Dipartimento di Matematica e Fisica
Sezione di Matematica

Dottorato di Ricerca in Matematica
XXVI Ciclo

Numerical Treatment of Fractional Differential Equations

Candidate:
Moreno Concezzi

Advisor:
Prof. Renato Spigler

Coordinator of the Ph.D. Program:
Prof. Luigi Chierchia
ACADEMIC YEAR 2014/2015

Acknowledgements

A special thanks to my advisor Prof. Renato Spigler.

Contents

Introduction

Many authors refer to a specific date as the Birthday of the so called "Fractional Calculus". In a letter dated September 30th, 1695, Guillaume de L'Hopital wrote to Leibniz asking him about the special notation he had used in his papers to denote the nth-derivative of the linear function $f(x) = x$, that is $\frac{D^n x}{Dx^n}$. L'Hopital asked Leibniz what would the result be if $n = 1/2$. Leibniz's reply was: "An apparent paradox, from which one day useful consequences will be drawn." Fractional calculus was born in (or from) such words.

Fractional Calculus was primarily intended as a study reserved to the best mathematical minds. Fourier, Euler, and Laplace are among those who got involved in fractional calculus and its mathematical consequences [106]. Many people, using their own notation and methodology, found definitions that fit somehow the concept of a non-integer order integral or derivative. The most famous and popular ones are those due to Riemann and Liouville, and to Grünwald and Letnikov. While there exists a number of definitions, they are mostly variations on the themes of these two.

Most of the mathematical theory applicable to the fractional calculus was developed before the end of the 20th century. However, it is only in the past 20 years that the most intriguing advances in Engineering and applied sciences took place. In some cases, the mathematics had to be adapted to meet the requirements of the physical models. In 1967, in his work [19], Michele Caputo reformulated the by then *classic* definition of the Riemann-Liouville fractional derivative in a way that allowed using initial conditions of integer order, when solving certain fractional order differential equations [20–24, 50, 75, 112]. Caputo definition of fractional derivative is now one the most used and useful definitions. In 1996, Kolowankar reformulated once more the Riemann-Liouville fractional derivative in order to perform fractional derivatives of nowhere differentiable fractal functions [80].

Considering all works accomplished over the last 300 years, Leibniz's reply to L'Hopital has proven to be at least partially correct. Indeed, by the end of the 20th century, a number of applications and physical evidence of the usefulness of fractional calculus

have been found. However, both, such applications and the mathematical background underlying the fractional calculus are far from being paradoxical [114].

The recent years have witnessed a broad interest in fractional calculus, and in particular in fractional differential equations. This is justified by the fact that, for instance, fractional time-derivatives may account for (time) *delays* or memory and hereditary effects [91], while fractional space-derivatives may explain some *nonlocal* behavior, typically observed in a power law (rather than exponential law) decay of solutions. Therefore, transport and diffusion in certain media seem to be better modeled by *fractional* partial differential equations. In particular, anomalous diffusion (sub- or super-diffusion) have been described in this way by many authors [24, 31, 85]. Consequently, a number of works pertaining to hydrology and percolation through porous media, meteorology, and even financial modeling, have appeared in the recent literature [11, 12].

Modern microscopic techniques following the stochastic motion of labeled tracer particles have uncovered significant deviations from the laws of Brownian motion in a variety of animate and inanimate systems. This kind of anomalous diffusion has been studied in the literature adopting models which use fractional partial differential equations [103].

We give now some important definitions of fractional derivatives and their importance in applications.

It is customary to adopt the *Caputo* definition for time derivatives, while for the space derivatives is usual adopting the *Riemann-Liouville* (RL) definition. For a given function, $f(t)$, the Caputo fractional derivative or order α it is defined as

$$ {}_{C}D_t^{\alpha} f(t) := \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{f^{(n)}(\xi)}{(t-\xi)^{\alpha-n+1}}\, d\xi, \tag{1} $$

while for a given function $g(x)$, the Riemann-Liouville (RL) fractional derivative it is defined as

$$ {}_{RL}D_t^{\alpha} f(t) := \frac{1}{\Gamma(n-\alpha)} \frac{\mathrm{d}^n}{\mathrm{d}t^n} \int_0^t \frac{f(\xi)}{(t-\xi)^{\alpha-n+1}}\, \mathrm{d}\xi, \tag{2} $$

where $n \in \mathbf{N}$ is such that $n-1 < \alpha \leq n$, and Γ is the Euler Gamma function, see [112]

The Caputo derivative is more convenient when imposing initial values. On the other hand, RL derivatives have been found to be useful to characterize space derivatives ?? in anomalous diffusion, Levy flights, and traps [30, 58, 153].

The latter is defined as

$$ {}_{GL}D_x^{\alpha} \psi(x) := \frac{1}{\Gamma(-\alpha)} \lim_{h \to 0} \frac{1}{h^{\alpha}} \sum_{i=0}^{[\frac{x}{h}]} \frac{\Gamma(i-\alpha)}{\Gamma(i+1)}\, \psi(x-ih), \tag{3} $$

see [88, 112]. In several useful algorithms, is is also much used the so-called *shifted* Grünwald-Letnikov derivative,

$$ {}_{GLs}D_x^{\alpha}\psi(x) = \frac{1}{\Gamma(-\alpha)} \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{i=0}^{[\frac{x}{h}]+1} \frac{\Gamma(i-\alpha)}{\Gamma(i+1)} \psi(x-(i-1)h), \tag{4} $$

where $[x]$ denotes the integer part of x, h being the step size that will be used in the numerical schemes [112], and Γ is the Gamma function. Clearly, the GL definitions are useful in numerical schemes since they are directly related to finite differences.

The definition can be written in sense of a series of term, obtaining the next definition

$$ {}_{GL*}D_x^{\alpha}\psi(x) := \lim_{h\to 0} \frac{1}{h^{\alpha}} \sum_{i=0}^{[\frac{x}{h}]+1} (-1)^i \binom{\alpha}{i} \psi(x-(\alpha-i)h), . \tag{5} $$

The following coefficients, called "normalized" Grünwald weights,

$$ g_{\alpha,i} = \frac{\Gamma(i-\alpha)}{\Gamma(-\alpha)\Gamma(i+1)} = (-1)^i \binom{\alpha}{i}, $$

appear in some schemes, and have the properties

$$ g_{\alpha,i} = \begin{cases} = 1, & i = 0 \\ < 0, & i = 1, 2, \dots \end{cases}, \qquad g_{1+\alpha,i} = \begin{cases} = 1, & i = 0 \\ < 0, & i = 1 \\ > 0, & i = 2, 3, \dots \end{cases} \tag{6} $$

being $0 < \alpha < 1$. The $g_{\alpha,i}$'s also enjoy the properties

$$ \sum_{i=0}^{\infty} g_{\alpha,i} = 0, \quad \sum_{i=0}^{\infty} g_{1+\alpha,i} = 0. \tag{7} $$

These coefficients will be used explicitly in the numerical approximation of the Grünwald-Letnikov fractional discrete operator.

About the numerical integration methods for fractional order derivatives. but also integrals fractional computations, is important the recent paper by Brzeziński et al. [15], which presents high-accuracy methods.

It is also important the Riesz definition of fractional derivative, as the integer derivative of Riesz potentials, see [72], the definition of fractional derivative by Riesz is obtained through the Fourier transform. Consider

$$ \mathcal{F}\{DI^{1-\alpha}f\}(k) = (ik)|k|^{\alpha-1}\mathcal{F}\{f\}(k) = (i\,\mathrm{sign}\,k)|k|^{\alpha}\mathcal{F}\{f\}(k) $$

$$\mathcal{F}\{D\widetilde{I^{1-\alpha}}f\}(k) = (ik)(-i\,\mathrm{sign}\,k)|k|^{\alpha-1}\mathcal{F}\{f\}(k) = |k|^{\alpha}\mathcal{F}\{f\}(k)$$

for $0 < \alpha < 1$, i being the imaginary unit, D the *fractional differential operator*, I is the *fractional integral operator*, and $\widetilde{I}$ is the Fourier transform of I. The operator I^{α} applied to a function f can be defined as

$$(I^{\alpha}f)(x) = \frac{1}{\Gamma(\alpha)}\int_0^{\infty} y^{\alpha-1}f(x-y)\mathrm{d}y.$$

Recall the properties (see [72]),

$$\begin{aligned}\mathcal{F}\{I^{\alpha}f\}(k) &= |k|^{-\alpha}\mathcal{F}\{f\}(k),\\ \mathcal{F}\{\widetilde{I^{\alpha}}f\}(k) &= (-i\,\mathrm{sign}\,k)|k|^{-\alpha}\mathcal{F}\{f\}(k),\end{aligned}$$

where sign is the *signum* function. We can now consider

$$\frac{\mathrm{d}}{\mathrm{d}x}(\widetilde{I^{1-\alpha}f})(x) = \lim_{h\to 0}\frac{1}{h}[(\widetilde{I^{1-\alpha}f})(x+h) - (\widetilde{I^{1-\alpha}f})(x)] \tag{8}$$

as a definition for the Riesz fractional derivative.

Following [17], the *strong Riesz fractional derivative* of order α, ${}_R D^{\alpha}(f)$ of a function $f \in L^p(\mathbb{R}), 1 \le p < \infty$, is defined through the limit

$$\lim_{h\to 0}\left\|\frac{1}{h}(f * K_h^{1-\alpha}) - {}_R D^{\alpha}(f)\right\|_p = 0$$

whenever it exists. the convolution kernel, obtained from equation (8), is defined as

$$K_h^{1-\alpha} := \frac{1}{2\Gamma(1-\alpha)\sin(\alpha\pi/2)}\left[\frac{\mathrm{sign}(x+h)}{|x+h|^{\alpha}} - \frac{\mathrm{sign}\,x}{|x|^{\alpha}}\right],$$

see [72].

It is furthermore noteworthy to observe that whenever a given function has a continuous $(n-1)$th-order derivative with an integrable nth-order derivative, then the *Caputo*, the RL, the G-L and the definition through the Fourier transform, of fractional derivatives of order α, with $n-1 < \alpha < n$, coincide [89, 112].

To give an idea of the occurrence of a power law behavior, consider the Mittag-Leffler (M-L) functions [68] . These functions are important in the framework of fractional differential equations (fODE) because they play for them, in some sense, the same role that the exponential functions do in classical ODEs. In fact, as $v(t) := e^{-t}$ is the unique solution of the ODE

$$\frac{dv}{dt} = -v,\ t > 0,\quad v(0) = 1,$$

so the M-L function

$$e_\alpha(t) := E_\alpha(-t^\alpha) = \sum_{n=0}^{\infty}(-1)^n \frac{t^{\alpha n}}{\Gamma(\alpha n+1)},$$

with $t > 0$, $0 < \alpha < 1$, solves the fODE [fractional differential equation]

$${}_C D_t^\alpha u = -u,\ t > 0,\quad u(0^+) = 1.$$

We may also recall here the generalized M-L functions

$$E_{\alpha,\beta}(z) = \sum_{n=0}^{\infty}(-1)^n \frac{z^n}{\Gamma(\alpha n+\beta)},$$

with $z \in \mathbf{C}$, $\alpha, \beta > 0$, noting that $E_{\alpha,1}(z) \equiv E_\alpha(z)$. Other generalizations do exist, but we not need tem here.

It may be interesting to observe that when the celebrated Bateman project was carried out, in the 1950s, Special Functions such as the Mittag-Leffler functions and some of its generalizations were considered as no more than curiosities, and were relegated among other "miscellanea functions", in Volume 3 [56].

The aforementioned property of the M-L functions of solving the simplest Caputo fODE, confers them some centrality within theory and applications of fODEs. Consequently, their relevance in (pure and) applied mathematics and sciences has increased considerably.

Among the numerous properties of the M-L functions, we mention that $E(-t^\alpha)$ is an *entire* function of t^α, for every fixed value of α. Moreover, α can be complex, with $\mathrm{Re}(\alpha) > 0$. Note that for $\alpha = 1$ we obtain $e_1(t) = e^{-t}$, so that $e_\alpha(t)$ is a generalization of e^{-t}. In Fig 1, we display the behavior of the M-L functions $e_\alpha(t)$ for few values of α.

A special interest has the case of *fractional relaxations* as well that of *fractional oscillations*, which can de described by simple fODEs. This case is analogous to that of the corresponding classical oscillators. Thus, we considered and solved numerically equations where the *Riemann-Liouville* integrals were replaced by *Erdélyi-Kober*-type integrals, obtaining more general integro-differential equations. In particular, the fODEs we considered are characterized by linear fODEs of order α or $1+\alpha$, with $0 < \alpha < 1$, and *time variable coefficients*. It seems that fODEs with such last peculiarity were *not* studied yet in the literature, in particular numerically [35].

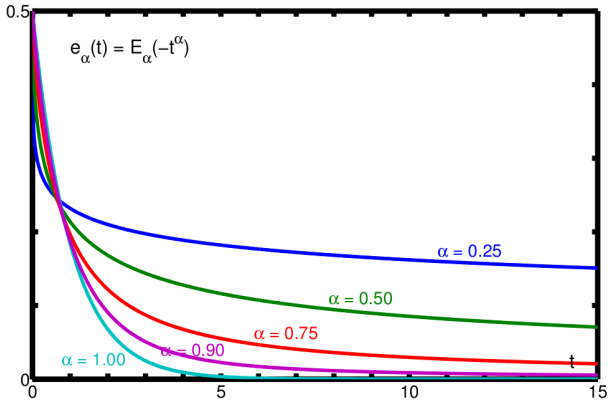

FIGURE 1: graphs of the M-L functions $e_\alpha(t) \equiv E(-t^\alpha)$, for various α (reproduced under permission of F. Mainardi).

