges of α can be constructed by using the general procedure derived in Ref. [7], if they become needed.

Algorithm 1
Computation of a Caputo fractional derivative $(0 < \alpha < 2, \alpha \neq 1)$:
For interval $[0, X]$ *with grid* $\{x_n = nh\colon n = 0, 1, 2, \ldots, N\}$ *where* $h = X/N$, *compute*

$$D_\star^\alpha \mathbf{y}_N(h) = [h^{-\alpha}/\Gamma(2-\alpha)] \sum_{n=0}^{N} a_{n,N} \Big(\mathbf{y}_{N-n} - \sum_{k=0}^{\lfloor \alpha \rfloor} [(N-n)^k h^k / k!] \mathbf{y}_{0+}^{(k)} \Big) ,$$

$$D_\star^\alpha \mathbf{y}(X) = D_\star^\alpha \mathbf{y}_N(h) + O(h^{2-\alpha}),$$

using the quadrature weights (derived from a product trapezoidal rule)

$$a_{n,N} = \begin{cases} 1, & \text{if } n = 0, \\ (n+1)^{1-\alpha} - 2n^{1-\alpha} + (n-1)^{1-\alpha}, & \text{if } 0 < n < N, \\ (1-\alpha)N^{-\alpha} - N^{1-\alpha} + (N-1)^{1-\alpha}, & \text{if } n = N. \end{cases}$$

Refine, if desired, using Richardson extrapolation

$$D_\star^\alpha \mathbf{y}_v^u(X) = \Big(D_\star^\alpha \mathbf{y}_{v-1}^{u-1}(X) - 2^{r_{u-1}} D_\star^\alpha \mathbf{y}_v^{u-1}(X) \Big) / (1 - 2^{r_{u-1}}),$$

$$D_\star^\alpha \mathbf{y}(X) = D_\star^\alpha \mathbf{y}_u^u(X) + O(h^{r_u}),$$

such that if $0 < \alpha < 1$ *then* r_{u-1} *is assigned as*

$$r_0 = 2 - \alpha,$$

$$r_1 = 2,\ r_2 = 3 - \alpha,\ r_3 = 4 - \alpha,$$

$$r_4 = 4,\ r_5 = 5 - \alpha,\ r_6 = 6 - \alpha,$$

$$r_7 = 6, \cdots .$$

The Grünwald-Letnikov algorithm is often used to numerically approximate the Riemann-Liouville fractional derivative (cf., e.g., with Oldham and Spanier [41, §8.2] and Podlubny [45, Chp. 7]), and it was the first algorithm to appear for approximating fractional derivatives (and integrals).

[5] Apparently this algorithm first appeared in the PhD thesis of Chern [6], unbeknownst to us (KD) at the time of writing Ref. [7]. Chern used this algorithm to differentiate a Kelvin-Voigt, fractional-order, viscoelastic, material model in a finite element code. He did not address stability or uniqueness of solution issues; he did not compute error estimates; and he did not utilize an extrapolation scheme to enhance solution accuracy.

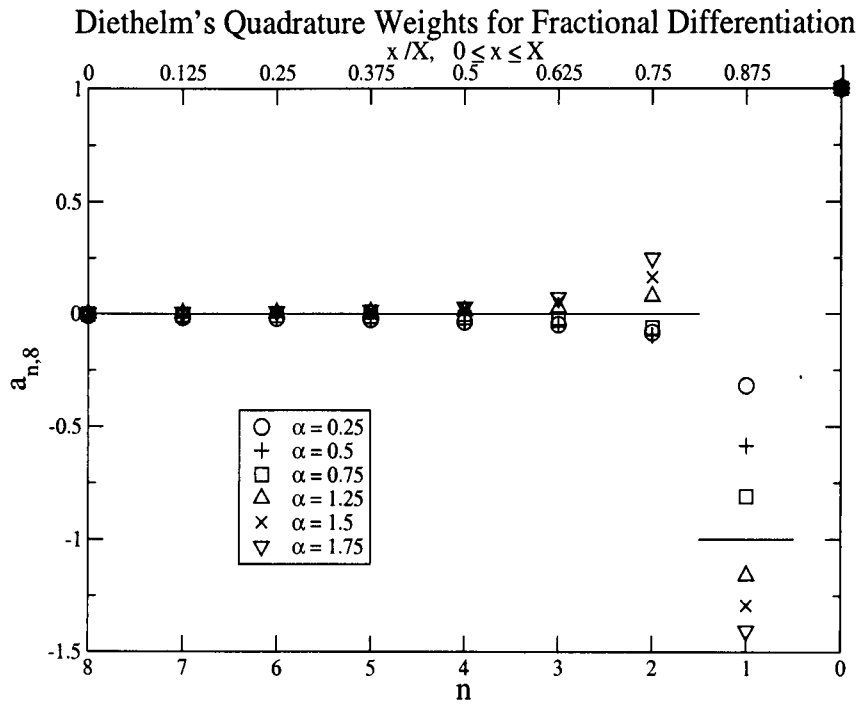

Fig. 1. Weights of quadrature $a_{n,N}$ for approximating Caputo's fractional derivative (12) over interval $[0, X]$ using Diethelm's [7] Alg. 1, plotted here for various values of α with $N = 8$.

5.2 Quadratures

The extent of rememberance of past states exhibited by the hereditary nature of a fractional derivative is manifest, for example, in its weights of quadrature, as illustrated in Fig. 1. This operator exhibits a fading memory: $0.001 < |a_{8,8}| < 0.01$ for the six cases plotted in this figure. If $D\mathbf{y}(X)$ were to be approximated by a backward difference with $h = X/8$, then the effective weights of quadrature would be $a_{0,8} = 1$ and $a_{1,8} = -1$ with all remaining weights being zero, as represented by the line segments in this figure. Similarly, if $D^2\mathbf{y}(X)$ were to be approximated by a like backward-difference scheme, then $a_{0,8} = 1$, $a_{1,8} = -2$ and $a_{2,8} = 1$ with all remaining weights being zero. It is evident from the data presented in Fig. 1 that the weights of quadrature $a_{n,8}$ for approximating $D^{\alpha}_{\star}\mathbf{y}(X)$ are compatible with those for the first- and second-order backward differences, and that fractional quadratures have additional contributions that monotonically diminish with increasing nodal number from node $n = 2$ fading all the way back to the origin at node $n = N$. This suggests that a truncation scheme may be able to be used to enhance algorithmic efficiency for some classes of functions, but not all.

5.3 *Richardson extrapolation*

Richardson extrapolation is a technique that can often be used to increase the accuracy of results [12]. As we utilize it, this technique follows a triangular scheme—a Romberg tableau—that has the form

$$\begin{array}{llll} D_\star^\alpha \mathbf{y}_0^0 & & & \\ D_\star^\alpha \mathbf{y}_1^0 & D_\star^\alpha \mathbf{y}_1^1 & & \\ D_\star^\alpha \mathbf{y}_2^0 & D_\star^\alpha \mathbf{y}_2^1 & D_\star^\alpha \mathbf{y}_2^2 & \\ D_\star^\alpha \mathbf{y}_3^0 & D_\star^\alpha \mathbf{y}_3^1 & D_\star^\alpha \mathbf{y}_3^2 & D_\star^\alpha \mathbf{y}_3^3 \\ \vdots & \vdots & \vdots & \vdots \quad \ddots \end{array} \tag{15}$$

Constructing the first column of the tableau constitutes the bulk of the computational effort. Here $D_\star^\alpha \mathbf{y}_0^0 := D_\star^\alpha \mathbf{y}_N(h)$ denotes the value of $D_\star^\alpha \mathbf{y}$ evaluated numerically at X over $[0, X]$ using an initial stepsize of h $(= X/N)$, $D_\star^\alpha \mathbf{y}_1^0 := D_\star^\alpha \mathbf{y}_N(h/2)$ is computed using the refined stepsize of $\frac{1}{2}h$ $(= X/2N)$, $D_\star^\alpha \mathbf{y}_2^0 := D_\star^\alpha \mathbf{y}_N(h/4)$ is computed using the even more refined stepsize of $\frac{1}{4}h$ $(= X/4N)$, while $D_\star^\alpha \mathbf{y}_3^0 := D_\star^\alpha \mathbf{y}_N(h/8)$ is computed using the further refined stepsize of $\frac{1}{8}h$ $(= X/8N)$, etc. The remaining columns are quickly evaluated using the recursive formula listed in Alg. 1. The advantage of constructing this tableau is that the elements $D_\star^\alpha \mathbf{y}_v^u(X)$ in the u^{th} column converge for fixed u and increasing v towards the true value of the Caputo derivative as $O(h^{r_u})$. Hence, the further one moves to the right in the tableau the faster the column converges, and this level of convergence requires less computational effort to achieve than a direct computation of $D_\star^\alpha \mathbf{y}_N(X; \bar{h})$ when computed to a similar accuracy of $O(\bar{h}^{r_0}) \sim O(h^{r_u})$.

5.4 *Step-size choice*

The error analysis mentioned above is only a truncation error analysis. It assumes that the calculations are done in exact arithmetic, and it does not take into account effects like roundoff. When one needs to look at these effects too, it is possible to ask for a step size $h = X/N$ whose combined effect arising from both error sources is minimized. As we have seen above, it is likely that the truncation error decreases with the step size h, whereas roundoff tends to have the opposite behavior, so we should be looking for a sort of compromise. The considerations in this context are very similar to those for integer-order derivatives [48, §5.7]. Roughly speaking, it turns out that the roundoff error behaves as $h^{-\alpha}\epsilon_y \mathbf{y}(\eta)$, where ϵ_y is the relative accuracy with which one can compute $\mathbf{y}$, and where η designates some number within the

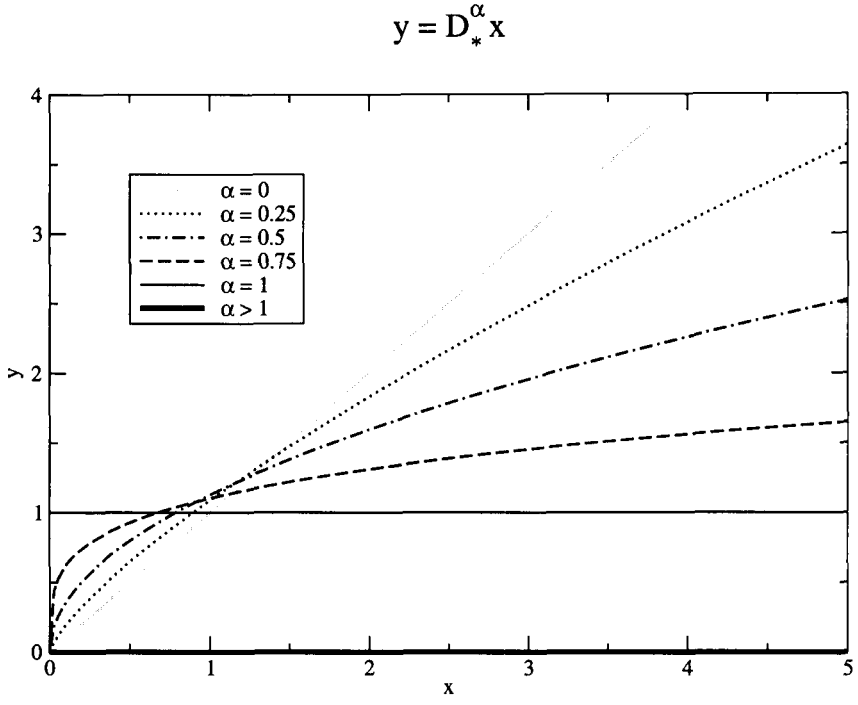

Fig. 2. The Caputo derivative $D_\star^\alpha x$ for various values of α.

interval $[0, X]$. Moreover, the truncation error is close to $c_\alpha h^{2-\alpha}\mathbf{y}''(\xi)$, where c_α is some constant independent of $\mathbf{y}$, and ξ is some other number also contained in the interval $[0, X]$. Consequently, an optimum step size would be of order $h \sim [\epsilon_y \mathbf{y}(\eta)/\mathbf{y}''(\xi)]^{1/2}$ when minimizing with respect to both trucation and roundoff errors.

Unless specific information indicating the contrary is available, one may assume that $\mathbf{y}$ and $\mathbf{y}''$ are not too irregularly behaved. Under these conditions $\mathbf{y}(\eta) \approx \mathbf{y}(X)$ and $\mathbf{y}''(\eta) \approx \mathbf{y}''(X)$, and one can then follow the suggestion of Press et al. [48, p. 187] by setting $\mathbf{y}(X)/\mathbf{y}''(X) \approx X$ (except near $X = 0$ where some other estimate for this quantity should be used). This scheme provides some advice on the choice of step size if roundoff effects are considered problematic in some specific application at hand.

5.5 Examples

The simplest fractional-order derivative that one can consider is $D_\star^\alpha \mathbf{y}(x)$ where $\mathbf{y}(x) = x$. A plot of this derivative is presented in Fig. 2. There are three special cases shown in this figure: $D_\star^0 x = x$, $D_\star^1 x = 1$, and $D_\star^\alpha x = 0 \; \forall \; \alpha > 1$, which are in accordance with the classic definition for differentiation. The intermediate values of $\alpha = 1/4$, $1/2$, and $3/4$ suggest a smooth transition in the response between $\alpha = 0$ and 1, as one would expect.

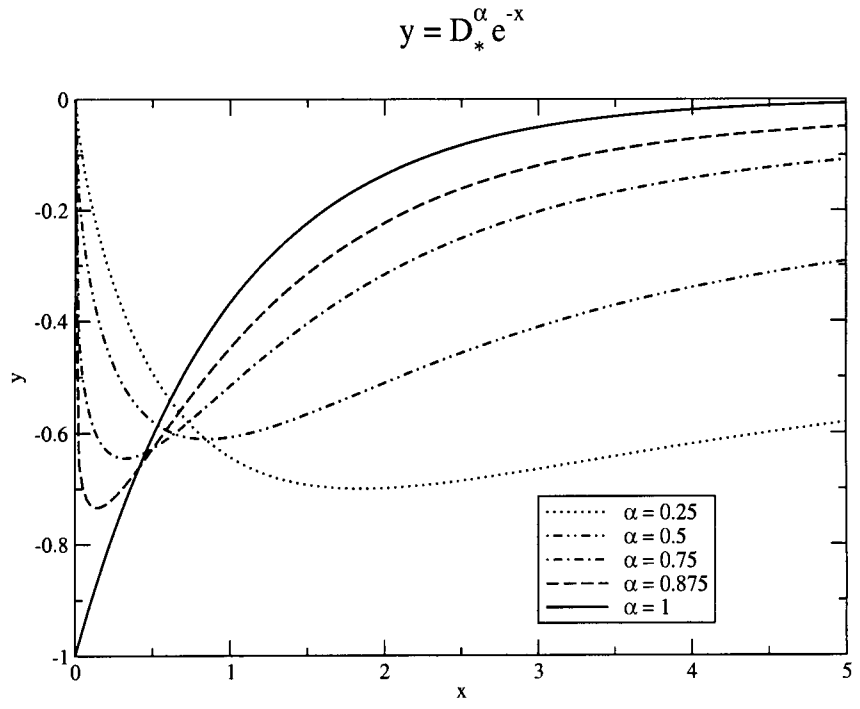

Fig. 3. The Caputo derivative $D_\star^\alpha \, \mathrm{e}^{-x}$ for various values of α.

The second example, presented in Fig. 3, is for $D_\star^\alpha \mathbf{y}(x)$ where $\mathbf{y}(x) = \mathrm{e}^{-x}$. There is a discontinuous jump at $x = 0$ between the first-order derivative $D\,\mathrm{e}^{-x} = -\,\mathrm{e}^{-x}$ and its Caputo counterparts $D_\star^\alpha \, \mathrm{e}^{-x}$ given that $0 < \alpha < 1$. These Caputo derivatives start at zero, drop rapidly piercing the $-\,\mathrm{e}^{-x}$ curve, and then gently swing up eventually asymptoting to zero from below.

6 Numerical Approximation of Riemann-Liouville Integrals

6.1 Fundamental ideas

Over the course of our research we have not only had to approximate fractional derivatives, but also fractional integrals. As indicated above, the natural concept for the fractional integral to be used in connection with Caputo derivatives is the Riemann-Liouville integral described in Eq. (2). We therefore present a numerical scheme for the solution of this problem, too. The underlying idea of the algorithm, stated in a formal way in Alg. 2 below, is completely identical to the idea presented above for the Caputo derivative; that is, we employ a product integration technique based on the trapezoidal quadrature rule. Said differently, we replace the given function $\mathbf{y}(x')$ on the right-hand side of Eq. (2) by a piecewise linear interpolant, and then we calculate the resulting integral exactly. As a matter of fact, this algorithm will also be part of the scheme introduced in Alg. 3 in the pages that follow for the

numerical solution of a certain type of fractional-order differential equation.

Algorithm 2
Computation of a Riemann-Liouville fractional integral ($\alpha > 0$):
For interval $[0, X]$ *with grid* $\{x_n = nh\colon n = 0, 1, 2, \ldots, N\}$ *where* $h = X/N$, *compute*

$$J^\alpha \mathbf{y}_N(h) = [h^\alpha/\Gamma(2+\alpha)] \sum_{n=0}^{N} c_{n,N}\, \mathbf{y}_n \ ,$$
$$J^\alpha \mathbf{y}(X) = J^\alpha \mathbf{y}_N(h) + O(h^2) \ ,$$

using the quadrature weights (derived from a product trapezoidal rule)

$$c_{n,N} = \begin{cases} (1+\alpha)N^\alpha - N^{1+\alpha} + (N-1)^{1+\alpha}, & \text{if } n = 0, \\ (N-n+1)^{1+\alpha} - 2(N-n)^{1+\alpha} + (N-n-1)^{1+\alpha}, & \text{if } 0 < n < N, \\ 1, & \text{if } n = N. \end{cases}$$

Refine, if desired, using Richardson extrapolation

$$J^\alpha \mathbf{y}_v^u(X) = \left(J^\alpha \mathbf{y}_{v-1}^{u-1}(X) - 2^{r_{u-1}}\, J^\alpha \mathbf{y}_v^{u-1}(X)\right)/(1 - 2^{r_{u-1}}) \ ,$$
$$J^\alpha \mathbf{y}(X) = J^\alpha \mathbf{y}_u^u(X) + O(h^{r_u}),$$

such that if $0 < \alpha < 1$ *then* r_{u-1} *is assigned as*

$$r_0 = 2,\ r_1 = 2 + \alpha,$$
$$r_2 = 3,\ r_3 = 3 + \alpha,$$
$$r_4 = 4,\ r_5 = 4 + \alpha,$$
$$r_6 = 5, \cdots .$$

Note 1 *Whenever* $\alpha > 1$ *in Alg. 2, the same values appear in the sequence* $r_0, r_1, r_2, \ldots$, *but they must be re-ordered to keep the sequence monotonic. For example, if* $1 < \alpha < 2$ *then we have* $r_0 = 2, r_1 = 3, r_2 = 2 + \alpha, r_3 = 4, r_4 = 3 + \alpha, \ldots$.

It is easily seen that the error of this algorithm is of the order $O(h^2)$ where, as in Alg. 1, h denotes the step size. Once again, we can improve the accuracy by adding a Richardson extrapolation procedure to the plain algorithm. The required exponents are known (cf. Ref. [26, §4]) and the resulting scheme is detailed in Alg. 2. Both the fundamental algorithm itself, and the Richardson extrapolation procedure, may be used for any positive value of α; there is no need to impose an upper bound on the legal range for α. This is due to the fact that the Abel (power law) kernel in the definition (2) of the Riemann-Liouville integral is regular, or at worst, weakly singular, and hence, integrable (at least in the improper sense) for any $\alpha > 0$. In contrast, the corresponding kernel in representation (12) of the Caputo derivative is not integrable. This kernel requires special regularization methods that are compatible with our approximation method, and as such, our scheme for approximating Caputo derivatives is only valid for $0 < \alpha < 2$; whereas, our corresponding scheme for approximating Riemann-Liouville integrals is valid for all $\alpha > 0$.

Table 1
Averaging procedure used to compute effective weights of quadrature for approximating Riemann-Liouville integration as they relate to Alg. 2.

Subinterval	Averaged Quadrature Weight $\langle c_{n,N} \rangle$
$[0, X/N]$	$\langle c_{0,N} \rangle = c_{0,N} + \frac{1}{2} c_{1,N}$
$[nX/N, (n+1)X/N]$	$\langle c_{n,N} \rangle = \frac{1}{2}(c_{n,N} + c_{n+1,N}), \quad (n = 1, 2, \ldots, N-2)$
$[(N-1)X/N, X]$	$\langle c_{N-1,N} \rangle = \frac{1}{2} c_{N-1,N} + c_{N,N}$

Notice the formal correspondence between Alg. 2 (for fractional *integration* of order α) and Alg. 1 (for fractional *differentiation* of order α). Except for the initial conditions that have to be taken into account additionally, the latter is simply obtained from the former by replacing the parameter α by $-\alpha$.[6] This relates to the intuitive (but not mathematically strictly correct) interpretation of fractional differentiation and integration being inverse operations. Also notice that the index ordering is inverted between these two algorithms, which is in keeping with accepted indexing conventions. Algorithm 1 indexes from $x_0 = X$ to $x_N = 0$, while Alg. 2 indexes from $x_0 = 0$ to $x_N = X$.

6.2 Quadratures

A visualization of quadrature weight versus nodal index for several values of α pertaining to Alg. 2 is presented in Fig. 4. What is striking about this figure is the obvious difference between domains $0 < \alpha < 1$ and $1 < \alpha < 2$. Whenever $\alpha = 1$, the algorithm reduces to classic trapezoidal integration. Whenever $0 < \alpha < 1$, the earlier states will contribute less to the overall solution than will the more recent states, but they do not entirely fade out. Fractional integration exhibits a long-term memory loss when $0 < \alpha < 1$ but, unlike fractional differentiation, fractional integration does not experience a rapid loss (or fading away) of past memories. Also, the smaller the value of α (i.e., the closer it is to zero) the greater the degree of long-term memory loss will be. In contrast, whenever $1 < \alpha < 2$, the earlier states will contribute more to the overall solution than will the more recent states. Fractional integration therefore exhibits a short-term memory loss when $1 < \alpha < 2$. This is like an elderly person who remembers in vivid detail what happened years ago, but who cannot recall what took place yesterday. Furthermore, the greater the value of α (i.e., the closer it is to two) the more pronounced the short-term memory loss will be.

[6] Similarly, the Grünwald-Letnikov algorithm for approximating Riemann-Liouville fractional derivatives of order α also applies for approximating Riemann-Liouville fractional integrals by replacing their algorithmic parameter α with $-\alpha$ [41, §8.2].

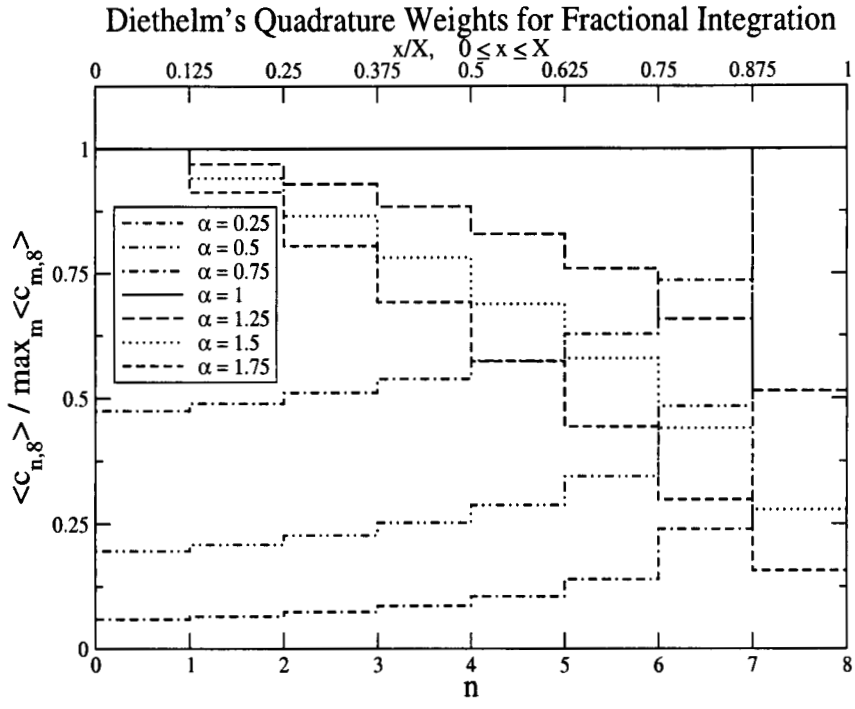

Fig. 4. Effective weights of quadrature $\langle c_{n,N} \rangle$ for approximating the Riemann-Liouville fractional integral (2) over interval $[0, X]$ using Alg. 2, plotted here for various values of α with $N = 8$.

The line segments displayed in Fig. 4 represent normalized weights of quadrature averaged over each subinterval. The actual nodal weights, $c_{n,N}$, are often observed to be non-monotonic at either of the two nodal endpoints. In this integration scheme there are $N + 1$ nodal weights that apply to N subintervals, but there should be exactly one weight per subinterval. So how the algorithm works (internally, and roughly speaking) is to average these weights in a trapezoidal fashion, as outlined in Table 6.2. In other words, the inner weights are divided into two equal halves with each half going to one of the two adjoining subintervals. In addition to averaging, the displayed line segments have been normalized over the interval $[0, 1]$. Normalizing allows one to discern the influence of α on quadrature in a meaningful way. The outcome is an averaged and normalized quadrature weighting that is monotonic in the nodal index number, as demonstrated by the line segments in Fig. 4, where there is a monotonic increase (decrease) in the effective weight of quadrature, $\langle c_{n,N} \rangle \; / \; \max_m \langle c_{m,N} \rangle$, with increasing nodal index number for $0 < \alpha < 1$ $(1 < \alpha < 2)$.

6.3 Optimizing performance

In a recent paper, Ford and Simpson [16] developed a faster scheme for the calculation of fractional integrals. A reduction in the amount of computational

work can be achieved by using a graded mesh, thereby taking the $O(N^2)$ method of Alg. 2 and making it $O(N \log N)$. The underlying idea is based on the fact that the fractional integral possesses a fundamental scaling property that can be exploited in a natural way.

If $q > 0$ and the function $\mathbf{y}$ is suitably integrable then, with J^α defined as before, we observe that

$$J^\alpha \mathbf{y}(qx) = \frac{1}{\Gamma(\alpha)} \int_0^{qx} \frac{\mathbf{y}(x')}{(qx - x')^{1-\alpha}} \, \mathrm{d}x'. \tag{16}$$

We make the substitution of x' by qx' to obtain

$$J^\alpha \mathbf{y}(qx) = \frac{q^\alpha}{\Gamma(\alpha)} \int_0^{x} \frac{\mathbf{y}(qx')}{(x - x')^{1-\alpha}} \, \mathrm{d}x', \tag{17}$$

to which it follows that, for any $p \in \mathbb{N}$,

$$J^\alpha \mathbf{y}(q^p x) = \frac{q^{p\alpha}}{\Gamma(\alpha)} \int_0^{x} \frac{\mathbf{y}(q^p x')}{(x - x')^{1-\alpha}} \, \mathrm{d}x'. \tag{18}$$

Thus, the weights ω_i necessary to calculate $\Omega_h^\alpha \mathbf{y}(nh) \approx J^\alpha \mathbf{y}(nh)$ (using a step length of h) can be used to calculate $\Omega_{q^p h}^\alpha \mathbf{y}(nq^p h) \approx J^\alpha \mathbf{y}(nq^p h)$ (using a step length of $q^p h$) for any $p \in \mathbb{N}$ simply by multiplying the resulting sum by $q^{p\alpha}$. That is,

$$\Omega_h^\alpha \mathbf{y}(nh) = \sum_{j=0}^{n} \omega_{n-j} \mathbf{y}(jh) \Leftrightarrow \Omega_{q^p h}^\alpha \mathbf{y}(nq^p h) = q^{p\alpha} \sum_{j=0}^{n} \omega_{n-j} \mathbf{y}(jq^p h). \tag{19}$$

We use a sequence of nested meshes in the following way: for $h \in \mathbb{R}_+$, we define the mesh M_h by $M_h = \{hn,\ n \in \mathbb{N}\}$. If $p, q, r \in \mathbb{N},\ q > 0,\ r > p$, then $M_{q^p h} \supset M_{q^r h}$. We then decompose the interval $[0, x]$, for fixed $X > 0$, in the following way:

$$[0, x] = [0, x - q^m X] \cup [x - q^m X, x - q^{m-1} X] \cup \cdots \cup [x - qX, x - X] \cup [x - X, x],$$

where $m \in \mathbb{N}$ is the smallest integer such that $x < q^{m+1} X$.