In chapter 1, we discussed two issues concerning the M-L functions, namely, (i) a by now confirmed conjecture, made by F. Mainardi on estimates of the M-L functions in terms of certain simple rational functions, and (ii) a new numerical approach to compute efficiently their values.

In case of time-space dependent phenomena, fractional partial differential equations may play a fundamental role. Fractional partial differential equations (fPDEs) are obtained by generalizing classical partial differential equations (PDEs) to arbitrary (non integer) order. Since PDEs and in many instances even better fPDEs are used to model complex phenomena, they play a crucial role in Engineering, Physics and Applied Mathematics. Since fPDEs may account for memory effects, nonlocal relations in space and time, complex phenomena can be better modeled by using such equations. Due to this fact, applications in diverse fields such as materials with memory and hereditary effects, fluid flow through porous media, rheology, diffusive transport, electrical networks, electromagnetic theory, signal processing, and probability.

Flow through porous media, for instance, is a subject of interest in many branches of applied Science and Engineering, i.e., hydrology, geology, chemical engineering, and oil extraction. Investigating the relevant features of such processes is a major task in understanding phenomena, such as subsidence due to water shortage, or the process of crystallization of the ores in a well thermal exit, which makes them unusable for the heat extraction [119]. Researchers have been focusing on establishing relations among piezometric pressure gradients, macro-scale geometric properties of the medium, thermophysical properties of both, medium and fluid, and the fluid velocity field relative to the solid matrix ([138]). The Darcy law is usually employed in this analysis.

However, it has been checked experimentally [5, 55, 116] that, when a fluid flows through a porous medium, the permeability of the matrix may vary locally in time [22, 27, 75],

for a number of reasons. These are chemical dissolution of the medium, swelling and flocculation, pore plugging, and precipitation reactions, transport of particles obstructing the pores, mechanical compactification, and grain crumbling due to high pressure [5, 21]. All these phenomena, together with possible chemical reactions between the fluid and the host medium, create continuously local changes of the porosity, resulting in memory effects. This means that, at a given time, the advection process is affected by the past history of pressure and flow [75]. Consequently, the equations governing advection are affected by certain peculiarities, since the variation of the permeability is not known in advance [24]. Many researchers have attempted to extend Darcy's law but, due to the complexity of the problem, the first attempts merely relied on intuitive or empirical models [77, 138].

While, historically, only one parameter was introduced in Darcy's law to relate the pressure gradient to the fluid velocity, the flow rate was made depend on the *order* of the fractional derivative which replaces now the classical derivative, as well as on several other parameters (the memory parameters), which describe mathematically the changes of both, the solid matrix and the fluid characteristics, due to the aforementioned phenomena of crumbling, mechanical compacting, and chemical reactions.

A number of laboratory experiments have validated such a generalized theory. Over the last 30 years, fractional differential equations have been adopted to mimic memory effects in electromagnetism [76], biology [25], chaos [92], economy [23]), and to describe the rheological properties of solids [1, 4, 41, 82]. as for the memory effects on diffusion and propagation [92], a general framework can be provided by fractional-type model equations.

In a recent study accomplished by Iaffaldano et al. [75], the *constitutive* equations of *diffusion* have been rewritten in such a form in order to provide a memory model for diffusion of water through porous media. This model reproduces well the flux rate observed in experiments of water flow through certain media such as sand.

Since explicit analytical solutions to fractional diffusion equations in general are not available, several numerical methods have been constructed to evaluate them

In chapter 2, in order to give some motivation about the *fractional derivatives* approach to model some physical dynamics, e.g. transport of some fluids through porous media, have been compared numerical results with *laboratory tests* [75].

In [50], e.g., in order to adequately represent these phenomena, have been introduced a memory formalism operating on some physical like pressure gradient–flux and the pressure–density variations. The memory formalism is then represented with fractional

order derivatives. Have been perform a number of laboratory experiments in uniformly packed columns where a constant pressure is applied on the lower boundary. Both homogeneous and heterogeneous media of different characteristic particle size dimension were employed. The low value assumed by the memory parameters, and in particular by the fractional order, demonstrates that memory is largely influencing the experiments. The data and theory show how mechanical compaction can decrease permeability and consequently flux. Have been considered the three constitutive equations of the classical theory of advection state

$$\mathbf{q} = -D\nabla p$$
$$p = Gm$$
$$\frac{\partial m}{\partial t} + \nabla \mathbf{q} = 0$$

where $\mathbf{q}$ is the flux rate, D is the *Darcy* coefficient depending on both the porous matrix and the fluid phase characteristics, m is the fluid mass per unit volume, and p is pressure. Have been a constitutive equation containing a memory formalism to replace the classic *Darcy's law* as

$$f_1(t) * \mathbf{q} = -f_2(t) * \nabla p$$

where the symbol " * " indicates convolution, and $f_1(t)$ and $f_2(t)$ are integro-differential operators. Using some mathematics and physical considerations they attain the modified equations

$$\left(\gamma + \varepsilon \frac{\partial^{n_1}}{\partial t^{n_1}}\right) \mathbf{q} = -\left(c + d\frac{\partial^{n_2}}{\partial t^{n_2}}\right) \nabla p,$$
$$\left(a + b\frac{\partial^{m_1}}{\partial t^{m_1}}\right) p = \left(\alpha + \beta \frac{\partial^{m_2}}{\partial t^{m_2}}\right) m$$

$0 \leq n_1 < 1, 0 \leq n_2 < 1, 0 \leq m_1 < 1$, and $0 \leq m_2 < 1$, $a, b, c, d, \alpha, \beta, \gamma$, and ε are the so called "memory parameters", see [50].

In our case, we noted that for certain values of fractional derivative orders and initial conditions, the numerical results with *laboratory tests* approaches. Have been used a 1D in space diffusion fPDE. We have considered fractional order derivatives in space and in time, by varying those exponents we have seen that for certain values of the equation fits better the experimental data than in case of integer order derivatives.

In chapter 3 we have considered, as a prototype of a 2D (and a 1D) fractional diffusion equation

$$\frac{\partial u}{\partial t} = \mathbf{K} \cdot \nabla^{\gamma} \mathbf{u} + f(u; x, y, t) = 0 \qquad \text{in } \Omega \times (0, T], \tag{9}$$

where $\mathbf{K} = (K_x, K_y), \nabla^\gamma = (\frac{\partial^{\gamma x}}{\partial x^{\gamma x}}, \frac{\partial^{\gamma y}}{\partial y^{\gamma y}}), 1 < \gamma_x, \gamma_y \leq 2$, $\Omega := (0, L_x) \times (0, L_y)$. Dirichlet boundary conditions and an initial condition like

$$u(x, y, t)|_{\partial\Omega} = \Phi(x, y), \qquad u(x, y, 0) = \varphi(x, y), \tag{10}$$

are imposed, for some given functions Φ and φ. Throughout this Thesis, we will assume that such an initial-boundary value problem has a unique, sufficiently smooth solution.

Here the operator $\frac{\partial^{\gamma x}}{\partial x^{\gamma x}}$ denotes the fractional Riemann-Liouville derivative of order γ_x with respect to x; similarly for the variable y.

In [26, 43, 44], a numerical method, third-order accurate in time, has been proposed (but *not* implemented) in [26] to solve two-dimensional fPDEs of the diffusive type. This approach consists of combining the idea of the Alternating Direction Implicit (ADI) method with a Crank-Nicolson discretization to obtain a third-order accurate in time finite difference algorithm. Results concerning the numerical treatment of fPDEs in several space dimensions, that can be found in the existing literature are still few, which fact calls for effective numerical methods, say, for two- and three-dimensional fPDEs.

We have implemented for the first time (to the best of our knowledge), a third-order in time and second-order in space accurate Crank-Nicolson "weighted and shifted" Grünwald difference (CN-WSGD) scheme, to solve two-dimensional fractional diffusion and reaction diffusion equations. In addition, a new three-step extrapolation method has been derived and applied, also for the first time, to solve the same problems. Finally, an acceleration technique consisting of an *optimized extrapolation* method has been introduced here to improve the convergence rate of the aforementioned ADI-like method [64].

We remark at this point that the so-called *fractional Laplacian* is something different. The fractional Laplacian $(-\Delta)^{\alpha/2}$ is an operator which reduces to the standard Laplacian when $\alpha = 2$, used in the field of fractional calculus for modeling nonlocal diffusions. One can think of $-(-\Delta)^{\alpha/2}$ as the most basic elliptic linear integro-differential operator of order α and can be defined in several equivalent ways, see [18, 59]. A range of powers of special interest is $\alpha \in (1, 2)$. A definition of the operator is given by a singular integral

$$(-\Delta)^{\alpha/2} u(x) := C_{n,\alpha} \int_{\mathbb{R}^n} \frac{u(x) - u(y)}{|x - y|^{n+\alpha}} \mathrm{d}y,$$

where the constant $C_{n,\alpha}$ is given by

$$C_{n,\alpha} = \frac{\alpha 2^{\alpha-1} \Gamma\frac{\alpha+n}{2}}{\pi^{n/2} \Gamma\frac{2-\alpha}{2}}$$

In probability theory, the fractional Laplacian is the infinitesimal generator of a symmetric $\alpha - stable$ Lévy process. In financial mathematics, it appears as an alternative model to Brownian motion, incorporating the jumps in asset prices. These representations are shown to be equivalent in [81, 126, 136]. Many numerical method for the fractional Laplacian have been proposed, we recall Huang and Oberman which in [74] have introduced a method based on the singular integral representation for the operator, the method combines finite differences with numerical quadrature to obtain a discrete convolution operator with positive weights.

In chapter 4 we consider fractional diffusion equation, for the 3D space case. It was observed [71] that the celebrated Darcy's law, along with the assumption of continuity of the seepage flow in heterogeneous media, and the ensuing diffusion partial differential equation (PDE), do *not* hold in the general case of real seepage flows.

In this chapter, a new well "balanced", fractional version of the Alternating Direction Implicit (ADI) method is introduced to solve numerically 3D diffusion and reaction-diffusion fPDEs, which can be applied, e.g., to describe fluid flows through porous media. We term "balanced" our scheme since we distribute the right-hand side in the ADI scheme in equal amount among the three equations representing the three steps steps of the ADI method, instead of using the full right-hand side in the first equation only. This choice allows us to increase the space accuracy from the first to the second order. On the other hand, Liu et al [88] derived a scheme accurate to the second order in space resorting to Richardson extrapolation.

Our method is shown to be *unconditionally stable* for every fractional order of space derivatives, is *second-order* accurate *in space* (as stated here above), and *third-order* accurate *in time*. This latter property is here established resorting to an extrapolation technique along with an optimization realized through Google's *PageRank* algorithm. The method followed here to do this is similar to that used for the 2D case in [33]. Liu et al.'s method is only of the second order in time.

The for the 3D cases we set up the typical problems, and discuss some fractional versions of the ADI (fADI) 3D schemes, we establish a convergence result, and some details about optimization and extrapolation are given, and a few numerical examples are presented.

Chapter 1

Some analytical and numerical properties of the Mittag-Leffler functions

In this Chapter we discuss two issues concerning the M-L functions, namely, (i) a conjecture, made by F. Mainardi on estimates of the M-L functions in terms of certain simple rational functions, and (ii) a new numerical approach to compute efficiently values of them. We first recall, for convenience, some known theoretical results on the Cauchy problem for fODEs.

1.1 Ordinary fractional differential equations. Existence and uniqueness theorems

In this section we give a brief overview of the results on existence and uniqueness for ordinary differential equations of fractional order on a finite interval of the real axis [79].

Most of the investigations in this field involve existence and uniqueness of solutions to fractional differential equations with Riemann-Liouville fractional derivative $(D^{\alpha}_{a+}y)(x)$ defined for $(\Re(\alpha) > 0)$. The "model" nonlinear differential equation of fractional order α $(\Re(\alpha) > 0)$ on a finite interval $[a, b]$ of the real axis $\mathbb{R}$ has the form

$$(D^{\alpha}_{a+}y)(x) = f(x, y(x)) \quad (\Re(\alpha) > 0; x > a), \tag{1.1}$$

with initial conditions

$$(D_{a+}^{\alpha-k}y)(a+) = b_k, \quad b_k \in \mathbb{C} \quad (k = 1, \dots, n), \tag{1.2}$$

where $n = \Re(\alpha) + 1$ for $\alpha \notin \mathbb{N}$ and $\alpha = n$ for $\alpha \in \mathbb{N}$.

In this subsection we establish the existence of a unique solution to the Cauchy type problem (1.1),(1.2) in the space $\mathbf{L}^{\alpha}(a, b)$ defined as

$$\mathbf{L}^{\alpha}(a, b) := \{y \in L(a, b) : D_{a+}^{\alpha} y \in L(a, b)\}. \tag{1.3}$$

Here $L(a, b) := L^1(a, b)$ is the space of summable functions in a finite interval $[a, b]$ of the real axis $\mathbb{R}$. Consider now some relations.