Now, as is clear from the definition of J^α, the singularity in the kernel of the fractional derivative (integral) occurs when $x = x'$, while the value of the kernel $1/(x - x')^{1-\alpha} \to 0$ as $x \to \infty$ for $x' \to 0$. We combine this insight with the scaling property, and we use a step length of h over the most recent time interval $[x - X, x]$ and successively larger step lengths over earlier intervals in the following way: let $x,\ X,\ h \in \mathbb{R},\ q^{m+1} X > x \geq q^m X,\ X > 1,\ h > 0$ with $x = nh$ for some $n \in \mathbb{N}$. We can therefore rewrite the Riemann-Liouville

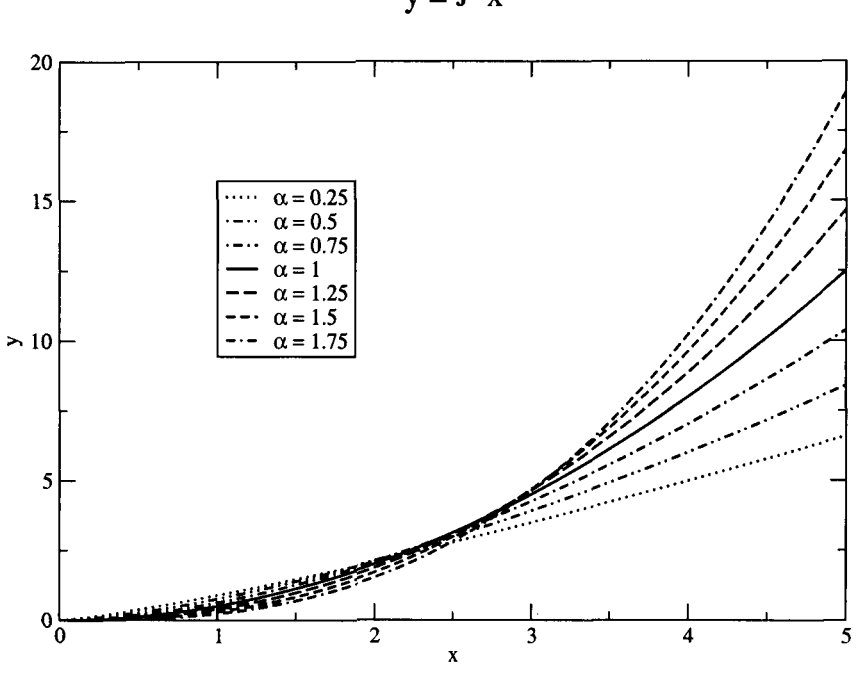

Fig. 5. The Riemann-Liouville integral $J^\alpha x$ for various values of α.

fractional integral in a form suitable for efficient computation; it being,

$$J^\alpha_{[0,x]}\mathbf{y}(x) = J^\alpha_{[x-X,x]}\mathbf{y}(x) + \sum_{i=0}^{m-1} J^\alpha_{[x-q^{i+1}X,x-q^iX]}\mathbf{y}(x) + J^\alpha_{[0,x-q^mX]}\mathbf{y}(x)$$

$$= J^\alpha_{[x-X,x]}\mathbf{y}(x) + \sum_{i=0}^{m-1} q^{i\alpha} J^\alpha_{[x-q^{i+1}X,x-q^iX]}\mathbf{y}(q^ix) + q^{m\alpha} J^\alpha_{[0,x-q^mX]}\mathbf{y}(q^mx). \quad (20)$$

This reduces the computational effort required, both to calculate the weights we use (since now we need fewer different weights) and (more importantly) to evaluate the integral. We employ the same weights that are listed in Alg. 2, but the above scheme is independent of the algorithm used for integration.

6.4 Examples

To complement Fig. 2, consider $J^\alpha\mathbf{y}(x)$ where $\mathbf{y}(x) = x$, which is plotted in Fig. 5 for values of α that vary between 0 and 2. The well-known solution $J^1x = \frac{1}{2}x^2$ bisects the displayed family of curves, while the fractional (in α) solutions smoothly traverse between the integer solutions.

Our second example for fractional integration is akin to our second example for Caputo differentiation, viz., $J^\alpha\mathbf{y}(x)$ where $\mathbf{y}(x) = \mathrm{e}^{-x}$, which is plotted in Fig. 6. The curves in this figure are essentially mirror images of those presented earlier in Fig. 3 for $D^\alpha_\star\,\mathrm{e}^{-x}$, except that the α-ordering is reversed.

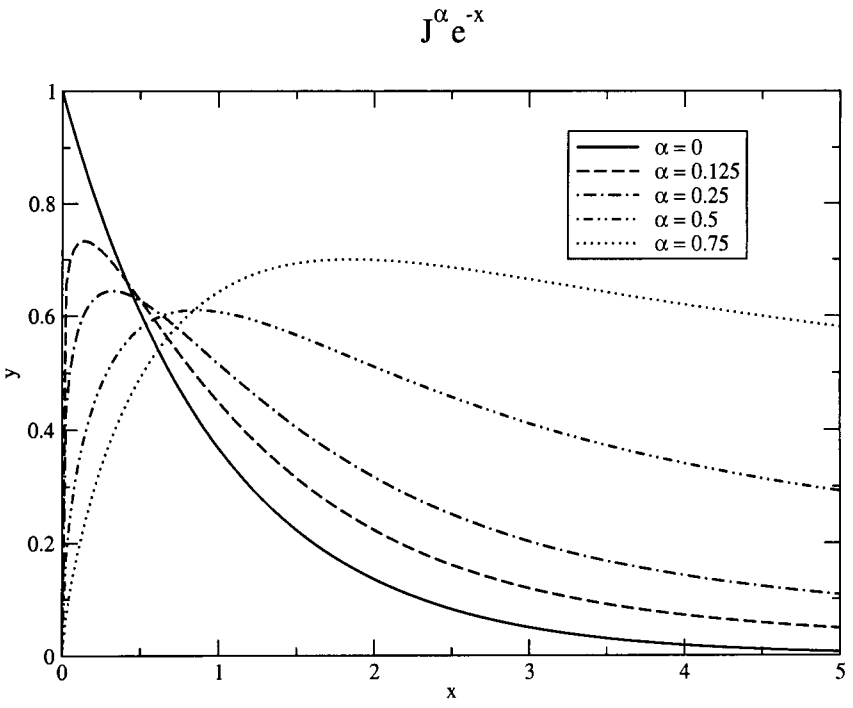

Fig. 6. The Riemann-Liouville integral $J^{\alpha}\, \mathrm{e}^{-x}$ for various values of α.

7 Numerical Approximation of Caputo-Type FDE's

7.1 A basic general-purpose algorithm

A numerical algorithm that solves Caputo-type FDE's has been derived by Diethelm et al. [10] and is listed in Alg. 3. A thorough analysis of its algorithmic error is given in [9]. This algorithm is of the PECE (Predict-Evaluate-Correct-Evaluate) type. Other numerical algorithms exist that solve FDE's (e.g., Gorenflo [21] and Podlubny [45, Chp. 8]), but they focus on solving Riemann-Liouville FDE's and usually restrict the class of FDE's to be linear with homogeneous initial conditions. Algorithm 3 solves non-linear Caputo FDE's with inhomogeneous initial conditions, if required.

The restriction that $\alpha \neq 1$ in this algorithm is purely for formal reasons. If $\alpha = 1$, then we can still implement the algorithm exactly in the indicated way. It must be noted, however, that it then is the limit case of an algorithm for *fractional* differential equations, and these equations involve *non-local* differential operators. Thus, the resulting scheme is non-local, too. In contrast, a method constructed specifically for a first-order equation will, in practice, always make explicit use of the *local* structure of such an equation to save memory and computing time. Therefore, the case $\alpha = 1$ of our algorithm will never be a competitive alternative to the usual methods for first-order equations. In particular, our algorithm is distinct from the algorithm known

as the second-order Adams-Bashforth-Moulton technique for first-order problems when $\alpha = 1$.

Algorithm 3
Computation of a Caputo FDE ($0 < \alpha < 2$, $\alpha \neq 1$):
For interval $[0, X]$ *with grid* $\{x_n = nh\colon n = 0, 1, 2, \ldots, N\colon h = X/N\}$,
predict with

$$\mathbf{y}_N^P(h) = \sum_{k=0}^{\lfloor \alpha \rfloor} (X^k/k!)\mathbf{y}_{0+}^{(k)} + [h^\alpha/\Gamma(1+\alpha)] \sum_{n=0}^{N-1} b_{n,N}\, \mathbf{f}(x_n, \mathbf{y}_n) \ ,$$

using the quadrature weights (derived from a product rectangular rule)

$$b_{n,N} = (N-n)^\alpha - (N-n-1)^\alpha \ ,$$

and evaluate $\mathbf{f}(X, \mathbf{y}_N^P)$, *then correct with*

$$\mathbf{y}_N(h) = \sum_{k=0}^{\lfloor \alpha \rfloor} (X^k/k!)\mathbf{y}_{0+}^{(k)} + [h^\alpha/\Gamma(2+\alpha)] \left(\sum_{n=0}^{N-1} c_{n,N}\, \mathbf{f}(x_n, \mathbf{y}_n) + c_{N,N}\, \mathbf{f}(X, \mathbf{y}_N^P) \right) \ ,$$

$$\mathbf{y}(X) = \mathbf{y}_N(h) + O(h^{\min(1+\alpha, 2)}) \ ,$$

using the quadrature weights (derived from a product trapezoidal rule)

$$c_{n,N} = \begin{cases} (1+\alpha)N^\alpha - N^{1+\alpha} + (N-1)^{1+\alpha}, & \text{if } n = 0, \\ (N-n+1)^{1+\alpha} - 2(N-n)^{1+\alpha} + (N-n-1)^{1+\alpha}, & \text{if } 0 < n < N, \\ 1, & \text{if } n = N, \end{cases}$$

and finally, re-evaluate $\mathbf{f}(X, \mathbf{y}_N)$ *saving it as* $\mathbf{f}(x_N, \mathbf{y}_N)$.
Refine, if desired, using Richardson extrapolation

$$\mathbf{y}_v^u(X) = \left(\mathbf{y}_{v-1}^{u-1}(X) - 2^{r_{u-1}}\, \mathbf{y}_v^{u-1}(X) \right) / (1 - 2^{r_{u-1}}) \ ,$$

$$\mathbf{y}(X) = \mathbf{y}_u^u(X) + O(h^{r_u}) \ ,$$

such that whenever $0 < \alpha < 1$ *the exponent* r_{u-1} *is assigned as*

$$r_0 = 1 + \alpha,$$

$$r_1 = 2,\ r_2 = 2 + \alpha,\ r_3 = 3 + \alpha,$$

$$r_4 = 4,\ r_5 = 4 + \alpha,\ r_6 = 5 + \alpha,$$

$$r_7 = 6, \cdots,$$

or whenever $1 < \alpha < 2$ *it is assigned as*

$$r_0 = 2,\ r_1 = 1 + \alpha,\ r_2 = 2 + \alpha,$$

$$r_3 = 4,\ r_4 = 3 + \alpha,\ r_5 = 4 + \alpha,$$

$$r_6 = 6, \cdots.$$

7.2 Quadratures

Illustrations of quadrature weight versus nodal index for several values of α, as they pertain to the PECE method of Alg. 3, are presented in Figs. 4 & 7

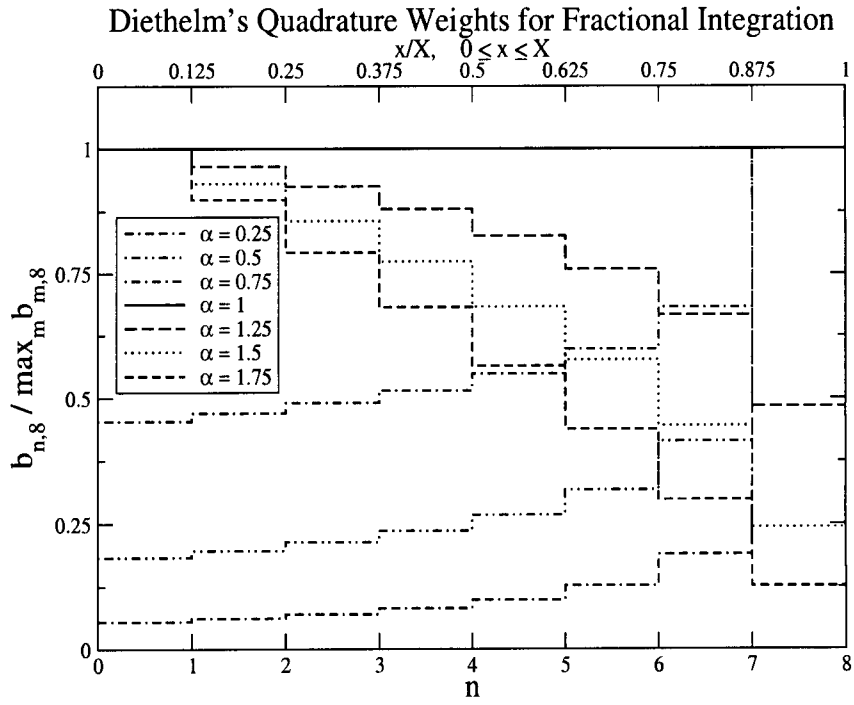

Fig. 7. Normalized weights of quadrature $b_{n,N}$ for the predictor that approximates Caputo FDE's (13) over interval $[0, X]$ when using Diethelm et al.'s [10] Alg. 3, plotted here for various values of α with $N = 8$.

with the former figure pertaining to the corrector and the latter pertaining to the predictor. FDE's, like fractional integrals, exhibit long-term memory loss when $0 < \alpha < 1$, no memory loss when $\alpha = 1$, and short-term memory loss when $1 < \alpha < 2$.