When $0 < \Re(\alpha) < 1$, the problem (1.1)-(1.2) takes the form

$$(D_{a+}^{\alpha}y)(x) = f(x, y(x)), \quad (I_{a+}^{1-\alpha}y)(a+) = b \in \mathbb{C}, \tag{1.4}$$

where $I_{a+}^{1-\alpha}$ represents the *fractional integral* of order $1 - \alpha$ calculated in $a+$, see [3],

$$(D_{a+}^{\alpha}y)(x) = f(x, y(x)), \quad (\alpha > 0), \tag{1.5}$$

$$(D_{a+}^{\alpha-k}y)(a+) = b_k, \quad b_k \in \mathbb{R} \quad (k = 1, \dots, n; \quad n = -[-\alpha]), \tag{1.6}$$

and introduce the nonlinear Volterra integral equation of the second as

$$y(x) = \sum_{j+1}^{n} \frac{b_j}{\Gamma(\alpha - j + 1)}(x - a)^{\alpha-j} + \frac{1}{\Gamma(\alpha)} \int_a^x \frac{f(t, y(t))}{(x - t)^{1-\alpha}} \mathrm{d}t, \quad (x > a), \tag{1.7}$$

if $n = 1$ the Volterra nonlinear integral equation (1.7) goes into

$$y(x) = \frac{b_1(x - a)^{\alpha-1}}{\Gamma(\alpha)} + \frac{1}{\Gamma(\alpha)} \int_a^x \frac{f(t, y(t))}{(x - t)^{1-\alpha}} \mathrm{d}t, \quad (x > a, 0 < \alpha \leq 1). \tag{1.8}$$

Here we recall few basic theorems concerning existence and uniqueness of the Cauchy problem for fODEs [47].

Theorem 1. Let $\alpha > 0, n = -[-\alpha]$. Let G be an open set in $\mathbb{R}$. and let $f : (a, b] \times G \to \mathbb{R}$ be a function such that $f(x, y) \in L(a, b)$ for any $y \in G$. If $y(x) \in L(a, b)$, then $y(x)$ satisfies a.e. the relations (1.5) and (1.6) if, and only if, $y(x)$ satisfies a.e. the integral equation (1.7).

Theorem 2. Let $\alpha \in \mathbb{C}$ and $n - 1 < \Re(\alpha) < n \quad (n \in \mathbb{N})$. Let G be an open set in $\mathbb{C}$ and let $f : (a, b] \times G \to \mathbb{C}$ be a function such that $f(x, y) \in L(a, b)$ for any $y \in G$. If $y(x) \in L(a, b)$, then $y(x)$ satisfies a.e. the relations (1.1) and (1.2) if, and only if, $y(x)$

satisfies a.e. the equation (1.7). In particular, if $0 < \Re(\alpha) < 1$, then $y(x)$ satisfies a.e. the relations in (1.4) if, and only if, $y(x)$ satisfies a.e. the equation (1.8).

We will consider an additional Lipschitzian-type condition on $f(x, y)$ with respect to the second variable: for all $x \in (a, b]$ and for all $y_1, y_2 \in G \subset \mathbb{C}$,

$$|f(x, y_1) - f(x, y_2)| \leq A|y_1 - y_2|, \quad (A > 0), \tag{1.9}$$

where $A > 0$ does not depend on $x \in [a, b]$. First we derive a unique solution to the Cauchy-problem (1.5)-(1.6) with a real $\alpha > 0$.

Theorem 3. Let $\alpha > 0, n = -[-\alpha]$. Let G be an open set in $\mathbb{R}$ and let $f : (a, b] \times G \to \mathbb{R}$. be a function such that $f(x, y) \in L(a, b)$ for any $y \in G$ and the condition (1.9) is satisfied. Then there exists a unique solution $y(x)$ to the Cauchy type problem (1.5)- (1.6) in the space $\mathbf{L}(a, b)$.

Theorem 3 is extended from a real $\alpha > 0$ to a complex $\alpha \in C(\Re(\alpha) > 0)$.

Theorem 4. Let $\alpha \in \mathbb{C}, n - 1 < \Re(\alpha) < n \quad (n \in \mathbb{N})$. Let G be an open set in $\mathbb{C}$ and let $f : (a, b] \times G \to \mathbb{C}$ be a function such that $f(x, y) \in L(a, b)$ for any $y \in G$ and the condition (1.9) holds. Then there exists a unique solution $y(x)$ to the Cauchy type problem (1.1)- (1.2) in the space $\mathbf{L}(a, b)$ defined in (1.3). In particular, if $0 < \Re(\alpha) < 1$, then there exists a unique solution $y(x)$ to the Cauchy type problem

$$(D^{\alpha}_{a+}y)(x) = f(x, y(x)) \quad (0 < \Re(\alpha) < 1) \quad (I^{1+\alpha}_{a+}y)(a+) = b \in \mathbb{C} \tag{1.10}$$

in the space $\mathbf{L}(a, b)$.

Now we will establish the existence of a unique solution to the Cauchy type problem (1.5)-(1.6) in the space $\mathbf{C}^{\alpha}_{n-\alpha}[a, b]$ defined as

$$\mathbf{C}^{\alpha}_{n-\alpha}[a, b] = \{y(x) \in C_{n-\alpha}[a, b] : (D^{\alpha}_{a+}y)(x) \in C_{n-\alpha}[a, b]\}. \tag{1.11}$$

under the conditions stated in the following Theorem and an additional Lipschitzian condition (1.9).

Theorem 5. Let $\alpha > 0, n = -[-\alpha]$. Let G be an open set in $\mathbb{R}$. and let $f : (a, b] \times G \to \mathbb{R}$ be a function such that $f(x, y) \in C_{n-\alpha}[a, b]$ for any $y \in G$. If $y(x) \in C_{n-\alpha}[a, b]$, then $y(x)$ satisfies the relations (1.5) and (1.6) if, and only if, $y(x)$ satisfies the Volterra integral equation (1.7).

Further results can be found in [47].

1.2 Proving part of Mainardi's conjecture

In [95], F. Mainardi formulated the following *conjecture*: For every $t > 0$, and for every fixed α, with $0 < \alpha < 1$, the estimates

$$g_\alpha(t) \le e_\alpha(t) \le f_\alpha(t)$$

hold, where

$$g_\alpha(t) := \frac{1}{1 + t^\alpha \Gamma(1-\alpha)},$$

$$e_\alpha(t) := \sum_{n=0}^{\infty} (-1)^n \frac{t^{\alpha n}}{\Gamma(\alpha n + 1)},$$

$$f_\alpha(t) := \frac{1}{1 + \frac{t^\alpha}{\Gamma(1+\alpha)}}.$$

The three functions involved, $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$, are all known to be *completely monotone* functions (at least for $t > 0$ and for every fixed $\alpha \in (0,1)$), thus they can be considered as the Laplace transforms of positive measures (in particular, Laplace transforms of nonnegative integrable functions). Therefore, one may expect to be able to exploit such a property to prove this conjecture. This is still an open way to test.

An attempt was done in [149], where the authors provide better approximations for the M-L function via global Padé approximants, but neither lower nor upper bounds have been obtained there. In a recent paper [137], the Mittag-Leffler function $e_\alpha(t)$ was computed suitably interpolating the two functions $f_\alpha(t)$ and $g_\alpha(t)$, i.e., approximating it by a linear combination of the form $\phi_\alpha(t)\, f_\alpha(t) + (1 - \phi_\alpha(t))\, g_\alpha(t)$, and finding an explicit expression for the function $\phi_\alpha(t)$. This procedure, however, cannot prove anything rigorously and, moreover, in [137] it was even stated that, for $\alpha < 0.5$ and $t > 1$, results were not considered, owing to an observed numerical instability.

Instead, very recently, T. Simon was able to prove Mainardi's conjecture basing his proof on probabilistic arguments [120]. In [34], it was proved part of Mainardi's conjecture, only on some neighborhood of $t = 0$, where however the inequalities appear to be sharper. The contribution in [34] is quite limited, but the result is obtained by simple classical estimates, and concerns small values of t, where the conjecture seems to be more critical, cf. Fig 1.

In [34], the results obtained by some new numerical approach where compared with those provided by the code available through MATLAB, as well as that of *Mathematica*. The method adopted in [34] consists of an *adaptive* predictor-corrector scheme, based on K. Diethelm's predictor-corrector algorithm [46, 48, 49]. Even though the idea of

computing the M-L function integrating the fODE that it solves is rather trivial, it turns out that such algorithm outperforms the other two methods, both in terms of smaller CPU time and of smaller RAM required. It should be noted, however, that the other codes are more far reaching, since they are capable to evaluate the more general two-parameters M-L function $E_{\alpha,\beta}(z)$, and moreover they can do it on the complex domain.

This conjecture was supported by rather extensive numerical evaluations of the three functions, $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$. Note that the prospective *upper bound* [i.e., $f_\alpha(t)$] is related to the convergent power series approximation

$$e_\alpha(t) \sim 1 - \frac{t^\alpha}{\Gamma(1+\alpha)},$$

valid for small values of t, while the prospective *lower bound* [i.e., $g_\alpha(t)$] refers to the asymptotic approximation

$$e_\alpha(t) \sim \frac{t^{-\alpha}}{\Gamma(1-\alpha)},$$

valid for large values of t. Both approximations can be related to the [0/1] *Padé* approximants of $e_\alpha(t)$. More precisely, it was known that

$$e_\alpha(t) \sim \sum_{n=1}^{\infty} (-1)^{n-1} \frac{t^{-\alpha n}}{\Gamma(1-\alpha n)}, \quad t \to +\infty,$$

(see Erdélyi et al. [56], i.e., the Bateman project's Volume 3, 1953), and that

$$e_\alpha(t) = 1 - \frac{t^\alpha}{\Gamma(1+\alpha)} + \ldots \sim \exp\left[-\frac{t^\alpha}{\Gamma(1+\alpha)}\right], \quad t \to 0^+.$$

F. Mainardi and his collaborators reported that their numerical computations, made to validate such conjecture, were accurate to within 1% (relative errors), see Figs. 1.1–1.5.

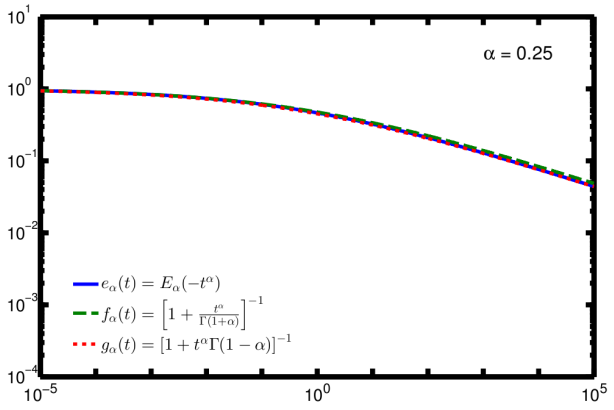

FIGURE 1.1: Graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.25$ (reproduced under permission of F. Mainardi).

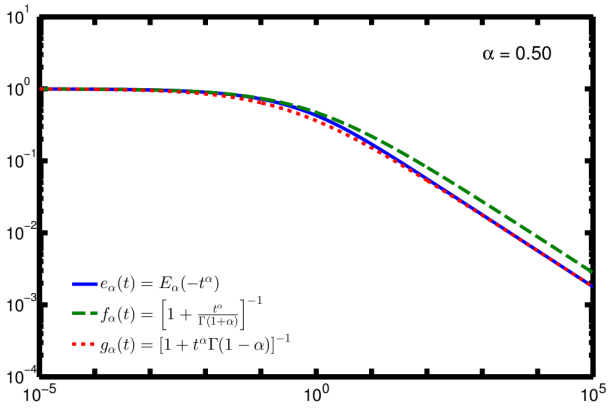

FIGURE 1.2: Graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.50$ (reproduced under permission of F. Mainardi).

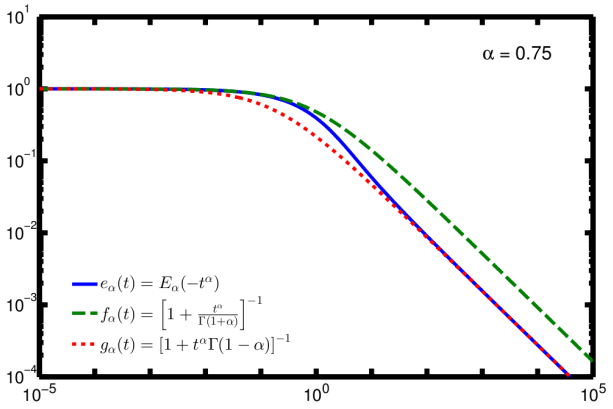

FIGURE 1.3: Graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.75$ (reproduced under permission of F. Mainardi).

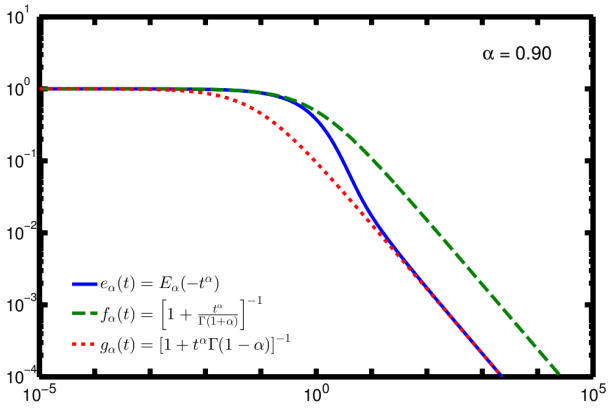

FIGURE 1.4: Graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.90$ (reproduced under permission of F. Mainardi).

We computed the same plots with a higher accuracy, wherefrom the conjecture seemed to be valid within 0.001% (i.e., with errors less than 10^{-5}), see Figs. 1.6–1.11. All these

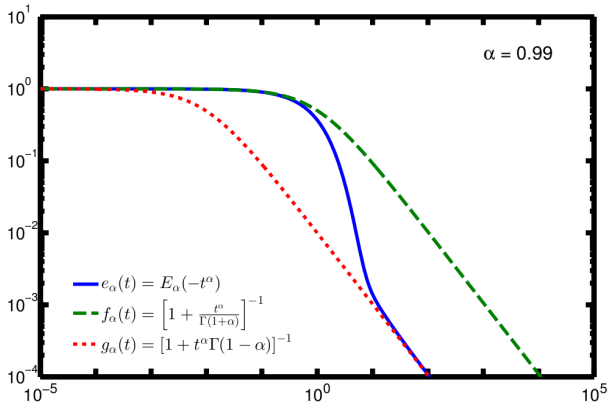

FIGURE 1.5: Graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.99$ (reproduced under permission of F. Mainardi).

are of course merely numerical experiments, and only a theoretical proof can decide on the correctness of the conjecture.

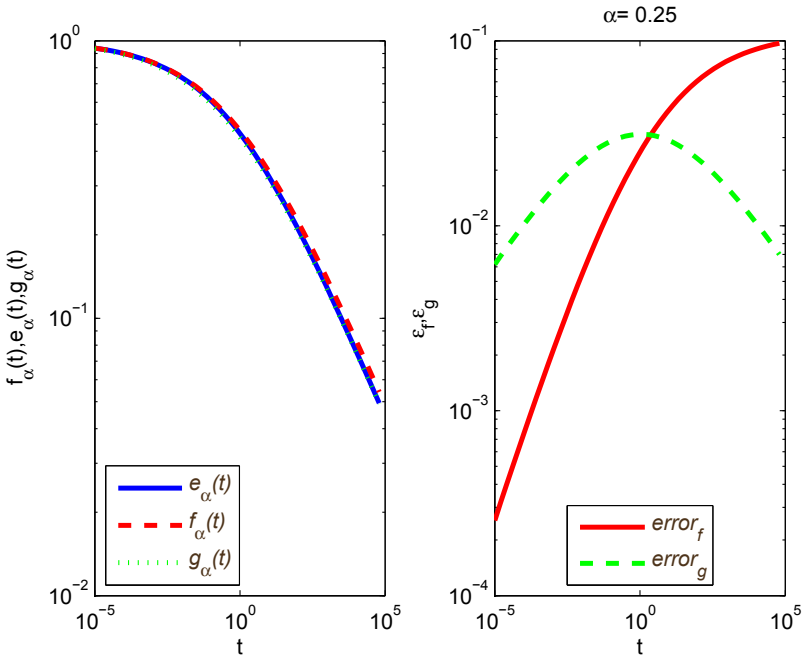

FIGURE 1.6: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.25$.

Here we follow a direct approach to prove the conjecture, i.e., we compare certain power series, considering that the rational functions g_α and f_α can be expanded in geometric series, on suitable t-intervals. Recall that

$$\frac{1}{1+x} = \sum_{n=0}^{\infty} (-1)^n x^n,$$

for $|x| < 1$, *but* we can write

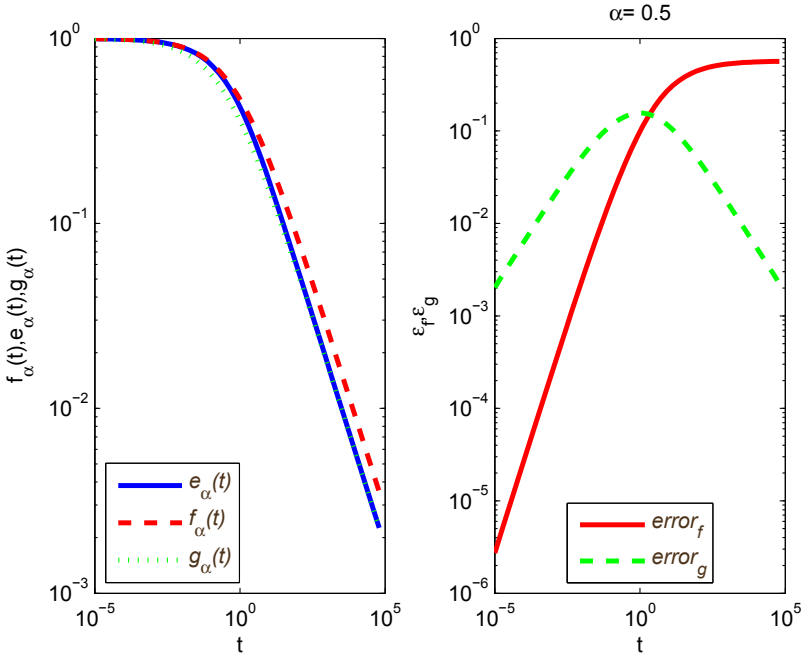

FIGURE 1.7: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.50$.

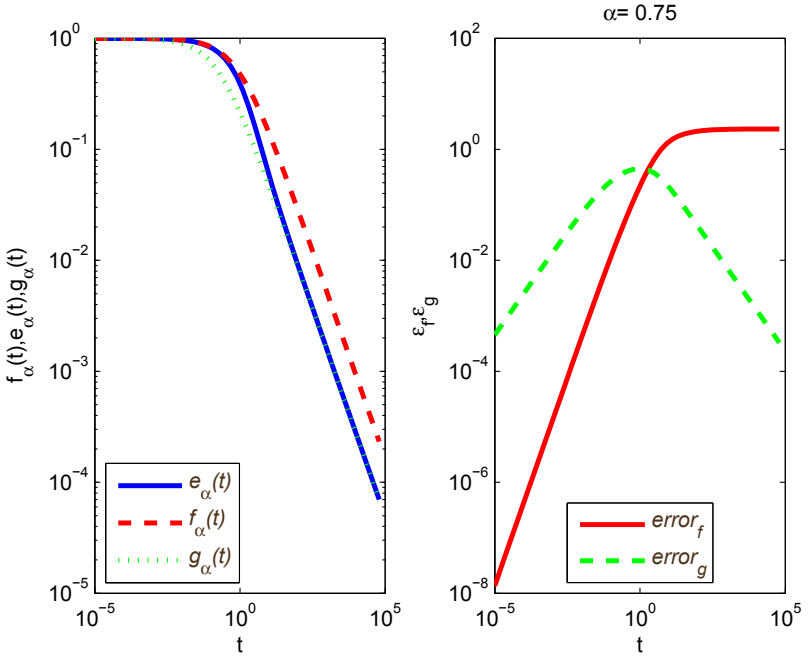

FIGURE 1.8: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.75$.

$$\frac{1}{1+x} = \frac{1}{x}\frac{1}{1+\frac{1}{x}} = \sum_{n=0}^{\infty}(-1)^n x^{-n-1}$$

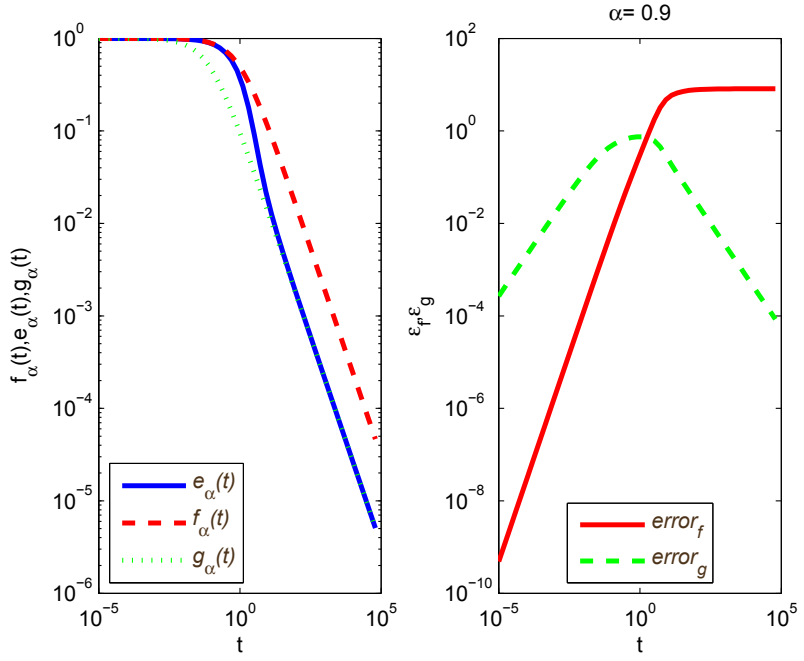

FIGURE 1.9: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.90$.

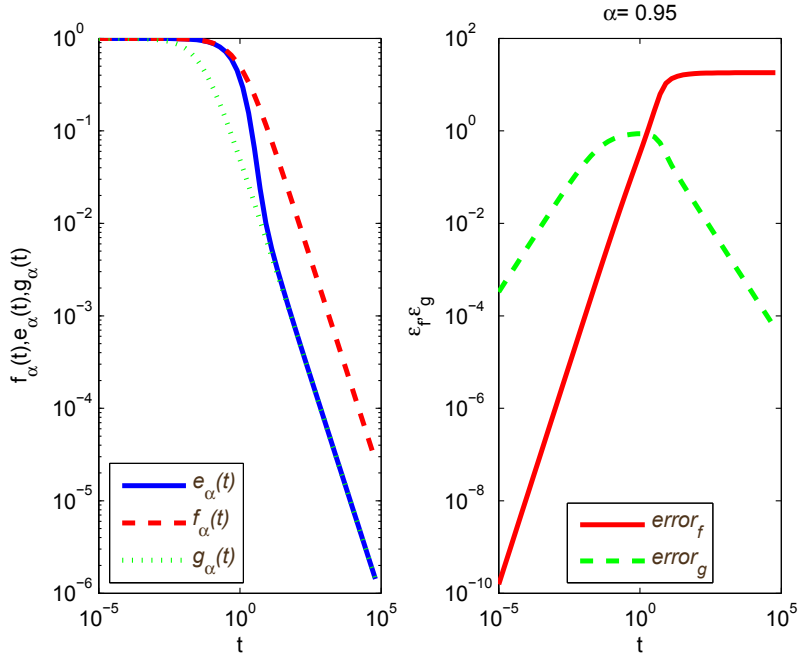

FIGURE 1.10: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.95$.

when $|x| > 1$. The special case $x = 1$ (for us only $x > 0$ matters) should be treated separately.

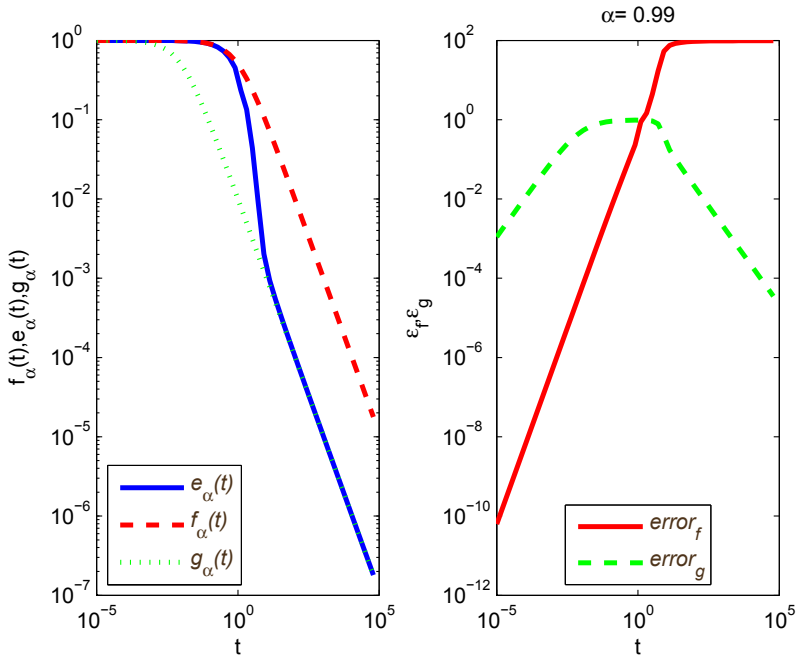

FIGURE 1.11: Our graphs of $g_\alpha(t)$, $e_\alpha(t)$, and $f_\alpha(t)$ for $\alpha = 0.99$.

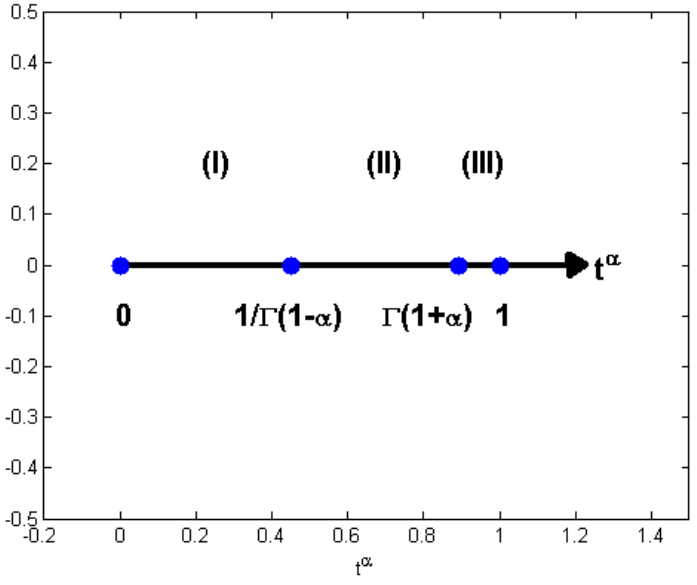

FIGURE 1.12: $x = t^\alpha$-axis, $\alpha = 0.6$.