Unlike the $\langle c_{n,N} \rangle$ in Fig. 4, where the $N+1$ quadrature weights are averaged at the beginning and end of each subinterval in order to get N effective weights for N subintervals, the $b_{n,N}$ in Fig. 7 are fixed to the beginning of each of the N subintervals, and as such, do not require any 'effective averaging' to take place. This is a consequence of the $b_{n,N}$ quadrature weights belonging to an explicit integrator, while the $c_{n,N}$ weights belong to an implicit integrator. Contrasting Figs. 4 & 7, there is little difference between the $b_{n,N}$ and $\langle c_{n,N} \rangle$ curves, indicating that there is a much stronger influence of α on the weights of quadrature than there is on the order of accuracy (e.g., $O(h^{\min(2,1+\alpha)})$) that a particular integration scheme provides.

Quadratures for non-equidistant nodes have also been derived by Diethelm and Freed [11], but they are vastly more expensive to compute, and as such, are not the preferred quadratures to use.

Differential equations of fractional order have found recent applications in a variety of fields in science and engineering (e.g., see references in [27,32,45]): chemical kinetics theory; electromagnetic theory; transport (diffusion) theory;

fractals, splines, and wavelets; control theory; electronic circuit theory; porous media; etc.

7.3 Efficient approximations

The nested mesh algorithm for the evaluation of fractional integrals given by Ford and Simpson [16] and described in §6.3 can also be adapted to provide an effective fast algorithm for the solution of Caputo FDE's. Whereas, the unmodified method possesses $O(N)$ operation counts at each stage and $O(N^2)$ overall, this modification leads to a more efficient scheme with $O(N \log N)$ counts overall, while retaining the accuracy of the method.

7.4 Examples

Many physical processes can be represented by a quantity that begins at some nominal value, and later asymptotes to some saturated state under steady conditions, e.g., we [11] have used such an equation to describe the evolution of back stress (an internal state variable) in viscoplasticity. This effect can be modeled by the differential equation $D^{\alpha}_{\star}\mathbf{y}(x) = 1 - \mathbf{y}(x)$ whose solution is presented in Fig. 8 for values of α less than 1, and in Fig. 9 for values of α greater than 1. The solution will be monotone provided that $0 < \alpha \leq 1$; it will be a damped oscillator whenever $1 < \alpha < 2$; it will be a perfect oscillator if $\alpha = 2$; and it will become an unstable oscillator for any value of $\alpha > 2$.

8 Mittag-Leffler Function

Analytic solutions to fractional-order differential equations are often expressed in terms of the Mittag-Leffler function, and so we also need a scheme that computes this special function, which is the topic of this section.

8.1 Fundamental definitions

The (generalized) Mittag-Leffler function $E_{\alpha,\beta}(z)$ is an entire function (in $z \in \mathbb{C}$) of order $1/\alpha$ that is defined by the power series [15, §18.1]

$$E_{\alpha,\beta}(z) = \sum_{k=0}^{\infty} \frac{z^k}{\Gamma(\beta + \alpha k)}, \quad \alpha \in \mathbb{R}_+,\ \beta \in \mathbb{R},\ z \in \mathbb{C}, \tag{21}$$

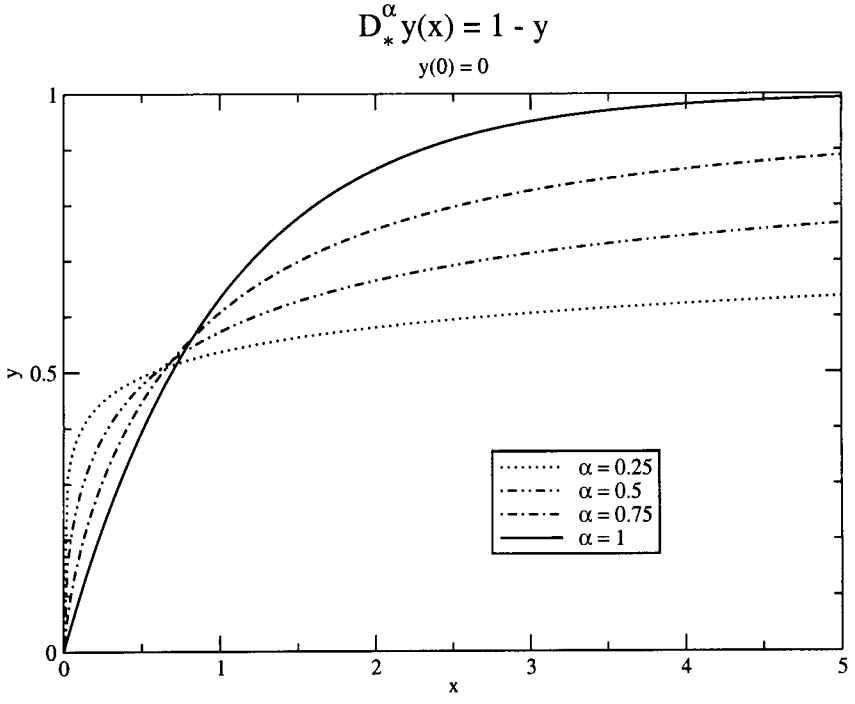

Fig. 8. The solution of the Caputo differential equation $D^{\alpha}_{\star}\mathbf{y}(x) = 1 - y(x)$ for various values of $\alpha \in (0, 1]$ satisfying an initial condition of $\mathbf{y}(0) = 0$.

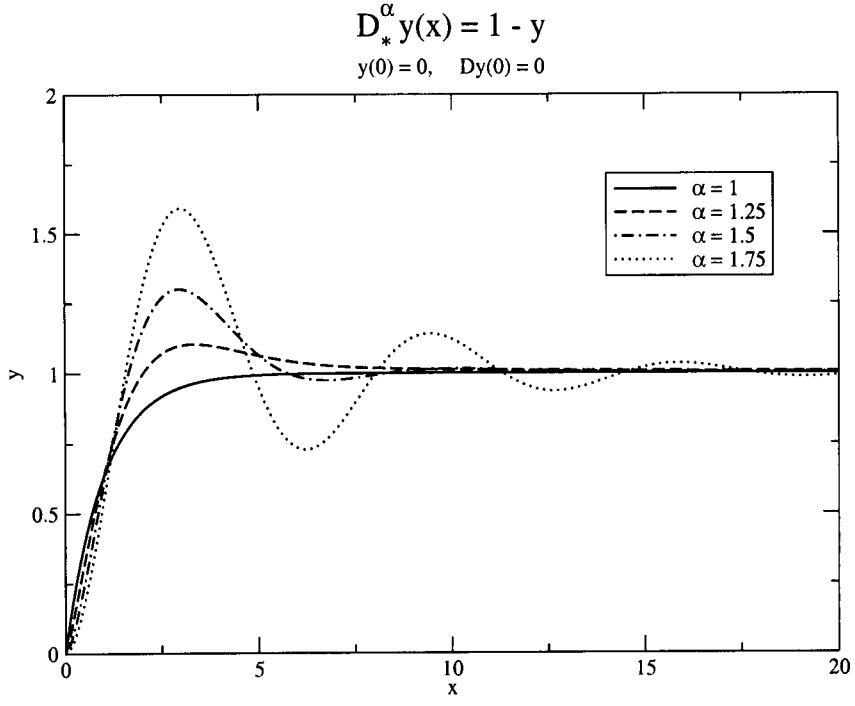

Fig. 9. The solution of the Caputo differential equation $D^{\alpha}_{\star}\mathbf{y}(x) = 1 - y(x)$ for various values of $\alpha \in [1, 2)$ satisfying initial conditions $\mathbf{y}(0) = 0$ and $D\mathbf{y}(0) = 0$.

whose derivative is

$$E'_{\alpha,\beta}(z) = \frac{\mathrm{d}E_{\alpha,\beta}(z)}{\mathrm{d}z} = \sum_{k=0}^{\infty} \frac{(1+k)z^k}{\Gamma[\beta+\alpha(1+k)]}, \tag{22}$$

with $E_\alpha(x) := E_{\alpha,1}(x)$ being the original function studied by Mittag-Leffler [38]. This function plays the same role in differential equations of fractional order that the exponential function e^z plays in ordinary differential equations; in fact, $E_{1,1}(z) = \mathrm{e}^z$.

A special form of the Mittag-Leffler function,

$$G(t-t') := E_{\alpha,1}\Big(-[(t-t')/\tau]^\alpha\Big), \quad 0<\alpha<1, \quad 0<\tau, \quad 0 \le t' \le t,$$

and its derivative,

$$M(t-t') := \frac{\partial G(t-t')}{\partial t'} = \frac{-E_{\alpha,0}(-[(t-t')/\tau]^\alpha)}{t-t'},$$

appear in the theory of fractional-order viscoelasticity (FOV) [17]; it is also of major significance in any application using linear FDE's with constant coefficients [23,24].

We now present some important properties of the Mittag-Leffler function, and a numerical algorithm for its accurate evaluation, both of which are useful when considering differential equations of fractional order.

8.2 Analytical properties

In spite of the fact that in applications to differential equations of fractional order the Mittag-Leffler function is typically restricted to the real line, we still need to give some of its properties in the complex plane. The main reason for this is that the numerical algorithm we present in the following pages consists of two parts: the first part gives a numerical value for the Mittag-Leffler function with $\alpha \le 1$, while the second one uses, for $\alpha > 1$, some special formulæ that reduce this case to the previous one. These special formulæ are defined over the complex plane and are given by

$$E_{\alpha,\beta}(z) = \frac{1}{2m+1} \sum_{h=-m}^{m} E_{\alpha/(2m+1),\beta}(z^{1/(2m+1)}\, \mathrm{e}^{\mathrm{i}2\pi h/(2m+1)}), \ m = 0,1,2,\ldots, \tag{23}$$

and

$$E_{\alpha,\beta}(z) = \frac{1}{m} \sum_{h=0}^{m-1} E_{\alpha/m,\beta}(z^{1/m}\, \mathrm{e}^{\mathrm{i}2\pi h/m}), \ m = 1,2,\ldots, \tag{24}$$

where i ($:= \sqrt{-1}$) is the imaginary unit number. Obviously, if $\alpha > 1$, and even if z is a real number, we still need to evaluate the numerical values of the Mittag-Leffler function with, in general, a complex argument (and $\alpha \leq 1$).

First, we present some important integral representations of the Mittag-Leffler function. Let us denote by $\gamma(\rho;\varphi)$ ($\rho > 0$, $0 < \varphi \leq \pi$) a contour in the complex λ-plane with non-decreasing $\arg\lambda$ consisting of the following parts:

(1) the ray $\arg\lambda = -\varphi$, $|\lambda| \geq \rho$;
(2) the arc $-\varphi \leq \arg\lambda \leq \varphi$ from the circumference $|\lambda| = \rho$;
(3) the ray $\arg\lambda = \varphi$, $|\lambda| \geq \rho$.

In the case where $0 < \varphi < \pi$, the complex λ-plane is divided into two unbounded parts by the contour $\gamma(\rho;\varphi)$: domain $G^{(-)}(\rho;\varphi)$ is to the left of the contour, while domain $G^{(+)}(\rho;\varphi)$ is to its right. If $\varphi = \pi$, the contour $\gamma(\rho;\varphi)$ consists of the circumference $|\lambda| = \rho$ and the cut $-\infty < \lambda \leq -\rho$. In this case, domain $G^{(-)}(\rho;\varphi)$ becomes the circle $|\lambda| < \rho$, while domain $G^{(+)}(\rho;\varphi)$ becomes the region $\{\lambda : |\arg\lambda| < \pi,\ |\lambda| > \rho\}$.

Let $0 < \alpha < 2$, let β be an arbitrary (real or complex) number, and let a non-negative number φ be chosen such that

$$\frac{\alpha\pi}{2} < \varphi \leq \min\{\pi, \alpha\pi\}. \tag{25}$$

Then we have the following integral representations for the Mittag-Leffler function:

$$E_{\alpha,\beta}(z) = \frac{1}{\mathrm{i}2\pi\alpha} \int_{\gamma(\rho;\varphi)} \frac{\mathrm{e}^{\lambda^{1/\alpha}}\, \lambda^{(1-\beta)/\alpha}}{\lambda - z}\, \mathrm{d}\lambda,\ z \in G^{(-)}(\rho;\varphi), \tag{26}$$

and

$$E_{\alpha,\beta}(z) = \frac{z^{(1-\beta)/\alpha}\, \mathrm{e}^{z^{1/\alpha}}}{\alpha} + \frac{1}{\mathrm{i}2\pi\alpha} \int_{\gamma(\rho;\varphi)} \frac{\mathrm{e}^{\lambda^{1/\alpha}}\, \lambda^{(1-\beta)/\alpha}}{\lambda - z}\, \mathrm{d}\lambda,\ z \in G^{(+)}(\rho;\varphi). \tag{27}$$

If β is a real number, as assigned in Eq. (21), then the formulæ of Eqs. (26 & 27) can be rewritten in forms that are more suitable for numerical evaluation (cf. Gorenflo et al. [22]). In particular, if $0 < \alpha \leq 1$, $\beta \in \mathbb{R}$, $|\arg z| > \alpha\pi$, $z \neq 0$, then

$$E_{\alpha,\beta}(z) = \int_{\rho}^{\infty} K(\alpha,\beta,\chi,z)\, \mathrm{d}\chi + \int_{-\alpha\pi}^{\alpha\pi} P(\alpha,\beta,\rho,\varphi,z)\, \mathrm{d}\varphi,\ \rho > 0,\ \beta \in \mathbb{R}, \tag{28}$$

$$E_{\alpha,\beta}(z) = \int_{0}^{\infty} K(\alpha,\beta,\chi,z)\, \mathrm{d}\chi,\ \text{if } \beta < 1 + \alpha, \tag{29}$$

$$E_{\alpha,\beta}(z) = -\frac{1}{z} - \frac{\sin(\alpha\pi)}{\alpha\pi} \int_{0}^{\infty} \frac{\mathrm{e}^{-\chi^{1/\alpha}}}{\chi^2 - 2\chi z\cos(\alpha\pi) + z^2}\, \mathrm{d}\chi,\ \text{if } \beta = 1 + \alpha, \tag{30}$$

wherein

$$K(\alpha,\beta,\chi,z) = \frac{\chi^{(1-\beta)/\alpha}\,\mathrm{e}^{-\chi^{1/\alpha}}}{\alpha\pi}\,\frac{\chi\sin[\pi(1-\beta)] - z\sin[\pi(1-\beta+\alpha)]}{\chi^2 - 2\chi z\cos(\alpha\pi) + z^2},$$

$$P(\alpha,\beta,\rho,\varphi,z) = \frac{\rho^{1+(1-\beta)/\alpha}}{2\alpha\pi}\,\frac{\mathrm{e}^{\rho^{1/\alpha}\cos(\varphi/\alpha)}[\cos(\omega) + \mathrm{i}\sin(\omega)]}{\rho\,\mathrm{e}^{\mathrm{i}\varphi} - z},$$

$$\omega = \rho^{1/\alpha}\sin(\varphi/\alpha) + \varphi[1 + (1-\beta)/\alpha].$$

The representations in Eqs. (28–30), and similar formulæ for the cases $|\arg z| = \alpha\pi$ and $|\arg z| < \alpha\pi$ presented in Gorenflo et al. [22], are an essential part of the numerical algorithm listed below.