In our problem, either $x := t^\alpha\Gamma(1-\alpha)$, or $x := t^\alpha/\Gamma(1+\alpha)$. It seems natural divide it into three cases (see Fig. 1.12), according to

(I) $0 < t^\alpha < 1/\Gamma(1-\alpha) \qquad (< \Gamma(1+\alpha) < 1)$;

(II) $1/\Gamma(1-\alpha) < t^\alpha < \Gamma(1+\alpha) \qquad (< 1)$

(III) $t^\alpha > \Gamma(1+\alpha))$ $\qquad (> 1/\Gamma(1-\alpha))$.

besides the two cases for $t^\alpha = 1/\Gamma(1-\alpha)$, and $t^\alpha = \Gamma(1+\alpha)$. Note in fact that, being $0 < \alpha < 1$, we have

$$0 < \frac{1}{\Gamma(1-\alpha)} < \Gamma(1+\alpha) < 1,$$

and recall moreover that

$$\Gamma(1-\alpha)\Gamma(1+\alpha) = \frac{\pi\alpha}{\sin\pi\alpha} \geq 1,$$

and $1 < 1+\alpha < 2$, so that $0 < \Gamma(1+\alpha) < 1$, see Fig. 1.13: $0 < \Gamma(x) < 1$ for $1 < x < 2$, and $\Gamma(x) > 1$ for $0 < x < 1$.

All series involved are *alternating series*, i.e., their terms have *alternating sign*. Recall that, when, in

$$\sum_{n=0}^{\infty}(-1)^n a_n = a_0 - a_1 + a_2 - a_3 + \ldots,$$

we have $a_n \geq 0$ for every n, and $a_n \to 0$ as $n \to \infty$ *monotonically*, not only convergence is guaranteed, but estimating the sum, say S, is possible, according to:

$$S \geq 0, \quad S \leq a_0, \quad S \geq a_0 - a_1, \quad S \leq a_0 - a_1 + a_2, \quad \text{etc.}$$

This property holds for *all* the geometric series involved, *but* it does *not* hold for the M-L functions, since

$$\frac{x^n}{\Gamma(\alpha n+1)}$$

does *not* tend to zero *monotonically*, as $n \to \infty$, for every fixed $x > 0$ (for every α with $0 < \alpha < 1$ and $n \in \mathbf{N}$, we have $0 < \alpha\, n < 1$ for all $n < 1/\alpha$, hence $\Gamma(\alpha\, n+1) < 1$ if $1 < \alpha\, n + 1 < 2$).

In this paper, we are only able to prove part of the conjecture, proceeding as done below. We notice, however that the interval containing the smallest values of x (or t), is also the most critical one, since, according to the numerical results, the validity of the conjecture is more delicate to establish for such values, see Figs. 1.1–1.5, and Figs. 1.6–1.11.

(1) We first show that, for every $0 < x := t^\alpha < 1/\Gamma(1-\alpha)$ (case (I)), and for every fixed $0 < \alpha < 1$, the M-L function $e_\alpha(t)$ is a series with alternating signs *and indeed* with the general term decaying *monotonically* to 0 as $n \to \infty$.

Since the two geometric series corresponding to the rational functions $g_\alpha(t)$ and $f_\alpha(t)$ enjoy *the same* property *in such* interval, estimates can be established in the form

(2)
$$g_\alpha(t) \le 1 - x\Gamma(1-\alpha) + x^2\Gamma(1-\alpha)^2 \le 1 - \frac{x}{\Gamma(1+\alpha)} \le e_\alpha(t)$$

for the *lower* bound, and

(3)
$$e_\alpha(t) \le 1 - \frac{x}{\Gamma(1+\alpha)} + \frac{x^2}{\Gamma(2\alpha+1)}$$
$$\le 1 - \frac{x}{\Gamma(1+\alpha)} + \frac{x^2}{\Gamma(1+\alpha)^2} - \frac{x^3}{\Gamma(1+\alpha)^3} \le f_\alpha(t)$$

for the *upper* bound.

As for (1), we define

$$F(x,y) := \frac{x^y}{\Gamma(\alpha y+1)} =: \varphi(y),$$

i.e., we replace $n \in \mathbf{N}$, with $y \in \mathbf{R}_0^+$, for fixed x and α, with $0 < x := t^\alpha < 1/\Gamma(1-\alpha)$, $0 < \alpha < 1$. Thus, we obtain $\varphi(0) = 1$ and

$$\varphi'(y) < 0 \quad \text{if and only if} \quad \psi(\alpha y+1) > \frac{1}{\alpha}\ln x \equiv \ln t, \tag{1.12}$$

where ψ denotes the logarithmic derivative of Γ.

Now, being just $t > 0$, the $\ln t$ on right-hand side can be either positive or negative. However, confining t to being $0 < t^\alpha < 1/\Gamma(1-\alpha)$ (case (I)), and being

$$\psi(\alpha y+1) \ge -\gamma,$$

where $\gamma = 0.57721...$ is the Euler-Mascheroni constant (since $\alpha y+1 \ge 1$), we infer that $\varphi'(y) < 0$ holds true provided that $0 < t^\alpha \le e^{-\alpha\gamma}$ (approximately, $0 < t < 0.56$, see Fig. 1.13, (b)).

Defining now

$$G(\alpha) := e^{-\alpha\gamma}\Gamma(1-\alpha)$$

for $0 < \alpha < 1$, we have $G(0^+) = 1$, and

$$G'(\alpha) = -e^{-\alpha\gamma}\Gamma(1-\alpha)[\gamma + \psi(1-\alpha)],$$

where from $G'(\alpha) \ge 0$ since $\psi(1-\alpha) \le -\gamma$. Therefore $G(\alpha) > 1$ for all $y > 0$, and this amounts to the validity of our claim, i.e., that indeed $t^\alpha < 1/\Gamma(1-\alpha) < e^{-\alpha y}$.

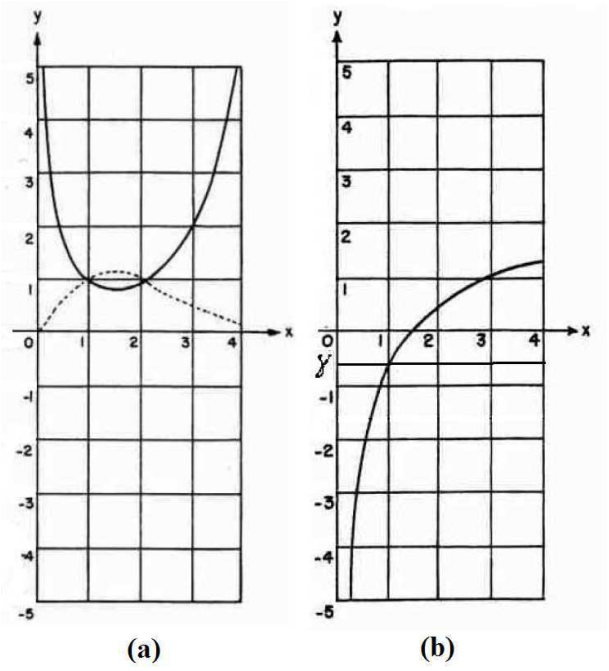

FIGURE 1.13: (a) The Gamma function, $\Gamma(x)$ and its reciprocal (dashed line); (b) the psi function, $\psi(x) := \frac{d}{dx}\log\Gamma(x) \equiv \frac{\Gamma'(x)}{\Gamma(x)}$.

As for (2), note that

$$g_\alpha(t) \le 1 - x\Gamma(1-\alpha) + x^2\Gamma(1-\alpha)^2 \le 1 - \frac{x}{\Gamma(1+\alpha)} \le e_\alpha(t),$$

after a little algebra, turns out to be equivalent to

$$t^\alpha\Gamma(1-\alpha) \le 1 - \frac{1}{\Gamma(1-\alpha)\Gamma(1+\alpha)} = 1 - \frac{\sin\pi\alpha}{\pi\alpha},$$

and the right-hand side is a number less than 1. Hence, we have an *effective restriction* to the values of t in case (I). It may be useful to plot the boundaries for t given by the previous relation, i.e., the function

$$F(\alpha) := \left[\left(1 - \frac{\sin\pi\alpha}{\pi\alpha}\right)\frac{1}{\Gamma(1-\alpha)}\right]^{1/\alpha},$$

for $0 < \alpha < 1$, see Fig. 1.14.

As for (3),

$$e_\alpha(t) \le 1 - \frac{x}{\Gamma(1+\alpha)} + \frac{x^2}{\Gamma(2\alpha+1)}$$

$$\le 1 - \frac{x}{\Gamma(1+\alpha)} + \frac{x^2}{\Gamma(1+\alpha)^2} - \frac{x^3}{\Gamma(1+\alpha)^3} \le f_\alpha(t),$$

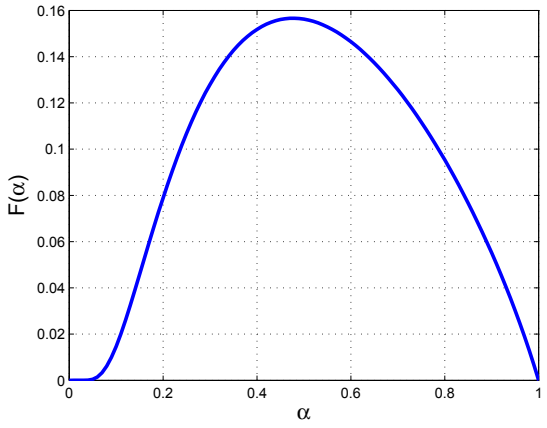

FIGURE 1.14: Function $F(\alpha)$ vs α, for $0 < \alpha < 1$.

it is equivalent to (after a little algebra)

$$0 < t^{\alpha} < \Gamma(1+\alpha)\left(1 - \frac{\Gamma(1+\alpha)^2}{\Gamma(2\alpha+1)}\right).$$

The fraction on the right-hand side can be shown to be positive, by Alsina and Tomàs' inequality [2],

$$\frac{1}{n!} \leq \frac{\Gamma(1+x)^n}{\Gamma(1+nx)} \leq 1, \quad 0 \leq x \leq 1,\ n \in \mathbf{N}.$$

Hence this is indeed an effective restriction to the values of t in case (I).

1.3 Discretization of ordinary fractional derivatives

We will show in the next a scheme used to discretize the fractional derivatives and compute the numerical solution of a fODE as in the next paragraph. It is known that the "left-sided" Riemann-Liouville as well as the Caputo fractional derivative, say, $v^{(\gamma)}(t) :=_a D_x^{\gamma} v(t)$, can be approximated, at each node of the evenly spaced grid $t := j\tau$, $j = 0, 1, \ldots, N$, being N the number of nodes for the planned simulation time, and τ the discretization time step-size, simultaneously, by the help of an upper triangular matrix B_N^{γ}, as

$$\begin{bmatrix} v_N^{(\gamma)} & v_{N-1}^{(\gamma)} & \ldots v_1^{(\alpha)} & v_0^{(\alpha)} \end{bmatrix}^T = B_N^{(\alpha)} \begin{bmatrix} v_N & v_{N-1} & \ldots v_1 & v_0 \end{bmatrix}^T \tag{1.13}$$

where

$$B_N^{(\gamma)} = \frac{1}{\tau^\gamma}\begin{bmatrix} \omega_0^{(\gamma)} & \omega_1^{(\gamma)} & \ddots & \ddots & \omega_{N-1}^{(\gamma)} & \omega_N^{(\gamma)} \\ 0 & \omega_0^{(\gamma)} & \omega_1^{(\gamma)} & \ddots & \ddots & \omega_{N-1}^{(\gamma)} \\ 0 & 0 & \omega_0^{(\gamma)} & \omega_1^{(\gamma)} & \ddots & \ddots \\ \cdots & \cdots & \cdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 0 & \omega_0^{(\gamma)} & \omega_1^{(\gamma)} \\ 0 & 0 & \cdots & 0 & 0 & \omega_0^{(\gamma)} \end{bmatrix} \tag{1.14}$$

$$\omega_j^{(\gamma)} = (-1)^j \binom{\gamma}{j}, \qquad j = 0, 1, ..., N, \tag{1.15}$$

see [113].

Here, since we have enumerated the discretization nodes in a decreasing order, the role of the matrices $B_N^{(\gamma)}$, originally intended for backward fractional differences, and $F_n^{(\gamma)}$, originally intended for forward fractional differences, is interchanged, comparing with the notation of [113], where such matrices were introduced for the first time. However, we keep the same notation, i.e. B_N^γ and $F_n^{(\gamma)}$, for the backward and the forward fractional difference approximation, respectively.

Similarly, the "right-sided" Riemann-Liouville or Caputo fractional derivative, $v^{(\gamma)}(t) :=_x D_b^\gamma v(t)$, can be approximated at each node of the evenly spaced grid, $t := j\tau$, $j = 0, 1, \ldots, N$, simultaneously, by the help of the upper triangular matrix F_N^γ,

$$\begin{bmatrix} v_N^{(\gamma)} & v_{N-1}^{(\gamma)} & ...v_1^{(\gamma)} & v_0^{(\gamma)} \end{bmatrix}^T = F_N^{(\gamma)} \begin{bmatrix} v_N & v_{N-1} & ...v_1 & v_0 \end{bmatrix}^T \tag{1.16}$$

$$F_N^{(\gamma)} = \frac{1}{\tau^\gamma}\begin{bmatrix} \omega_0^{(\gamma)} & 0 & 0 & 0 & \cdots & 0 \\ \omega_1^{(\gamma)} & \omega_0^{(\gamma)} & 0 & 0 & \cdots & 0 \\ \omega_2^{(\gamma)} & \omega_1^{(\gamma)} & \omega_0^{(\gamma)} & 0 & \cdots & 0 \\ \ddots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \omega_{N-1}^{(\gamma)} & \ddots & \omega_2^{(\gamma)} & \omega_1^{(\gamma)} & \omega_0^{(\gamma)} & 0 \\ \omega_N^{(\gamma)} & \omega_{N-1}^{(\gamma)} & \ddots & \omega_2^{(\gamma)} & \omega_1^{(\gamma)} & \omega_0^{(\gamma)} \end{bmatrix} \tag{1.17}$$

The symmetric Riesz derivative of order α can be approximated as the average of the left- and the right-sided Riemann-Liouville derivatives, see [112, 113],

$$\frac{d^\alpha \psi(x)}{d|x|^\alpha} =_R D_x^\alpha \psi(x) = \frac{1}{2}\Big({}_aD_x^\alpha \psi(x) + {}_xD_b^\alpha \psi(x)\Big), \tag{1.18}$$

hence it can be approximated as a combination of the approximated values given by (1.15) and (1.16), of the left- and right-sided Riemann-Liouville derivatives. Alternatively, one can use the centered fractional difference approximation of the symmetric Riesz derivative

proposed by M.D. Ortigueira [107, 108]. The general formula is the same, i.e.,

$$\begin{bmatrix} v_N^{(\alpha)} & v_{N-1}^{(\alpha)} & ...v_1^{(\alpha)} & v_0^{(\alpha)} \end{bmatrix}^T = R_M^{(\alpha)} \begin{bmatrix} v_M & v_{M-1} & ...v_1 & v_0 \end{bmatrix}^T \tag{1.19}$$

In the first case, the approximation for the left-sided Caputo derivative is evaluated one step ahead, while that for the right-sided Caputo derivative is taken one step back. This leads to the matrix

$$R_M^{(\alpha)} = \frac{h^{-\gamma}}{2} \left[{}_{-1}U_M + {}_{+1}U_M \right] \tag{1.20}$$

where ${}_{-1}U_M$ is defined by

$${}_{-1}U_M = S_1 E_{N+1,1}^{-} U_{N+1} E_{N+1,1}^{-} S_{N+1}^T \tag{1.21}$$

where E_n is the $n \times n$ identity matrix, S_1 is obtained omitting only the first row of E, $S_{1,2}$ is obtained omitting only the first and the second row of E, S_2 is obtained omitting only the second row, and, in general, $S_{r_1,r_2,...,r_k}$ is obtained omitting the rows identified by the numbers $r_1, r_2, ..., r_k$, $E_{N,p}^{+}, p = 1, ..., N$, with 1's on p-th diagonal above the main diagonal and zeros elsewhere, and matrices $E_{N,p}^{-}$, $p = 1, ..., N$, with 1's on p-th diagonal below the main diagonal and zeros elsewhere. Note that, according to the previous notation, $E_{N,0}^{\pm} \equiv E_N$ coincides with the identity matrix I_N. We denote with U_N the upper triangular matrix

$$U_N = \begin{bmatrix} \omega_0 & \omega_1 & \omega_2 & \ddots & \omega_{N-1}^{(\gamma)} & \omega_N^{(\gamma)} \\ 0 & \omega_0 & \omega_1 & \ddots & \ddots & \omega_{N-1} \\ 0 & 0 & \omega_0^{(\gamma)} & \ddots & \omega_2 & \cdots \\ 0 & 0 & 0 & \ddots & \omega_1 & \omega_2 \\ \vdots & \cdots & \cdots & \cdots & \omega_0 & \omega_1 \\ 0 & 0 & 0 & \cdots & 0 & \omega_0 \end{bmatrix} \tag{1.22}$$

while ${}_{+1}U_M$ is defined by

$${}_{+1}U_M = S_{N+1} E_{N+1,1}^{+} U_{N+1} E_{N+1,1}^{+} S_1^T. \tag{1.23}$$

We can then construct the following symmetric matrix

$$R_M^{(\alpha)} = \begin{bmatrix} \omega_0^{(\alpha)} & \omega_1^{(\alpha)} & \omega_2^{(\alpha)} & \omega_3^{(\alpha)} & \cdot & \omega_M^{(\alpha)} \\ \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & \omega_1^{(\alpha)} & \omega_2^{(\alpha)} & \cdot & \omega_{M-1}^{(\alpha)} \\ \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & \omega_1^{(\alpha)} & \cdot & \omega_{M-2}^{(\alpha)} \\ \vdots & \ddots & \ddots & \ddots & \cdots & \vdots \\ \omega_{M-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} & \omega_1^{(\alpha)} \\ \omega_M^{(\alpha)} & \omega_{M-1}^{(\alpha)} & \ddots & \omega_2^{(\alpha)} & \omega_1^{(\alpha)} & \omega_0^{(\alpha)} \end{bmatrix} \tag{1.24}$$

see [107],

$$\omega_k^{(\alpha)} = \frac{(-1)^k \Gamma(\alpha+1)\cos(\alpha\pi/2)}{\Gamma(\alpha/2-k+1)\Gamma(\alpha/2+k+1)}, \qquad k = 0, 1, \ldots, M. \tag{1.25}$$

Both these approximations of either Caputo or Riemann-Liouville derivatives yield, in practice, the same numerical results. In view of the numerical solution of fractional partial differential equations, this formulation leads to a well-posed problem for the resulting algebraic system.

1.4 A method for computing Mittag-Leffler functions

In this section, we compute the M-L function $e_\alpha(t)$ on some large t-interval of the real positive line, and for several values of α, $\alpha \in (0, 1)$, solving numerically certain fODEs satisfied by it.

Several algorithms have been developed over the years to compute the M-L functions. As often computing a variety of Special Functions, several different methods are adopted on different intervals of t: power series for small arguments, asymptotic approximations for large arguments, integral representations for intermediate values. A code is available in MATLAB, based on these strategies. This code is due to I. Podlubny and M. Kacenak, and was constructed following contributions of R. Gorenflo, J. Loutchko, and Yu. Luchko, see [66, 67]. Such MATLAB code is a routine which evaluates the Mittag-Leffler function with two parameters (sometimes called there "generalized exponential function"), see [99].

Recently R. Garrappa has written a code available in MATLAB routines [61, 62], which evaluates the Mittag-Leffler function with 1, 2 or 3 parameters. The three parameters ML function is also known as the *Prabhakar* function.

Within *Mathematica*, there is the function called *MittagLefflerE*, suitable for both, symbolic and numerical manipulations, providing values of the M-L function [98]. This code is based on the work of Gorenflo et al. [66, 67].

Since the M-L function plays for fODEs the same role of the exponential for classical ODEs, and in particular, $e_\alpha(t)$ solves the simple fODE

$$\text{(A)} \qquad {}^C D_t^\alpha u = -u, \ t > 0, \quad u(0^+) = 1,$$

where ${}^C D_t^\alpha u(t)$ is the Caputo derivative, one could evaluate $e_\alpha(t)$ *solving numerically* such a fODE. This is the method we shall adopt. More precisely, we used a predictor-corrector (pc, for short) method, based on the so-called Grünwald-Letnikov (G-L) approximation of the Caputo derivative, developed by K. Diethelm [46]. We then improve this algorithm setting up an *adaptive* (say, apc) version of it. The idea, as usual, is to choose locally a step size inversely proportional to the size of the derivative of the solution being computed. In practice, we choose, at the ith integration step, the time step

$$h_i := \frac{c\,h}{|e_\alpha(t_i) - e_\alpha(t_{i-1})|},$$

for $i = 1, 2, \ldots, N$, $t_i = \sum_{j=1}^{i} h_j$, $t_0 = 0$, h being the initial step size, and

$$c = \frac{e'_\alpha(t_{i-2}) - e'_\alpha(t_{i-3})}{e'_\alpha(t_{i-1}) - e'_\alpha(t_{i-2})},$$

where e'_α denotes the (classical) first derivative of e'_α.

Convergence and accuracy of methods of this kind were studied by K. Diethelm et al. in [49]. An implementation with multiple corrector iterations was proposed and discussed for "multiterm" fODEs in [48]. In such implementation, the discrete convolutions appearing in the G-L formulation are evaluated by means of the FFT algorithm as described in [70]. This allows to keep the computational cost to order $\mathcal{O}(N \log N)$, instead of $\mathcal{O}(N^2)$, as it occurs in any straightforward implementation. Here N is the number of time-points where the solution is evaluated, i.e., $h = T/N$, assuming $[0, T]$ to be the interval of integration. The stability properties of this implementation of the aforementioned Adams-Bashforth-Moulton-type method have been studied in [60].

In Figs. 1.15-1.19, we compare the numerical solution, $u(t)$, to equation (A), computed by Diethelm's pc algorithm, with $e_\alpha^{PK}(t)$, obtained by the MATLAB code due to Podlubny and Kacenak.

We further validate Diethelm's pc method computing the numerical errors made in a case when the M-L function is explicitly known, namely when $\alpha = 1/2$. In fact, $e_{1/2}(t) \equiv E_{1/2}(-t^{1/2}) = e^t \operatorname{erfc}(t^{1/2})$, which was computed in double precision. In Table 1.1, we show such errors.

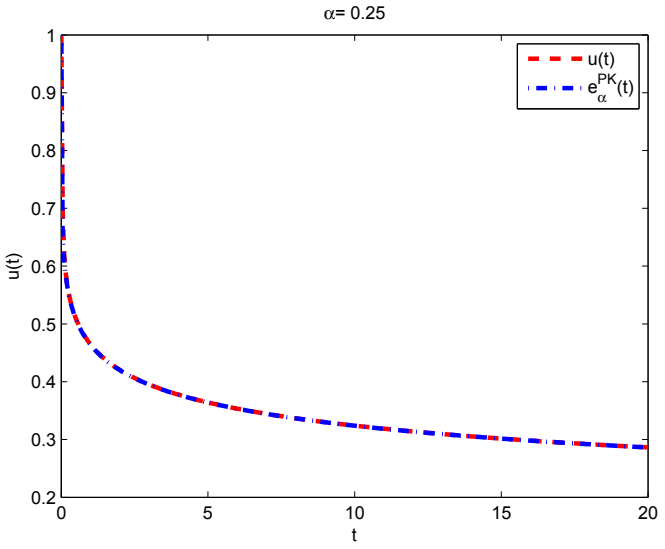

FIGURE 1.15: $u(t)$ is the numerical solution of the fODE in (A), while $e_\alpha^{PK}(t)$ is computed by the MATLAB code due to Podlubny and Kacenak, for $\alpha = 0.25$.

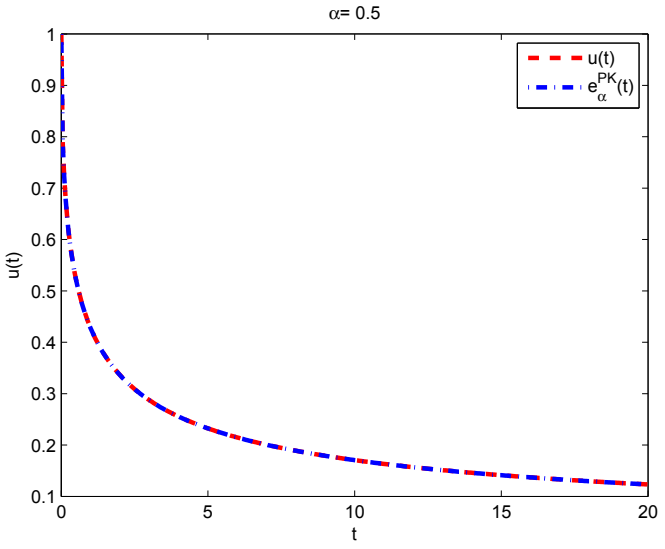

FIGURE 1.16: $u(t)$ is the numerical solution of the fODE in (A), while $e_\alpha^{PK}(t)$ is computed by the MATLAB code due to Podlubny and Kacenak, for $\alpha = 0.50$.

In Tables 1.2–1.17, we compare the CPU times spent by the pc and the apc algorithm, with those spent by the PK (Podlubny-Kacenak) MATLAB code, as well as by the *Mathematica* code (labeled by M). This is done for several values of the integration step size, h, and of the parameter α, at some times.

The results provided by the MATLAB code were obtained *always* requiring a 10^{-5} accuracy, while we chose step sizes (or initial step sizes, when using the apc algorithm) so to keep the discrepancy with the results of the PK algorithm within the same accuracy.

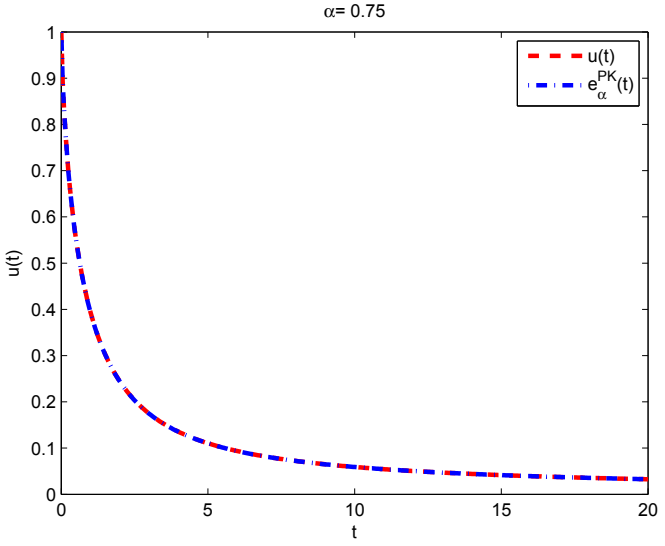

FIGURE 1.17: $u(t)$ is the numerical solution of the fODE in (A), while $e_\alpha^{PK}(t)$ is computed by the MATLAB code due to Podlubny and Kacenak, for $\alpha = 0.75$.