Using the integral representations in Eqs. (26 & 27), it is not too difficult to get asymptotic expansions for the Mittag-Leffler function in the complex plane. Let $\alpha < 2$, β be an arbitrary number, and φ be choosen to satisfy the condition in Eq. (25). Then we have, for any $p \in \mathbb{N}$ and $|z| \to \infty$:

(1) whenever $|\arg z| \le \varphi$,

$$E_{\alpha,\beta}(z) = \frac{z^{(1-\beta)/\alpha}\,\mathrm{e}^{z^{1/\alpha}}}{\alpha} - \sum_{k=1}^{p}\frac{z^{-k}}{\Gamma(\beta-\alpha k)} + O(|z|^{-1-p}). \tag{31}$$

(2) whenever $\varphi \le |\arg z| \le \pi$,

$$E_{\alpha,\beta}(z) = -\sum_{k=1}^{p}\frac{z^{-k}}{\Gamma(\beta-\alpha k)} + O(|z|^{-1-p}). \tag{32}$$

These formulæ are also used in our numerical algorithm.

Thorough discussions of properties of the Mittag-Leffler function can be found, for example, in Refs. [15,33,45].

8.3 Numerical algorithms

Based on these theoretical results, a numerical scheme listed in Alg. 4 for computing the general Mittag-Leffler function $E_{\alpha,\beta}(z)$ has been developed by Gorenflo et al. [22], which is reproduced here with corrections. Also presented is a scheme for computing the derivative of the Mittag-Leffler function $\mathrm{d}E_{\alpha,\beta}(z)/\mathrm{d}z$, it being Alg. 5, that we use as a memory function in our theory of fractional-order viscoelasticity [18].

Algorithm 4
Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z) + \mu(z)$, $|\mu(z)| < \epsilon$:
given $\alpha > 0$, $\beta \in \mathbb{R}$, $z \in \mathbb{C}$, $\epsilon > 0$, $0 < \zeta < 1$ **then**
if $z = 0$ **then**
$E_{\alpha,\beta}(z) = 1/\Gamma(\beta)$
elsif $\alpha \leq 1$ **then**
if $|z| < \zeta$ **then**
$k_0 = \max\{\lceil (1-\beta)/\alpha \rceil, \lceil \ln[\epsilon(1-|z|)]/\ln(|z|) \rceil\}$
$E_{\alpha,\beta}(z) = \sum_{k=0}^{k_0} z^k/\Gamma(\beta + \alpha k)$
elsif $|z| < \lfloor 10 + 5\alpha \rfloor$ **then**
$\chi_0 = \begin{cases} \max\{1,\ 2|z|,\ [-\ln(\epsilon\pi/6)]^\alpha\}, & \beta \geq 0 \\ \max\{(|\beta|+1)^\alpha,\ 2|z|,\ [-2\ln(\epsilon\pi/[6(|\beta|+2)(2|\beta|)^{|\beta|}])]^\alpha\}, & \beta < 0 \end{cases}$
$K(\alpha,\beta,\chi,z) = \frac{1}{\alpha\pi}\chi^{(1-\beta)/\alpha}\exp(-\chi^{1/\alpha})\frac{\chi\sin[\pi(1-\beta)] - z\sin[\pi(1-\beta+\alpha)]}{\chi^2 - 2\chi z\cos(\alpha\pi) + z^2}$
$\omega = \phi[1 + (1-\beta)/\alpha] + \rho^{1/\alpha}\sin(\phi/\alpha)$
$P(\alpha,\beta,\rho,\phi,z) = \frac{1}{2\alpha\pi}\rho^{1+(1-\beta)/\alpha}\exp[\rho^{1/\alpha}\cos(\phi/\alpha)]\frac{\cos(\omega) + \mathrm{i}\sin(\omega)}{\rho\exp(\mathrm{i}\phi) - z}$
if $|\arg z| > \alpha\pi$ **and** $||\arg z| - \alpha\pi| > \epsilon$ **then**
if $\beta < 1 + \alpha$ **then**
$E_{\alpha,\beta}(z) = \int_0^{\chi_0} K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi$
else
$E_{\alpha,\beta}(z) = \int_1^{\chi_0} K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi + \int_{-\alpha\pi}^{\alpha\pi} P(\alpha,\beta,1,\phi,z)\,\mathrm{d}\phi$
elsif $|\arg z| < \alpha\pi$ **and** $||\arg z| - \alpha\pi| > \epsilon$ **then**
if $\beta < 1 + \alpha$ **then**
$E_{\alpha,\beta}(z) = \int_0^{\chi_0} K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi + \frac{1}{\alpha}z^{(1-\beta)/\alpha}\,\mathrm{e}^{z^{1/\alpha}}$
else
$E_{\alpha,\beta}(z) = \int_{|z|/2}^{\chi_0} K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi + \int_{-\alpha\pi}^{\alpha\pi} P(\alpha,\beta,\frac{1}{2}|z|,\phi,z)\,\mathrm{d}\phi + \frac{1}{\alpha}z^{(1-\beta)/\alpha}\,\mathrm{e}^{z^{1/\alpha}}$
else
$E_{\alpha,\beta}(z) = \int_{|z|+1/2}^{\chi_0} K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi + \int_{-\alpha\pi}^{\alpha\pi} P(\alpha,\beta,|z|+\frac{1}{2},\phi,z)\,\mathrm{d}\phi$
else
$k_0 = \lfloor -\ln(\epsilon)/\ln(|z|) \rfloor$
if $|\arg z| < 3\alpha\pi/4$ **then**
$E_{\alpha,\beta}(z) = \frac{1}{\alpha}z^{(1-\beta)/\alpha}\,\mathrm{e}^{z^{1/\alpha}} - \sum_{k=1}^{k_0} z^{-k}/\Gamma(\beta - \alpha k)$
else
$E_{\alpha,\beta}(z) = -\sum_{k=1}^{k_0} z^{-k}/\Gamma(\beta - \alpha k)$
elsif $1 < \alpha < 2$ **then**
$E_{\alpha,\beta}(z) = [E_{\alpha/2,\beta}(\sqrt{z}) + E_{\alpha/2,\beta}(-\sqrt{z})]/2$
else
$k_0 = \lfloor \alpha/2 \rfloor + 1$
$E_{\alpha,\beta}(z) = \sum_{k=0}^{k_0-1} E_{\alpha/k_0,\beta}[z^{1/k_0}\exp(\mathrm{i}2\pi k/k_0)]/k_0$
end

Algorithm 5
Computing the derivative of the Mittag-Leffler function $\mathrm{d}E_{\alpha,\beta}(z)/\,\mathrm{d}z = E'_{\alpha,\beta}(z) + \mu(z)$, $|\mu(z)| < \epsilon$:
given $\alpha > 0$, $\beta \in \mathbb{R}$, $z \in \mathbb{C}$, $\epsilon > 0$, $0 < \zeta < 1$ **then**
if $z = 0$ **then**
$E'_{\alpha,\beta}(z) = 1/\Gamma(\alpha+\beta)$
elsif $|z| < \zeta$ **then**
$\omega = \alpha + \beta - 3/2$
$D = \alpha^2 - 4\alpha\beta + 6\alpha + 1$

$$k_1 = \begin{cases} 1 + (2-\alpha-\beta)/(\alpha-1), & \alpha > 1 \\ 1 + (3-\alpha-\beta)/\alpha, & 0 < \alpha \le 1,\ D \le 0 \\ \max\left\{1 + (3-\alpha-\beta)/\alpha,\, 1 + (1 - 2\omega\alpha + \sqrt{D})/(2\alpha^2)\right\}, & 0 < \alpha \le 1,\ D > 0 \end{cases}$$

$k_0 = \max\{\lceil k_1 \rceil, \lceil \ln[\epsilon(1-|z|)]/\ln(|z|)\rceil\}$
$E'_{\alpha,\beta}(z) = \sum_{k=0}^{k_0} (1+k) z^k / \Gamma[(1+k)\alpha + \beta]$
else
$E'_{\alpha,\beta}(z) = [E_{\alpha,\beta-1}(z) - (\beta-1)E_{\alpha,\beta}(z)]/(\alpha z)$
end

Note 2 *In Algs. 4 & 5, error tolerance ϵ should be set at ε_m = machine precision, while parameter ζ should be set around 0.9 to keep k_0 reasonably sized.*

8.3.1 Essentials of the construction

Algorithm 4 uses the defining series (Eq. 21) for arguments of small magnitude, its asymptotic representation (Eqs. 31 & 32) for arguments of large magnitude, and special integral representations (the formulæ in Eqs. (28–30) for the case where $|\arg z| > \alpha\pi$, and similar representations for the cases $|\arg z| = \alpha\pi$ and $|\arg z| < \alpha\pi$) for intermediate values of the argument that include a monotonic part $\int K(\alpha,\beta,\chi,z)\,\mathrm{d}\chi$ and an oscillatory part $\int P(\alpha,\beta,\rho,\phi,z)\,\mathrm{d}\phi$, which can themselves be evaluated using standard techniques (cf. App. B). These special integral representations are needed because the above series can become numerically unstable in this intermediate region where these integrals are employed. The outcome is a robust method for computing the Mittag-Leffler function.

8.3.2 Efficiency

Algorithms 4 & 5 can produce numerical results to any desired level of accuracy. These computations can be expensive, and therefore their use in a finite element setting, for example, may be prohibitive. To help meet this need, we

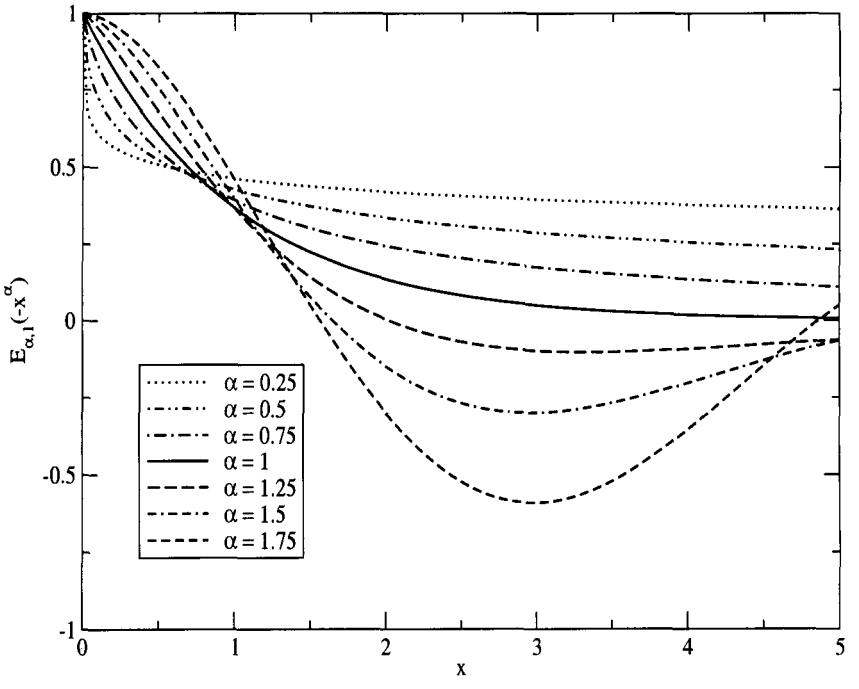

Fig. 10. The Mittag-Leffer function $E_{\alpha,\beta}(-x^{\alpha})$ for $\beta = 1$ and $\alpha \in (0, 2)$.

have constructed a table of Padé approximates for $E_{\alpha,1}(-x^{\alpha})$ in App. C for $x \geq 0$ and $\alpha \in \{0.01, 0.02, 0.03, \ldots, 0.98, 0.99\}$. As mentioned above, this form of the Mittag-Leffler function arises in many fractional-order models.

Another scheme based on Eqs. (21 & 32) for solving the Mittag-Leffler function $E_{\alpha,1}(x)$ ($0.02 < \alpha < 0.98$ with a reported relative error that is less than 1.6×10^{-5}) has been published by Welch et al. [54].

8.4 Examples

Variants of the Mittag-Leffler function $E_{\alpha,\beta}(-x^{\alpha})$ arise naturally in solutions of linear, fractional-order, differential equations. In Fig. 10, plots of this function are provided for $\beta = 1$ and $\alpha \in (0, 2)$, while in Fig. 11, plots are given for $\alpha = 1$ and $\beta \in (0, 2)$. Recall that $E_{1,1}(-x) = \mathrm{e}^{-x}$. Parameter α is seen to have a strong influence on the overall shape of the curve, while β has its most pronounced influence on the value of the function at $x = 0$, in this particular case. Numerous example figures can also be found in Gorenflo et al. [22].

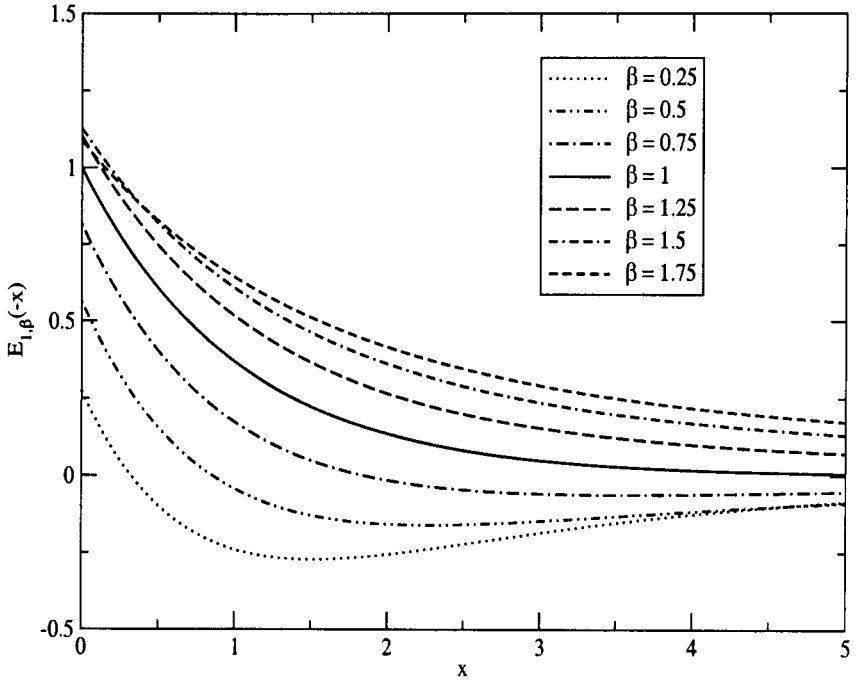

Fig. 11. The Mittag-Leffer function $E_{\alpha,\beta}(-x^{\alpha})$ for $\alpha = 1$ and $\beta \in (0, 2)$.

9 Summary

In this paper we provide numerical algorithms that solve fractional-order integrals, derivatives of the Caputo type, and differential equations, also of the Caputo type. We also present an algorithm that evaluates the Mittag-Leffler function and its derivative, which are present in analytic solutions to fractional-order differential equations. Collectively, these methods constitute a toolbox that engineers can use to address problems where the fractional calculus has been used to model some aspect of reality.

A Table of Caputo Derivatives

For the convenience of the reader, we provide this appendix where we give some Caputo-type derivatives of certain important functions.[7] We do not strive for completeness in any sense, but we do want to give at least the derivatives of classic examples.

Throughout this appendix, α will always denote the order of the Caputo-type differential operator under consideration. We shall only consider the case $\alpha > 0$ and $\alpha \notin \mathbb{N}$, where $\mathbb{N} := \{1, 2, 3, \ldots\}$ while $\mathbb{N}_0 := \{0, 1, 2, \ldots\}$. We use the ceiling function $\lceil \alpha \rceil$ to denote the smallest integer greater than or equal to α, and the floor function $\lfloor \alpha \rfloor$ to denote the largest integer less than or equal to α. Recall that for $\alpha \in \mathbb{N}$, the Caputo differential operator coincides with the usual differential operator of integer order.

Moreover, as in Eq. (21), $E_{\alpha,\beta}$ denotes the Mittag-Leffler function in two parameters α and $\beta > 0$, given by

$$E_{\alpha,\beta}(t) := \sum_{k=0}^{\infty} \frac{t^k}{\Gamma(\alpha k + \beta)},$$

while ψ is the Digamma function given by

$$\psi(x) := \frac{\Gamma'(x)}{\Gamma(x)}.$$

Functions ${}_1F_1$ and ${}_2F_1$ denote the usual hypergeometric functions, i.e.,

$${}_1F_1(a; b; z) := \frac{\Gamma(b)}{\Gamma(a)} \sum_{k=0}^{\infty} \frac{\Gamma(a+k)}{\Gamma(b+k)k!} z^k, \quad a \in \mathbb{R}, \quad -b \notin \mathbb{N}_0,$$

(sometimes called the Kummer, confluent, hypergeometric function [1, Chp. 13]), the power series being convergent for arbitrary complex z, and

$${}_2F_1(a, b; c; z) := \frac{\Gamma(c)}{\Gamma(a)\Gamma(b)} \sum_{k=0}^{\infty} \frac{\Gamma(a+k)\Gamma(b+k)}{\Gamma(c+k)k!} z^k, \quad a, b \in \mathbb{R}, \quad -c \notin \mathbb{N}_0,$$

(the Gauss hypergeometric function [1, Chp. 15]), in which case the power series converges for all complex z with $|z| < 1$, and may be extended analytically into the entire complex plane with a branch cut along the positive real axis from $+1$ to $+\infty$. (Note that in the formulæ below, we only need to evaluate this function for negative values of z, so the branch cut for positive z does not need to be addressed here.) Finally, $\mathrm{i} := \sqrt{-1}$ is the imaginary unit.

[7] Tables of Riemann-Liouville integrals and derivatives can be found in various places in the literature (cf. e.g., Podlubny [45] or Samko et al. [51]). We do not repeat those results here.

(1) Let $f(x) = x^j$. Here we have to distinguish some cases:

$$(D_\star^\alpha f)(x) = \begin{cases} 0 & \text{if } j \in \mathbb{N}_0 \text{ and } j < \lceil\alpha\rceil, \\ \dfrac{\Gamma(j+1)}{\Gamma(j+1-\alpha)}\, x^{j-\alpha} & \text{if } j \in \mathbb{N}_0 \text{ and } j \geq \lceil\alpha\rceil \text{ or } j \notin \mathbb{N} \text{ and } j > \lfloor\alpha\rfloor. \end{cases}$$

(2) Let $f(x) = (x+c)^j$ for arbitrary $c > 0$ and $j \in \mathbb{R}$. Then

$$\begin{aligned}(D_\star^\alpha f)(x) = {} & \frac{\Gamma(j+1)}{\Gamma(j-\lfloor\alpha\rfloor)} \frac{c^{j-\lceil\alpha\rceil-1}}{\Gamma(\lceil\alpha\rceil-\alpha+1)}\, x^{\lceil\alpha\rceil-\alpha} \\ & \times {}_2F_1(1, \lceil\alpha\rceil - j; \lceil\alpha\rceil - \alpha + 1; -x/c).\end{aligned}$$

(3) Let $f(x) = x^j \ln x$ for some $j > \lfloor\alpha\rfloor$. Then

$$\begin{aligned}(D_\star^\alpha f)(x) = {} & x^{j-\alpha} \sum_{k=0}^{\lfloor\alpha\rfloor} (-1)^{\lceil\alpha\rceil-k+1} \binom{j}{k} \frac{\lceil\alpha\rceil!}{\lceil\alpha\rceil - k} \frac{\Gamma(j-\lfloor\alpha\rfloor)}{\Gamma(j-\alpha+1)} \\ & + \frac{\Gamma(j+1)}{\Gamma(j-\alpha+1)}\, x^{j-\alpha}\Big(\psi(j-\lfloor\alpha\rfloor) - \psi(j-\alpha+1) + \ln x\Big).\end{aligned}$$

(4) Let $f(x) = \exp(jx)$ for some $j \in \mathbb{R}$. Then

$$(D_\star^\alpha f)(x) = j^{\lceil\alpha\rceil} x^{\lceil\alpha\rceil-\alpha} E_{1,\lceil\alpha\rceil-\alpha+1}(jx).$$

(5) Let $f(x) = \sin jx$ for some $j \in \mathbb{R}$. Here again we have two cases:

$$(D_\star^\alpha f)(x) = \begin{cases} -\dfrac{\mathrm{i} j^{\lceil\alpha\rceil} (-1)^{\lceil\alpha\rceil/2} x^{\lceil\alpha\rceil-\alpha}}{2\Gamma(\lceil\alpha\rceil-\alpha+1)} \big[{}_1F_1(1; \lceil\alpha\rceil-\alpha+1; \mathrm{i}jx) \\ \qquad - {}_1F_1(1; \lceil\alpha\rceil-\alpha+1; -\mathrm{i}jx)\big] \text{ if } \lceil\alpha\rceil \text{ is even}, \\ \dfrac{j^{\lceil\alpha\rceil} (-1)^{\lfloor\alpha\rfloor/2} x^{\lceil\alpha\rceil-\alpha}}{2\Gamma(\lceil\alpha\rceil-\alpha+1)} \big[{}_1F_1(1; \lceil\alpha\rceil-\alpha+1; \mathrm{i}jx) \\ \qquad + {}_1F_1(1; \lceil\alpha\rceil-\alpha+1; -\mathrm{i}jx)\big] \text{ if } \lceil\alpha\rceil \text{ is odd}. \end{cases}$$

(6) Finally we consider $f(x) = \cos jx$ with some $j \in \mathbb{R}$. As in the previous example, we obtain two cases:

$$(D_\star^\alpha f)(x) = \begin{cases} \dfrac{j^{\lceil\alpha\rceil} (-1)^{\lceil\alpha\rceil/2} x^{\lceil\alpha\rceil-\alpha}}{2\Gamma(\lceil\alpha\rceil-\alpha+1)} \big[{}_1F_1(1; \lceil\alpha\rceil-\alpha+1; \mathrm{i}jx) \\ \qquad + {}_1F_1(1; \lceil\alpha\rceil-\alpha+1; -\mathrm{i}jx)\big] \text{ if } \lceil\alpha\rceil \text{ is even}, \\ \dfrac{\mathrm{i} j^{\lceil\alpha\rceil} (-1)^{\lfloor\alpha\rfloor/2} x^{\lceil\alpha\rceil-\alpha}}{2\Gamma(\lceil\alpha\rceil-\alpha+1)} \big[{}_1F_1(1; \lceil\alpha\rceil-\alpha+1; \mathrm{i}jx) \\ \qquad - {}_1F_1(1; \lceil\alpha\rceil-\alpha+1; -\mathrm{i}jx)\big] \text{ if } \lceil\alpha\rceil \text{ is odd}. \end{cases}$$

B Automatic Integration

In Alg. 4 we have need to evaluate two integrals in order to be able to compute the Mittag-Leffler function. In this appendix we outline a possible strategy for the solution of these integrals. Essentially, we follow the ideas explained in QUADPACK [43]. The routines introduced in that book are the generally accepted standard when looking for efficient and reliable quadrature algorithms. They are in the public domain, but they can also be found in many commercially distributed software packages. The source code is written in FORTRAN77 and may be obtained, for example, from the URL `http://www.netlib.org/quadpack`. For integrands of the form encountered in Alg. 4, it turns out that the routine DQAG (Double precision Quadrature, Adaptive, General purpose) is the method of choice. Our description of this routine follows a top-down structure (i.e., we first explain the general strategy without giving much information on the details, and at a later stage we fill in those details).

B.1 Fundamental strategy

The fundamental idea of automatic integration is the following. The user supplies the integrand function, the bounds for the interval of integration (i.e., the data that define the integral in question), and a desired accuracy (i.e., a bound for the relative or absolute error that he/she is willing to accept). The routine is then supposed to return either an approximation for the correct value of the integral that is sufficiently accurate to satify the user's requirements, or an error flag if it fails to find such an approximation.

Typically, the algorithms try to achieve this goal by sub-dividing the interval of integration in an adaptive way, thus concentrating more quadrature nodes in areas where the integrand is difficult to approximate. This leads to a structure indicated in Alg. 6.

In practice, one may also terminate the **while** loop in this algorithm when, for example, too many sub-divisions have taken place thereby using up all available memory, or too much computing time has been consumed. In such cases, the algorithm should return an error flag.

B.2 Approximation of the integral

In Alg. 6 we have left open some of the key details. Specifically, we have not said how the approximation of the integral itself will be performed, and we

have not indicated how the required error estimates can be found. We now turn our attention to these two questions.

Algorithm 6
Automatic Integration:
given $a, b \in \mathbb{R}$, $f : [a, b] \to \mathbb{C}$, $\epsilon > 0$ **then**
Calculate an approximation $Q[f]$ *for* $\int_a^b f(x)\,\mathrm{d}x$
and an estimate $\hat{\epsilon}$ *for the error* $\int_a^b f(x)\,\mathrm{d}x - Q[f]$
Initialize a list L *of the approximations obtained so far*
as $L := \{([a, b], Q[f], \hat{\epsilon})\}$
while $\sum_L |\hat{\epsilon}| > \epsilon$ **do**
Take the interval from L *with the largest error estimate*
and remove it from the list;
Bisect this interval;
Calculate approximations and error estimates for the two
newly obtained subintervals;
Add these subintervals, their corresponding approximations,
and their error estimates to the list L
end
return *the sum of approximations over all the subintervals*

The basic idea behind the solution is the concept of Gauss-Kronrod integration. That is, we calculate two different approximations for the integral, both of which are of the form

$$\sum_{j=1}^{n} a_{j,n} f(x_{j,n}),$$

with suitably chosen values of n, $a_{j,n}$, and $x_{j,n}$ $(j = 1, \ldots, n)$. In particular, we begin with a first approximation that is just a Gauss quadrature formula with n_1 nodes, which is the (uniquely determined) quadrature formula that gives the exact value of the integral whenever the integrand is a polynomial of degree not exceeding $2n_1 - 1$. This formula has been thoroughly investigated. For a recent survey we refer to Ref. [3] and the references cited therein. Specifically, there is no quadrature formula with the same number of nodes that is exact for a larger class of polynomials. Moreover, as stated in [3], both theoretical and practical evidence suggest that this method is a very good one.

B.3 Approximation of error estimates

From rather general considerations, it is known that one, single, quadrature formula can only give an approximation, but not both an approximation and an error estimate. Therefore, to derive also an error estimate, it is necessary to

introduce a second quadrature formula. The heuristic argument is as follows. Assuming that the second quadrature formula is much more accurate than the first, then the difference between the two approximations will be a good approximation for the error of the cruder of the two approximations, and therefore, a rather conservative (and thus quite reliable) upper bound for the error of the finer of the two approximations. Hence, we need a second quadrature formula that is significantly better than the first (Gauss-type) formula. In view of the quality of the Gauss formula, this can only be achieved by selecting a formula with n_2 nodes, where $n_2 > n_1$. In principle, one could use a Gauss formula again, but this would be very uneconomical because a Gauss formula with n_1 nodes would have at most one node in common with a Gauss formula with n_2 nodes, and so almost none of the information gathered so far (i.e., the function values $f(x_{j,n_1})$, whose calculation is typically the most computationally expensive part of the algorithm) could be reused. To overcome this difficulty, Kronrod [28,29] suggested to construct a formula that is nowadays called the Kronrod extension of the Gauss formula or, shortly, the Gauss-Kronrod formula. His formula is based on the Gauss formula with n_1 nodes; it uses $n_2 = 2n_1+1$ nodes, and is constructed according to the following criteria:

- The n_1 nodes x_{j,n_1} of the Gauss method form a subset of the n_2 nodes x_{j,n_2} of the Gauss-Kronrod method.
- The remaining nodes and the weights a_{j,n_2} of the of the Gauss-Kronrod scheme are determined in such a way that the resulting method is exact for all polynomials of degree not exceeding $3n_1 + 1$.