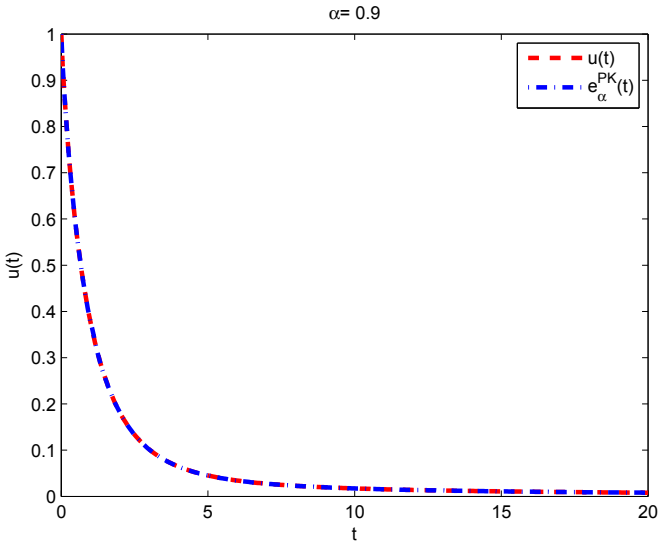

FIGURE 1.18: $u(t)$ is the numerical solution of the fODE in (A), while $e_\alpha^{PK}(t)$ is computed by the MATLAB code due to Podlubny and Kacenak, for $\alpha = 0.90$.

It turns out that the pc algorithm performs always better than the *Mathematica* code, for every value of α and time, while it performs better than the PK method with some limitations on the values of α and t.

In Table 1.2, the pc method outperforms the other two, when α is not too small and t is not too large. Note that CT_{PK}/CT_{pc} is about between 3.5 and 4.5. in Table 1.3, it is shown that at larger times the PK method wins for α larger than 0.5. This is due

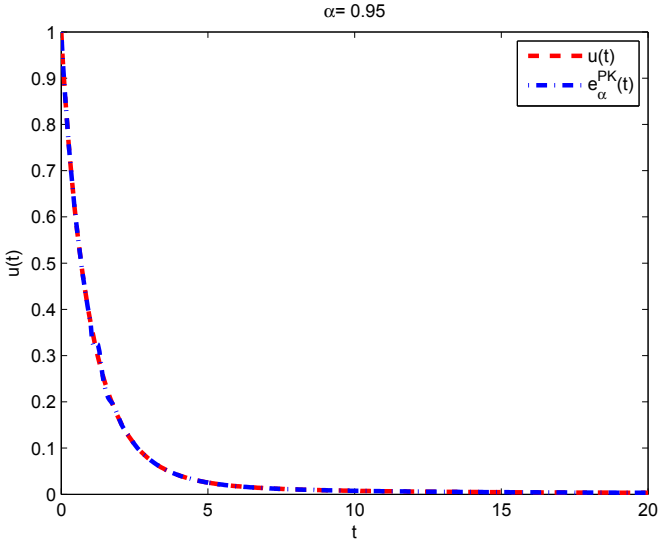

FIGURE 1.19: $u(t)$ is the numerical solution of the fODE in (A), while $e_\alpha^{PK}(t)$ is computed by the MATLAB code due to Podlubny and Kacenak, for $\alpha = 0.95$.

h	$\varepsilon^{pc}(20)$	$\varepsilon^{apc}(20)$	$\varepsilon^{apc}(100)$
0.05	$2.7210 \cdot 10^{-9}$	$6.6751 \cdot 10^{-10}$	$5.1283 \cdot 10^{-7}$
0.01	$2.6820 \cdot 10^{-9}$	$6.5623 \cdot 10^{-10}$	$5.0666 \cdot 10^{-7}$
0.005	$2.3950 \cdot 10^{-9}$	$6.4983 \cdot 10^{-10}$	$5.0189 \cdot 10^{-7}$

TABLE 1.1: Numerical errors, in the infinity norm, $\varepsilon^\dagger(T) := \max_{0\le t\le T} \left|e_\alpha^\dagger(t) - e_\alpha(t)\right|$, where the superscript $\dagger$ stands for *pc* or *apc* and $e_\alpha^\dagger(t)$ is the solution of equation (A) computed by the method denoted by $\dagger$. We chose $\alpha = 1/2$ so that $e_{1/2}(t) \equiv E_{1/2}(-t^{1/2}) = e^t \operatorname{erfc}(t^{1/2})$ (computed in double precision), and several values of h.

likely to a larger cumulated error in the integration of the fODE up to large time, which suggests that an adaptive strategy may resolve the issue,

In Tables 1.4–1.11, we chose more values of α, in the same range as above. In Tables 1.12-1.15, we also compare the CPU time spent using our apc algorithm. Clearly, such a method outperforms the PK method, significantly, for all values of $\alpha \in (0.25, 0.99)$ and even at times as large as $t = 500$.

In Tables 1.16, 1.17, the CPU times spent by pc, apc, PK, and M methods are shown, even for small values of α. When α is small, $e_\alpha(t)$ tends rapidly to have a very smooth behavior, but it becomes very steepen near $t = 0$. Therefore, the pc method, with a constant step size, h, requires a small value of h (see Table 1.10). A clear advantage is achieved adopting an adaptive strategy (see Table 1.11).

In Tables 1.18, 1.19, the memory (RAM) required by all methods is computed for several values of α and $t = 20$ or $t = 50$, respectively. It seems that the apc algorithm always

needs *significantly* less memory than the others, while the pc algorithm may need more memory than the PK method when α is very close to 1.

α	CT_{pc}	CT_{PK}	CT_M
0.25	0.4536	2.2464	0.6300
0.50	0.4212	2.1060	1.2950
0.60	0.4212	2.1528	1.8311
0.65	0.4212	2.1684	2.1641
0.70	0.4212	2.1060	2.5231
0.75	0.4212	2.1528	2.6271
0.80	0.4212	2.1372	2.7231
0.85	0.4056	2.1372	2.7691
0.90	0.4212	2.0124	2.8381
0.95	0.4212	1.7160	2.9581
0.99	0.4368	1.5132	3.4801

TABLE 1.2: CPU time, in seconds, spent to compute $e_\alpha(t)$ with the *predictor-corrector* (pc) method to integrate equation (A), the Podlubny-Kacenak (PK) *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 20$, step-size $h = 0.05$, and several values of $\alpha \in [0.25, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M
0.25	6.7656	38.4219	12.9012
0.5	7.3906	7.0000	12.9012
0.6	6.8750	4.0781	13.5720
0.65	6.8125	3.2813	14.0868
0.7	6.6875	2.8594	14.3208
0.75	6.7969	2.5313	12.6048
0.8	6.7031	2.3438	14.4144
0.85	6.7500	2.2188	18.7356
0.9	6.9219	2.0938	19.6788
0.95	7.2813	1.9844	22.5996
0.99	6.8906	1.8438	22.8696

TABLE 1.3: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to integrate equation (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 1000$, step-size $h = 0.05$, and several values of $\alpha \in [0.25, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M	α	CT_{pc}	CT_{PK}	CT_M	α	CT_{pc}	CT_{PK}	CT_M
0.25	2.1996	9.7345	3.1721	0.5	2.0592	10.0465	5.9003	0.75	2.0124	9.6721	12.6787
0.275	1.9656	9.5005	3.2921	0.525	2.1528	10.1089	7.0964	0.775	1.9500	9.6721	13.4957
0.3	1.9188	9.8125	3.3551	0.55	2.1372	10.0777	7.4664	0.8	1.9032	9.7033	13.6067
0.325	1.9032	9.6877	3.8442	0.575	2.3712	10.6081	8.5274	0.825	1.9500	9.5941	13.8967
0.35	2.0124	9.5005	3.9362	0.6	2.0904	10.3117	8.8475	0.85	2.3556	9.8593	14.5608
0.375	1.8876	9.5473	4.3372	0.625	2.0436	9.8437	9.7615	0.875	1.9344	9.5629	13.9978
0.4	2.3088	11.1541	4.6392	0.65	2.0592	10.3429	10.8666	0.9	1.9500	8.2993	14.1178
0.425	2.1372	10.3897	5.1662	0.675	2.3556	10.1557	11.6736	0.925	1.9501	8.2993	14.4088
0.45	2.3556	10.2181	5.8443	0.7	1.9188	9.5629	12.7507	0.9500	1.9812	7.7532	14.6948
0.475	2.1372	9.8749	6.2123	0.725	1.9032	9.5473	12.9007	0.99	2.1216	7.0202	17.0029

TABLE 1.4: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 20$, step-size $h = 0.01$, and several values of $\alpha \in [0.25, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M
0.25	24.5938	89.1094	33.3996
0.275	18.9375	107.5	34.6788
0.3	23.25	125.2344	35.9268
0.325	22.2656	127.9375	37.8144
0.35	20.8281	104.0313	35.0688
0.375	21.9219	93.2344	36.6288
0.4	19.2969	88.7188	30.9192
0.425	18.5	62.9375	39.0624
0.45	18.2188	52.6875	37.2528
0.475	18.1094	36.4688	37.9080

TABLE 1.5: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 500$, step-size $h = 0.01$, and several values of $\alpha \in [0.25, 0.475]$.

α	CT_{pc}	CT_{PK}	CT_M
0.5	19.1094	28.5469	28.1580
0.525	18.3438	23.25	39.1248
0.55	18.8594	22.1563	39.5928
0.575	18.2656	18.375	39.8112
0.6	18.4688	16.5156	34.6632
0.625	18.3125	14.6563	39.2496
0.65	18.6719	13.1563	42.1668
0.675	20.125	11.9375	43.6800
0.7	18.5156	10.875	44.5692
0.725	19.9063	12.2656	48.7188

TABLE 1.6: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 500$, step-size $h = 0.01$, and several values of $\alpha \in [0.5, 0.725]$.

α	CT_{pc}	CT_{PK}	CT_M
0.75	18.6875	9.0938	39.9984
0.775	18.3281	8.8594	51.2148
0.8	19.9688	8.75	47.3772
0.825	19.5313	8.2969	56.5344
0.85	19.8594	7.7344	61.5141
0.875	18.9063	7.3281	58.3909
0.9	20.1563	7.3125	62.9305
0.925	20.1875	11.3281	67.9537
0.95	19.0781	6.8281	72.3997
0.99	19.7813	6.2031	81.8845

TABLE 1.7: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 500$, step-size $h = 0.01$, and several values of $\alpha \in [0.75, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M
0.25	0.4536	2.2464	0.6300
0.50	0.4212	2.1060	1.2950
0.60	0.4212	2.1528	1.8311
0.65	0.4212	2.1684	2.1641
0.70	0.4212	2.1060	2.5231
0.75	0.4212	2.1528	2.6271
0.80	0.4212	2.1372	2.7231
0.85	0.4056	2.1372	2.7691
0.90	0.4212	2.0124	2.8381
0.95	0.4212	1.7160	2.9581
0.99	0.4368	1.5132	3.4801

TABLE 1.8: CPU times, in seconds, to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine, (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 20$, step-size $h = 0.005$, and several values of $\alpha \in [0.25, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M
0.25	24.5938	89.1094	55.3582
0.275	18.9375	107.5	57.8141
0.3	23.25	125.2344	52.5701
0.325	22.2656	127.9375	62.8804
0.35	20.8281	104.0313	62.6102
0.375	21.9219	93.2344	57.2201
0.4	19.2969	88.7188	56.5101
0.425	18.5	62.9375	74.0241
0.45	18.2188	52.6875	77.6622
0.475	18.1094	36.4688	83.7943

TABLE 1.9: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 50$, step-size $h = 0.001$, and several values of $\alpha \in [0.25, 0.475]$.

α	CT_{pc}	CT_{PK}	CT_M
0.5	19.1094	28.5469	72.8221
0.525	18.3438	23.25	95.5402
0.55	18.8594	22.1563	102.0492
0.575	18.2656	18.375	110.0062
0.6	18.4688	16.5156	105.3102
0.625	18.3125	14.6563	116.0803
0.65	18.6719	13.1563	132.8942
0.675	20.125	11.9375	142.1564
0.7	18.5156	10.875	145.8522
0.725	19.9063	12.2656	188.9640

TABLE 1.10: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 50$, step-size $h = 0.001$, and several values of $\alpha \in [0.5, 0.725]$.

α	CT_{pc}	CT_{PK}	CT_M
0.75	18.6875	9.0938	199.7044
0.775	18.3281	8.8594	236.1165
0.8	19.9688	8.75	269.4684
0.825	19.5313	8.2969	288.3674
0.85	19.8594	7.7344	1142.2454
0.875	18.9063	7.3281	332.3783
0.9	20.1563	7.3125	374.6992
0.925	20.1875	11.3281	349.9398
0.95	19.0781	6.8281	352.8882
0.99	19.7813	6.2031	411.0209

TABLE 1.11: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the MATLAB routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively), for $t = 50$, step-size $h = 0.001$, and several values of $\alpha \in [0.75, 0.99]$.