A recent survey on Gauss-Kronrod formulas is given in Ref. [13].

Algorithms for the concrete calculation of the required nodes and weights are available (cf. the survey papers mentioned above and the references cited therein). For our purposes, it is sufficient to use the tabulated values given in Ref. [43]. In particular, following the suggestions made there, we propose using the Gaussian method with 15 points in combination with the 31-point Kronrod extension for computing the integral with function K mentioned in Alg. 4 (the monotonic part). In view of the nice smoothness properties of the integrand, this gives us an approximation with quite high accuracy without too much computational effort. For the other integral in that algorithm, with the oscillatory integrand function P, these formulæ may be unable to follow the oscillations properly; thus, we propose using the 30-point Gauss method together with its 61-point Kronrod extension. This is essentially also the method suggested for oscillatory integrals in the NAG Library [40] (see the documentation for routine D01AKF).

C Table of Padé Approximates for Mittag-Leffler Function

This appendix presents a scheme for fast computations of

$$E_{\alpha,1}(-x^\alpha), \quad 0 < \alpha < 1, \quad x \in \mathbb{R}_+,$$

suitable for use in finite element analyses. As mentioned in §8, this function is of prime importance in many applications related to FDE's.

The defining series, Eq. (21), is used on the interval $0 < x < 0.1$, and its asymptotic series, Eq. (32), is used whenever $x > 15$, as in Alg. 4, but in contrast with Alg. 4, a Padé approximate (or rational polynomial) is applied inbetween where $0.1 \leq x \leq 15$; specifically,

$$E_{\alpha,1}(-x^\alpha) \approx \begin{cases} \displaystyle\sum_{k=0}^{4} \frac{(-x)^{\alpha k}}{\Gamma(1+\alpha k)}, & 0 \leq x \leq 0.1, \\ \dfrac{a_0 + a_1 x + a_2 x^2}{1 + b_1 x + b_2 x^2 + b_3 x^3}, & 0.1 < x < 15, \\ \displaystyle -\sum_{k=1}^{4} \frac{(-x)^{-\alpha k}}{\Gamma(1-\alpha k)}, & x \geq 15, \end{cases} \tag{C.1}$$

where the coefficients a_0, a_1, a_2, b_1, b_2, and b_3 are listed in Table C.1 for values of α varying between 0.01 and 0.99 by increments of 0.01. Also listed are the r^2 resultants from each nonlinear regression with the tolerance set sufficient to achieve accuracies of $r^2 > 0.999$, except for $\alpha > 0.9$ where such tight r^2 values could not be achieved using the order of approximate employed. The Padé approximates were obtained from fits to exact values (within machine precision) for the Mittag-Leffler function (obtained from Alg. 4) at 150 evenly spaced grid points over the interval $[0.1, 15]$. The form of this rational polynomial was selected after a study of numerous tables for function approximation listed in the appendices of Hart et al. [25]. Our fits were constrained to be exact at the end points so that transitions will be smooth when crossing a boundary between solution domains.

A number written as 0.123456(7) stands for $0.123456 \cdot 10^7 = 1,234,560$.

Table C.1: Coefficients for Padé approximates for $E_{\alpha,1}(-x^\alpha)$, $x \in [0.1, 15]$.

α	a_0	a_1	a_2	b_1	b_2	b_3	r^2
0.01	-.424129(3)	-.695856(6)	-.391944(6)	-.138637(7)	-.795778(6)	-.207400(3)	0.999371
0.02	-.291078(4)	-.231705(7)	-.121393(7)	-.460100(7)	-.250385(7)	-.127923(4)	0.999345
0.03	-.159594(4)	-.883786(6)	-.485177(6)	-.174711(7)	-.101562(7)	-.832687(3)	0.999381
0.04	-.401668(4)	-.151329(7)	-.678102(6)	-.298958(7)	-.144743(7)	-.138260(4)	0.999114
0.05	-.480619(4)	-.145550(7)	-.643881(6)	-.286638(7)	-.139751(7)	-.171118(4)	0.999129
0.06	-.358700(4)	-.953746(6)	-.452754(6)	-.186815(7)	-.996694(6)	-.160873(4)	0.999313
0.07	-.431271(4)	-.958824(6)	-.424475(6)	-.187447(7)	-.952144(6)	-.175012(4)	0.999215
0.08	-.104813(5)	-.220562(7)	-.109649(7)	-.427837(7)	-.248747(7)	-.594850(4)	0.999401
0.09	-.113182(5)	-.204162(7)	-.931135(6)	-.395568(7)	-.215524(7)	-.560193(4)	0.999340
0.10	-.100534(5)	-.165546(7)	-.753470(6)	-.319371(7)	-.177305(7)	-.527749(4)	0.999361
0.11	-.771890(4)	-.116952(7)	-.530686(6)	-.224658(7)	-.126954(7)	-.429168(4)	0.999382
0.12	-.209949(4)	-.331485(6)	-.178194(6)	-.628014(6)	-.428531(6)	-.183967(4)	0.999349
0.13	-.201966(5)	-.264628(7)	-.118361(7)	-.503944(7)	-.292906(7)	-.123451(5)	0.999408
0.14	-.144185(5)	-.177380(7)	-.787251(6)	-.336281(7)	-.198184(7)	-.923408(4)	0.999420
0.15	-.358699(4)	-.414163(6)	-.180624(6)	-.782040(6)	-.463020(6)	-.235432(4)	0.999423
0.16	-.367384(5)	-.387982(7)	-.157708(7)	-.732069(7)	-.414017(7)	-.218683(5)	0.999358
0.17	-.834348(4)	-.877610(6)	-.381391(6)	-.163991(7)	-.101120(7)	-.619007(4)	0.999453
0.18	-.665920(4)	-.661019(6)	-.279508(6)	-.123090(7)	-.755801(6)	-.495376(4)	0.999452
0.19	-.880313(4)	-.839225(6)	-.352670(6)	-.155465(7)	-.970538(6)	-.689880(4)	0.999463
0.20	-.516937(4)	-.463982(6)	-.186660(6)	-.857478(6)	-.525412(6)	-.391939(4)	0.999446
0.21	-.181654(5)	-.159929(7)	-.655973(6)	-.293380(7)	-.187316(7)	-.153968(5)	0.999480
0.22	-.287739(5)	-.241617(7)	-.962412(6)	-.441685(7)	-.280701(7)	-.244057(5)	0.999477
0.23	-.283969(5)	-.233523(7)	-.935683(6)	-.423970(7)	-.277398(7)	-.261404(5)	0.999497
0.24	-.218950(5)	-.174274(7)	-.687682(6)	-.314803(7)	-.207911(7)	-.208550(5)	0.999503
0.25	-.184865(5)	-.142847(7)	-.555414(6)	-.256670(7)	-.171282(7)	-.182554(5)	0.999510
0.26	-.154930(5)	-.116572(7)	-.448241(6)	-.208289(7)	-.140906(7)	-.159998(5)	0.999520
0.27	-.252656(5)	-.187367(7)	-.720787(6)	-.332362(7)	-.230555(7)	-.280571(5)	0.999534
0.28	-.348286(5)	-.250157(7)	-.938597(6)	-.441764(7)	-.306951(7)	-.392527(5)	0.999538
0.29	-.223178(5)	-.158819(7)	-.597404(6)	-.278265(7)	-.198723(7)	-.272201(5)	0.999553
0.30	-.262152(5)	-.180630(7)	-.659350(6)	-.315184(7)	-.224602(7)	-.321389(5)	0.999556
0.31	-.110316(5)	-.741001(6)	-.263517(6)	-.128658(7)	-.919175(6)	-.137449(5)	0.999558
0.32	-.344887(5)	-.231540(7)	-.831618(6)	-.398374(7)	-.294522(7)	-.473795(5)	0.999578
0.33	-.253440(5)	-.165943(7)	-.579558(6)	-.284133(7)	-.210336(7)	-.352888(5)	0.999583
0.34	-.320037(5)	-.206983(7)	-.712936(6)	-.351975(7)	-.264342(7)	-.467058(5)	0.999592
0.35	-.788648(5)	-.506620(7)	-.172863(7)	-.854865(7)	-.654223(7)	-.122026(6)	0.999604
0.36	-.289781(5)	-.195379(7)	-.707274(6)	-.324061(7)	-.268137(7)	-.555978(5)	0.999616

Table C.1: Coefficients for Padé approximates for $E_{\alpha,1}(-x^\alpha)$, $x \in [0.1, 15]$.

α	a_0	a_1	a_2	b_1	b_2	b_3	r^2
0.37	-.142051(5)	-.971770(6)	-.354821(6)	-.159385(7)	-.136569(7)	-.302386(5)	0.999617
0.38	-.816551(5)	-.523578(7)	-.175872(7)	-.860767(7)	-.705295(7)	-.156056(6)	0.999642
0.39	-.575871(5)	-.348533(7)	-.107617(7)	-.574012(7)	-.450994(7)	-.991720(5)	0.999625
0.40	-.173808(5)	-.117277(7)	-.408516(6)	-.187857(7)	-.167833(7)	-.429558(5)	0.999651
0.41	-.103477(5)	-.686777(6)	-.231234(6)	-.109392(7)	-.977034(6)	-.258727(5)	0.999700
0.42	-.195086(5)	-.129617(7)	-.429282(6)	-.204698(7)	-.185582(7)	-.514821(5)	0.999681
0.43	-.333165(5)	-.221252(7)	-.718492(6)	-.346552(7)	-.318194(7)	-.922518(5)	0.999693
0.44	-.473286(5)	-.311742(7)	-.981295(6)	-.484992(7)	-.447200(7)	-.134275(6)	0.999708
0.45	-.147598(5)	-.982326(6)	-.304968(6)	-.151324(7)	-.142189(7)	-.447789(5)	0.999719
0.46	-.152914(5)	-.104712(7)	-.326140(6)	-.159246(7)	-.154553(7)	-.517302(5)	0.999725
0.47	-.167037(5)	-.114533(7)	-.346912(6)	-.172774(7)	-.169118(7)	-.587605(5)	0.999740
0.48	-.142611(5)	-.991368(6)	-.293586(6)	-.148061(7)	-.147062(7)	-.531760(5)	0.999752
0.49	-.654706(5)	-.455618(7)	-.130849(7)	-.675094(7)	-.675481(7)	-.253173(6)	0.999766
0.50	-.188800(3)	-.135370(5)	-.383675(4)	-.198223(5)	-.202768(5)	-.797517(3)	0.999777
0.51	-.516448(5)	-.373170(7)	-.102716(7)	-.541357(7)	-.559798(7)	-.228252(6)	0.999790
0.52	-.392154(5)	-.289687(7)	-.780114(6)	-.415703(7)	-.437101(7)	-.185908(6)	0.999801
0.53	-.135449(5)	-.106392(7)	-.289899(6)	-.150142(7)	-.164334(7)	-.749106(5)	0.999802
0.54	-.953841(4)	-.748349(6)	-.193214(6)	-.104887(7)	-.114460(7)	-.530462(5)	0.999822
0.55	-.793386(3)	-.639333(5)	-.159875(5)	-.886439(5)	-.979799(5)	-.469973(4)	0.999835
0.56	-.599232(4)	-.443865(6)	-.952614(5)	-.619108(6)	-.646389(6)	-.288690(5)	0.999847
0.57	-.337393(5)	-.286190(7)	-.663678(6)	-.388535(7)	-.438767(7)	-.223581(6)	0.999859
0.58	-.463960(5)	-.405168(7)	-.899289(6)	-.544254(7)	-.620515(7)	-.324201(6)	0.999871
0.59	-.151791(4)	-.125852(6)	-.241507(5)	-.169238(6)	-.185486(6)	-.904156(4)	0.999886
0.60	-.502729(4)	-.464569(6)	-.919187(5)	-.611566(6)	-.705142(6)	-.377940(5)	0.999896
0.61	-.480103(5)	-.461103(7)	-.860143(6)	-.600297(7)	-.697998(7)	-.378472(6)	0.999907
0.62	-.671512(4)	-.708363(6)	-.131044(6)	-.906252(6)	-.108843(7)	-.626832(5)	0.999908
0.63	-.312780(5)	-.328986(7)	-.533551(6)	-.418611(7)	-.494518(7)	-.268704(6)	0.999928
0.64	-.567954(5)	-.632239(7)	-.945394(6)	-.794766(7)	-.947107(7)	-.509518(6)	0.999937
0.65	-.149230(4)	-.182752(6)	-.259499(5)	-.226154(6)	-.275373(6)	-.151317(5)	0.999939
0.66	-.433396(5)	-.549988(7)	-.664464(6)	-.674519(7)	-.815645(7)	-.407908(6)	0.999953
0.67	-.752321(4)	-.109682(7)	-.126030(6)	-.132094(7)	-.164296(7)	-.843703(5)	0.999948
0.68	-.503149(5)	-.782828(7)	-.716973(6)	-.932883(7)	-.115421(8)	-.500161(6)	0.999960
0.69	-.113475(5)	-.204800(7)	-.158167(6)	-.240096(7)	-.301897(7)	-.117538(6)	0.999955
0.70	-.277604(3)	-.535397(5)	-.226639(4)	-.621910(5)	-.768192(5)	-.150446(4)	0.999976
0.71	-.551694(5)	-.158192(8)	-.673086(6)	-.178833(8)	-.232888(8)	-.532630(6)	0.999919
0.72	-.310305(5)	-.102892(8)	0.663348(5)	-.115056(8)	-.146317(8)	0.202771(6)	0.99994

Table C.1: Coefficients for Padé approximates for $E_{\alpha,1}(-x^{\alpha})$, $x \in [0.1, 15]$.

α	a_0	a_1	a_2	b_1	b_2	b_3	r^2
0.73	-.160766(4)	-.449254(7)	-.116371(6)	-.482429(7)	-.682061(7)	-.111373(6)	0.999719
0.74	0.341935(4)	-.241906(7)	-.669717(5)	-.252295(7)	-.376790(7)	-.808538(5)	0.999510
0.75	0.173763(4)	-.103357(7)	-.513330(6)	-.106853(7)	-.212490(7)	-.787594(6)	0.999166
0.76	-.198004(4)	-.170209(6)	-.546410(6)	-.212224(6)	-.727938(6)	-.906762(6)	0.999483
0.77	-.255574(4)	-.399973(6)	-.408010(6)	-.461840(6)	-.889107(6)	-.736776(6)	0.999574
0.78	-.107310(4)	-.102865(6)	-.104591(6)	-.125079(6)	-.207143(6)	-.205284(6)	0.999673
0.79	-.494450(4)	-.463767(6)	-.357502(6)	-.563528(6)	-.802633(6)	-.767703(6)	0.999707
0.80	-.218066(4)	-.219047(6)	-.128115(6)	-.262855(6)	-.339154(6)	-.302753(6)	0.999721
0.81	-.534378(4)	-.539996(6)	-.258539(6)	-.645497(6)	-.766988(6)	-.675138(6)	0.999723
0.82	-.675467(4)	-.688012(6)	-.275764(6)	-.819312(6)	-.908437(6)	-.800006(6)	0.999715
0.83	-.702848(3)	-.690578(5)	-.240789(5)	-.824283(5)	-.847142(5)	-.779293(5)	0.999699
0.84	-.609980(4)	-.583751(6)	-.176849(6)	-.697471(6)	-.667838(6)	-.642815(6)	0.999674
0.85	-.103053(5)	-.915547(6)	-.247199(6)	-.110278(7)	-.963517(6)	-.101397(7)	0.999639
0.86	-.774771(4)	-.667011(6)	-.156824(6)	-.805409(6)	-.651177(6)	-.733497(6)	0.999594
0.87	-.838904(4)	-.693431(6)	-.141730(6)	-.840712(6)	-.624128(6)	-.763282(6)	0.999538
0.88	-.447750(4)	-.359964(6)	-.629983(5)	-.437447(6)	-.299094(6)	-.396021(6)	0.999468
0.89	-.390787(4)	-.296304(6)	-.446099(5)	-.362820(6)	-.222076(6)	-.331368(6)	0.999382
0.90	-.724688(4)	-.509096(6)	-.654940(5)	-.630388(6)	-.335377(6)	-.584487(6)	0.999277
0.91	-.152262(5)	-.100158(7)	-.106340(6)	-.125254(7)	-.574996(6)	-.117309(7)	0.999150
0.92	-.150108(5)	-.942028(6)	-.792810(5)	-.118669(7)	-.467003(6)	-.112366(7)	0.998998
0.93	-.106495(5)	-.626399(6)	-.398859(5)	-.797901(6)	-.255226(6)	-766267(6)	0.998816
0.94	-.153906(5)	-.847297(6)	-371308(5)	-.109242(7)	-.268144(6)	-.106481(7)	0.998600
0.95	-.600332(4)	-.311566(6)	0.756057(4)	-.406277(6)	-.710353(5)	-.401715(6)	0.998343
0.96	-.445797(5)	-.214492(7)	-.119855(5)	-.284119(7)	-.270455(6)	-.285641(7)	0.998041
0.97	-.355171(5)	-.159674(7)	0.206708(5)	-.214673(7)	-.423690(5)	-.219239(7)	0.997685
0.98	-.145397(5)	-.610793(6)	0.190402(5)	-.834188(6)	0.464083(5)	-.865369(6)	0.997270
0.99	-.105996(5)	-.416113(6)	0.204340(5)	-.577797(6)	0.755603(5)	-.608788(6)	0.996786

References

[1] M. Abramowitz and I. A. Stegun (eds.), *Handbook of mathematical functions: With formulas, graphs, and mathematical tables*, NBS Applied Mathematics Series, vol. 55, National Bureau of Standards, Washington, D.C., 1964, Second printing with corrections. Republished by Dover, New York, 1965.

[2] R. L. Bagley and P. J. Torvik, *A theoretical basis for the application of fractional calculus to viscoelasticity*, Journal of Rheology **27** (1983), 201–210.

[3] H. Brass, J.-W. Fischer, and K. Petras, *The Gaussian quadrature method*, Abhandlungen der Braunschweigischen Wissenschaftlichen Gesellschaft **47** (1996), 115–150.

[4] M. Caputo, *Linear models of dissipation whose Q is almost frequency independent–II*, The Geophysical Journal of the Royal Astronomical Society **13** (1967), 529–539.

[5] M. Caputo and F. Mainardi, *Linear models of dissipation in anelastic solids*, Rivista del Nuovo Cimento **1** (1971), 161–198.

[6] J.-T. Chern, *Finite element modeling of viscoelastic materials on the theory of fractional calculus*, Ph.D. thesis, The Pennsylvania State University, December 1993, University Microfilms No. 9414260.

[7] K. Diethelm, *Generalized compound quadrature formulæ for finite-part integrals*, IMA Journal of Numerical Analysis **17** (1997), 479–493.

[8] K. Diethelm and N. J. Ford, *Analysis of fractional differential equations*, Journal of Mathematical Analysis and Applications **265** (2002), 229–248.

[9] K. Diethelm, N. J. Ford, and A. D. Freed, *Detailed error analysis for a fractional Adams method*, In review.

[10] ———, *A predictor-corrector approach for the numerical solution of fractional differential equations*, Nonlinear Dynamics **29** (2002), 3–22.

[11] K. Diethelm and A. D. Freed, *On the solution of nonlinear fractional differential equations used in the modeling of viscoplasticity*, Scientific Computing in Chemical Engineering II: Computational Fluid Dynamics, Reaction Engineering, and Molecular Properties (F. Keil, W. Mackens, H. Voß and J. Werther, eds.), Springer, Heidelberg, 1999, pp. 217–224.

[12] K. Diethelm and G. Walz, *Numerical solution of fractional order differential equations by extrapolation*, Numerical Algorithms **16** (1997), 231–253.

[13] S. Ehrich, *Stieltjes polynomials and the error of Gauss-Kronrod quadrature formulas*, Applications and Computation of Orthogonal Polynomials (W. Gautschi, G. Golub, and G. Opfer, eds.), International Series of Numerical Mathematics, no. 131, Birkhäuser, Basel, 1999, pp. 57–77.

[14] D. Elliott, *An asymptotic analysis of two algorithms for certain Hadamard finite-part integrals*, IMA Journal of Numerical Analysis **13** (1993), 445–462.

[15] A. Erdélyi, W. Magnus, F. Oberhettinger, and F. G. Tricomi (eds.), *Higher transcendental functions*, Bateman manuscript project, vol. 2, McGraw-Hill, New York, 1955.

[16] N. J. Ford and A. C. Simpson, *The numerical solution of fractional differential equations: Speed versus accuracy*, Numerical Algorithms **26** (2001), 333–346.

[17] A. D. Freed, K. Diethelm, T. C. Doehring, M. A. Lillie, E. O. Carew, and I. Vesely, *Modeling passive soft tissues*, In preparation, 2003.

[18] A. D. Freed, K. Diethelm, and Y. Luchko, *Fractional-order viscoelasticity (FOV): Constitutive development using the fractional calculus: First annual report*, NASA/TM 2002-211914, NASA's Glenn Research Center, Brook Park, Ohio, December 2002.

[19] L. Gaul, P. Klein, and S. Kempfle, *Damping description involving fractional operators*, Mechanical Systems and Signal Processing **5** (1991), 81–88.

[20] W. G. Glöckle and T. F. Nonnenmacher, *A fractional calculus approach to self-similar protein dynamics*, Biophysical Journal **68** (1995), 46–53.

[21] R. Gorenflo, *Fractional calculus: Some numerical methods*, Fractals and Fractional Calculus in Continuum Mechanics (A. Carpinteri and F. Mainardi, eds.), CISM Courses and Lectures, no. 378, Springer, Wien, 1997, pp. 277–290.

[22] R. Gorenflo, I. Loutchko, and Yu. Luchko, *Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z)$ and its derivatives*, Fractional Calculus and Applied Analysis **5** (2002), 491–518.

[23] R. Gorenflo and F. Mainardi, *Fractional calculus: Integral and differential equations of fractional order*, Fractals and Fractional Calculus in Continuum Mechanics (A. Carpinteri and F. Mainardi, eds.), CISM Courses and Lectures, no. 378, Springer, Wien, 1997, pp. 223–276.

[24] R. Gorenflo and R. Rutman, *On ultraslow and intermediate processes*, Transform Methods and Special Functions, Sofia 1994 (P. Rusev, I. Dimovski, and V. Kiryakova, eds.), Science Culture Technology, Singapore, 1995, pp. 61–81.

[25] J. F. Hart, E. W. Cheney, C. L. Lawson, H. J. Maehly, C. K. Mesztenyi, J. R. Rice, H. G. Thacher, Jr., and C. Witzgall, *Computer approximations*, The SIAM Series in Applied Mathematics, John Wiley & Sons, New York, 1968.

[26] F. de Hoog and R. Weiss, *Asymptotic expansions for product integration*, Mathematics of Computation **27** (1973), 295–306.

[27] A. A. Kilbas and J. J. Trujillo, *Differential equations of fractional order: Methods, results and problems–I*, Applicable Analysis **78** (2001), 153–192.

[28] A. S. Kronrod, *Integration with control of accuracy*, Soviet Physics Doklady **9** (1964), no. 1, 17–19.

[29] ———, *Nodes and weights for quadrature formulæ: Sixteen place tables*, Nauka, Moscow, 1964, In Russian (English translation: Consultants Bureau, New York, 1965).

[30] J. Liouville, *Mémoire sur quelques questions de géométrie et de mécanique, et sur un nouveau genre de calcul pour résoudre ces questions*, Journal de l'École Polytechnique **13** (1832), no. cahier 21, 1–66.

[31] J. Lützen, 2001, Personal communication.

[32] F. Mainardi, *Fractional calculus: Some basic problems in continuum and statistical mechanics*, Fractals and Fractional Calculus in Continuum Mechanics (A. Carpinteri and F. Mainardi, eds.), CISM Courses and Lectures, no. 378, Springer, Wien, 1997, pp. 291–348.

[33] F. Mainardi and R. Gorenflo, *The Mittag-Leffler function in the Riemann-Liouville fractional calculus*, Boundary Value Problems, Special Functions and Fractional Calculus. Proceedings of international conference dedicated to 90th birthday of Prof. F. D. Gakhov (Minsk) (A. A. Kilbas, ed.), Belarusian State University, February 1996, pp. 215–225.

[34] R. J. Marks, II and M. W. Hall, *Differintegral interpolation from a bandlimited signal's samples*, IEEE Transactions on Acoustics and Speech and Signal Processing **29** (1981), 872–877.

[35] D. Matignon and G. Montseny (eds.), *Fractional differential systems: Models, methods, and applications*, ESAIM Proceedings, vol. 5, SMAI, Paris, 1998.

[36] R. Metzler, W. Schick, H.-G. Kilian, and T. F. Nonnenmacher, *Relaxation in filled polymers: A fractional calculus approach*, Journal of Chemical Physics **103** (1995), 7180–7186.

[37] K. S. Miller and B. Ross, *An introduction to the fractional calculus and fractional differential equations*, John Wiley & Sons, New York, 1993.

[38] G. Mittag-Leffler, *Sur la représentation analytique d'une branche uniforme d'une fonction monogène*, Acta Mathematica **29** (1904), 101–168.

[39] T. F. Nonnenmacher and R. Metzler, *On the Riemann-Liouville fractional calculus and some recent applications*, Fractals **3** (1995), 557–566.

[40] Numerical Algorithms Group, Ltd., Oxford, *NAG FORTRAN Library Mark 17*, 1995.

[41] K. B. Oldham and J. Spanier, *The fractional calculus*, Mathematics in Science and Engineering, vol. 111, Academic Press, New York, 1974.

[42] W. E. Olmstead and R. A. Handelsman, *Diffusion in a semi-infinite region with nonlinear surface dissipation*, SIAM Review **18** (1976), 275–291.

[43] R. Piessens, E. de Doncker-Kapenga, C. Überhuber, and D. K. Kahaner, *QUADPACK—A subroutine package for automatic integration*, Springer Series in Computational Mathematics, no. 1, Springer, Berlin, 1983.

[44] I. Podlubny, *Fractional-order systems and fractional-order controllers*, Tech. Report UEF-03-94, Institute for Experimental Physics, Slovak Academy of Sciences, 1994.

[45] ———, *Fractional differential equations: An introduction to fractional derivatives, fractional differential equations, to methods of their solution and some of their applications*, Mathematics in Science and Engineering, vol. 198, Academic Press, San Diego, 1999.

[46] ———, *Geometric and physical interpretation of fractional integration and fractional differentiation*, Fractional Calculus and Applied Analysis **5** (2002), 367–386.

[47] I. Podlubny, L. Dorcak, and J. Misanek, *Application of fractional-order derivatives to calculation of heat load intensity change in blast furnace walls*, Transactions of the Technical University of Košice **5** (1995), 137–144.

[48] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical recipes in C: The art of scientific computing*, 2nd ed., Cambridge University Press, Cambridge, 1999, Reprinted, corrected to software version 2.08.

[49] Yu. N. Rabotnov, *Creep problems in structural members*, North-Holland Series in Applied Mathematics and Mechanics, vol. 7, North-Holland Publishing Company, Amsterdam, 1969, Originally published in Russian as: *Polzuchest Elementov Konstruktsii*, Nauka, Moscow, 1966.

[50] B. Ross, *The development of fractional calculus 1695–1900*, Historia Mathematica **4** (1977), 75–89.

[51] S. G. Samko, A. Kilbas, and O. I. Marichev, *Fractional integrals and derivatives: Theory and applications*, Gordon and Breach, Yverdon, 1993.

[52] M. Unser and T. Blu, *Fractional splines and wavelets*, SIAM Review **42** (2000), 43–67.

[53] ———, *Wavelet theory demystified*, IEEE Transactions on Signal Processing **51** (2003), 470–483.

[54] S. W. J. Welch, R. A. L. Ropper, and R. G. Duren, Jr., *Application of time-based fractional calculus methods to viscoelastic creep and stress relaxation of materials*, Mechanics of Time-Dependent Materials **3** (1999), 279–303.