α	CT_{apc}	CT_{PK}	CT_M
0.25	0.024994	6.7656	28.3452
0.275	0.023783	6.9375	29.4840
0.3	0.026653	6.8125	28.3452
0.325	0.026868	6.8906	30.6384
0.35	0.028212	6.7969	31.0908
0.375	0.029891	6.7813	28.8288
0.4	0.030899	6.8125	26.4576
0.425	0.032617	6.7813	33.9768
0.45	0.032914	6.7969	34.1172
0.475	0.033921	6.375	35.0688

TABLE 1.12: CPU times, in seconds, spent to compute $e_\alpha(t)$ with our *adaptive predictor-corrector* (apc) method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{apc}, CT_{PK}, and CT_M, respectively), for $t = 200$, initial step-size $h = 0.05$, and several values of $\alpha \in [0.25, 0.475]$.

α	CT_{apc}	CT_{PK}	CT_M
0.5	0.034191	4.8594	26.4888
0.525	0.035265	3.9688	37.1124
0.55	0.036842	3.2656	38.9688
0.575	0.035936	2.75	47.754
0.6	0.036354	2.6094	49.4718
0.625	0.035936	2.2656	48.0817
0.65	0.035936	2.4375	53.4410
0.675	0.036354	1.7188	49.8268
0.7	0.035187	1.5469	49.3858
0.725	0.035675	1.3906	57.4428

TABLE 1.13: CPU times, in seconds, spent to compute $e_\alpha(t)$ with our *adaptive predictor-corrector* (apc) method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{apc}, CT_{PK}, and CT_M, respectively), for $t = 200$, initial step-size $h = 0.05$, and several values of $\alpha \in [0.5, 0.725]$.

α	CT_{apc}	CT_{PK}	CT_M
0.75	0.034593	1.2656	72.1831
0.775	0.034257	1.1563	82.9037
0.8	0.034366	1.1406	67.93488
0.825	0.032914	1.0469	78.40548
0.85	0.032242	0.96875	88.47792
0.875	0.031234	0.92188	103.7479
0.9	0.030563	0.875	109.1712
0.925	0.029891	0.8125	119.2497
0.95	0.029891	0.76563	132.8766
0.99	0.02722	0.85938	152.7927

TABLE 1.14: CPU times, in seconds, spent to compute $e_\alpha(t)$ with our *adaptive predictor-corrector* (apc) method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{apc}, CT_{PK}, and CT_M, respectively), for $t = 200$, initial step-size $h = 0.05$, and several values of $\alpha \in [0.75, 0.99]$.

α	CT_{apc}	CT_{PK}	CT_M	α	CT_{apc}	CT_{PK}	CT_M	α	CT_{apc}	CT_{PK}	CT_M
0.25	0.024601	8.5625	3.5632	0.5	0.029357	2.6406	2.8236	0.75	0.025073	0.84375	4.3112
0.275	0.022399	8.5	3.6282	0.525	0.029085	2.2031	3.9468	0.775	0.024404	0.78125	5.8613
0.3	0.023736	8.4375	3.6022	0.55	0.029085	1.8438	4.0404	0.8	0.024058	0.78125	5.2022
0.325	0.024739	8.4063	4.2782	0.575	0.02875	1.5781	4.2120	0.825	0.022945	0.73438	6.1323
0.35	0.026076	8.4531	4.4932	0.6	0.028416	1.5	3.6816	0.85	0.021933	0.6875	6.4393
0.375	0.026995	8.4219	3.5412	0.625	0.028082	1.3281	4.2432	0.875	0.021308	0.65625	6.4583
0.4	0.028266	8.4531	3.0576	0.65	0.027748	1.1719	5.0549	0.9	0.020058	0.625	7.3984
0.425	0.028082	5.9688	3.5880	0.675	0.027332	1.2031	4.7922	0.925	0.018877	0.59375	8.0824
0.45	0.029557	4.4063	3.6660	0.7	0.02641	0.96875	4.8472	0.95	0.018053	0.5625	8.0824
0.475	0.028482	3.3906	3.8220	0.725	0.025742	0.90625	5.0542	0.99	0.016559	0.54688	8.7645

TABLE 1.15: CPU times, in seconds, spent to compute $e_\alpha(t)$ with our *adaptive predictor-corrector* (apc) method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{apc}, CT_{PK}, and CT_M, respectively), for $T = 500$, step-size $h = 0.1$, and several values of $\alpha \in [0.25, 0.99]$.

α	CT_{pc}	CT_{PK}	CT_M
0.01	4.8828	12.5737	4.95400
0.025	3.7908	11.9029	4.6200
0.05	3.8223	12.4489	4.7620
0.1	3.6666	12.7765	4.6420
0.2	3.9312	12.5113	5.3720
0.3	4.0092	12.6361	6.0020
0.4	3.7128	12.3865	9.1880
0.5	3.8844	12.1525	13.1920
0.6	3.8688	12.1213	18.6180
0.7	3.822	12.1993	5.5400
0.8	3.7752	11.9809	27.8501
0.9	4.3992	12.0277	28.4761
0.95	4.0092	12.0121	28.2961
0.975	3.6816	11.9653	29.1401
0.99	3.8064	12.0277	30.6440

TABLE 1.16: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method to solve (A), the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{PK}, and CT_M, respectively, for $t = 20$, step-size $h = 0.005$, and several values of $\alpha \in [0.01, 0.99]$.

α	CT_{pc}	CT_{apc}	CT_{PK}	CT_M
0.01	8.6563	0.0164	43.7969	15.8496
0.025	8.0781	0.0091	42.7500	15.8340
0.05	8.2031	0.0074	42.8594	15.5532
0.1	8.2656	0.0116	42.5938	18.1370
0.2	8.2813	0.0221	42.5938	18.7040
0.3	9.0156	0.0329	43.2188	18.9180
0.4	8.2031	0.0449	43.8125	22.7463
0.5	8.6719	0.0581	42.7344	26.6845
0.6	8.0156	0.0731	34.5000	33.4169
0.7	8.2500	0.0849	18.0469	41.5683
0.8	8.2188	0.1109	13.9375	65.8827
0.9	8.3281	0.1194	10.3125	78.7445
0.95	8.8125	0.1326	9.2500	85.0825
0.975	8.6250	0.1352	8.2188	99.8176
0.99	8.8594	0.1428	7.6094	106.7241

TABLE 1.17: CPU times, in seconds, spent to compute $e_\alpha(t)$ with the pc method, or with our apc (*adaptive predictor-corrector*) method to solve (A) , the *MATLAB* routine, and the *Mathematica* routine (CT_{pc}, CT_{apc}, CT_{PK}, and CT_M, respectively), for $t = 100$, initial step-size $h = 0.005$, and several values of $\alpha \in [0.01, 0.99]$.

α	RAM_{pc}	RAM_{PK}	RAM_M
0.01	4309	5484	10776
0.025	10202	12985	25512
0.05	19947	25388	49880
0.1	45056	57344	112664
0.2	111640	142087	279160
0.3	189422	241082	473656
0.4	218574	278185	546552
0.5	194835	247972	487192
0.6	276667	352122	691816
0.7	435008	553646	1087752
0.8	438456	558036	1096376
0.9	438488	558076	1096456
0.95	438488	558076	1096456
0.975	438642	558272	1096840
0.99	438888	558585	1097456

TABLE 1.18: RAM, in MegaBytes, required to compute $e_\alpha(t)$ with the pc method to solve equation (A), the *MATLAB* routine, and *Mathematica*'s routine (RAM_{pc}, RAM_{PK}, and RAM_M, respectively), for $t = 20$, step-size $h = 0.005$, and several values of $\alpha \in [0.01, 0.99]$.

α	RAM_{pc}	RAM_{apc}	RAM_{PK}	RAM_M
0.01	38191	725	193231	69928
0.025	35640	404	188613	69859
0.05	36192	320	189095	68620
0.1	36467	518	187793	80020
0.2	36537	977	187938	82522
0.3	39776	1455	190681	83466
0.4	36192	1987	193300	100356
0.5	38260	2569	188544	117732
0.6	35364	3220	152214	147435
0.7	36399	3744	80064	183379
0.8	36261	1897	61492	290674
0.9	36743	5266	45498	347420
0.95	38880	5854	40811	374383
0.975	38053	5964	36261	440395
0.99	39087	6304	33572	470866

TABLE 1.19: RAM, in MegaBytes, required to compute $e_\alpha(t)$ with the pc method or with our apc (*adaptive predictor-corrector*) method to solve equation (A), the *MATLAB* routine, and *Mathematica*'s routine (RAM_{pc}, RAM_{apc}, RAM_{PK}, and RAM_M, respectively), for $t = 50$, step-size $h = 0.005$, and several values of $\alpha \in [0.01, 0.99]$.

Chapter 2

Justifying fractional diffusion modeling through realistic data

In this chapter, we give some justification of the widespread use of fPDE to model a number of physical phenomena, such as, for instance, fluid flow of tracers through porous media.

Several laboratory experiments have been performed concerning one-dimensional flow in order to determine the flow rate through a uniformly packed column filled in by certain porous media. This measurements were done using permeability meter. The experimental setup was designed aiming at determining the memory properties of this kind of dynamical behavior, which is well reproduced through a suitable fPDE modeling. Then, the parameters estimated by using the equation which describes the instantaneous flux rate through a given porous medium layer when a constant pressure is applied on the medium lower boundary. One can use water as fluid, and a variety of porous media.

We made a numerical approximation of experimental data to describe anomalous diffusion through porous media, and compared our numerical results with those obtained by *laboratory tests*, and observed the matching for certain values of the order of fractional derivatives and initial conditions,

2.1 Problem statement

Consider the one-dimensional fractional diffusion equation fractional in both, space and time,

$$ {}_{C}D_{t}^{\gamma}u = K_{RL}D_{x}^{\alpha}u + f(x,t), \tag{2.1} $$

where $u = u(x,t)$ represents, e.g., pressure, $K > 0$, $0 < \gamma < 1$, $1 < \alpha < 2$, $x \in [a,b]$, $t \in [0,T]$, for some a, b, $T > 0$, along with the boundary conditions $u(a,t) = u_1, u(b,t) = u_2$ and the initial condition $u(x,0) = u_0(x)$, for some u_1, u_2, and $u_0(x)$, $f(x;t)$ being some external source. We solved numerically such problem using finite difference Grünwald-Letnikov (GL) approximations. We started considering a mesh in the space-time region $[a,b] \times [0,T]$, where numerical estimates, say $U_i^{(n)}$, for the solution $u(x_i,t^n) = u_i^{(n)}, where (x_i,t^n)$ denote the coordinates of the (i,n) node of the mesh. We then replaced the continuous fractional derivatives operators, in both, time and space, in equation (2.1), with their GL approximation. Thus, we obtain a difference equation whose solution leads to the finite difference estimate of the exact solution, $u(x,t)$, at the mesh points. For a given fractional differential, one can consider several possible forms of difference operators, and hence several different finite difference methods to solve the ensuing finite difference equation.

We should recall, at this point, that many authors have considered similar problems, but in the most often adopting a *fractional* order only in either time or space, see Yuste et al. [54, 146–148], e.g.

Here we assume a uniform spatial step-size, $\Delta x := x_{i+1} - x_i$, and consider the space fractional differential operator in the sense of Riemann-Liouville, that we will approximate by means of "GL differences" while we will use fractional time derivatives in the sense of Caputo, using again GL differences with a constant time step-size $\Delta t := t^{n+1} - t^n$.

We thus obtain a finite difference algorithm ending up with a linear system like

$$U_i^{(n+1)} = (A - M)U_i^{(n)},$$

where A and M represent the discretized time and space operators, respectively, obtained through the GL discretization, see the Introduction, and [113].

A code was then written to compute the parameters governing the "memory effects" along with the fractional order of the derivatives, by minimizing the L^2 and the L^∞ norms (with respect to space) of the discrepancies between the experimental the numerically computed values obtained by solving (numerically) the aforementioned anomalous diffusion equation. These value are those of the pressure at the upper boundary of the domain occupied by the porous medium.

Recently, in [50], experimental results have been obtained to estimate the physical parameters governing the dynamical behavior of the flow of tracers through a certain porous medium, in particular about the aforementioned memory effects, described by parameters such as, for instance, the order of the fractional derivative with respect to time.

Memory effects may also due to the size of the tank containing the porous medium,the particle size distribution, the stability of the initial particle distribution, etc.

The function to minimize is then the error, the discrepancy, between the numerically computed value, $U^{(n)}$, and the pressure obtained by laboratory experiments say u, that is $\|u - U^{(n)}\|_k, k = 2$ or $k = \infty$,

2.2 Discretization of partial derivatives in time and space

The simplest implicit discretization scheme (the stencil) for classical diffusion equations is shown on Fig. 2.1, where two nodes are used to approximate the time derivative, and nodes are used for a symmetric approximation of the the space derivative. Therefore, the stencil in Fig. 2.1 involves only two time levels. When considering fractional-order time derivatives, all time levels should be involved, starting from the very beginning. This is shown in Fig. 2.2 (where five time levels are shown).

Similarly, if, in addition to a fractional time derivative, some fractional-order space derivatives are involved, all spatial nodes, from the leftmost to the rightmost, at all time levels, should be used. This more general case is depicted in Fig. 2.3.

Let us consider the nodes $(ih; j\tau), j = 0, 1, 2, ..., n$, corresponding to all time levels at i-th space discretization node; here h is the space discretization step, and then the approximation through the B matrix introduced above

$$\begin{bmatrix} u_{i,N}^{(\gamma)} & u_{i,N-1}^{(\gamma)} & ...u_{i,2}^{(\gamma)} & u_{i,1}^{(\gamma)} & u_{i,0}^{(\gamma)} \end{bmatrix}^T = B_N^{(\gamma)} \begin{bmatrix} u_{i,N} & u_{i,N-1} & ...u_{i,2} & u_{i,1} & u_{i,0} \end{bmatrix}^T \tag{2.2}$$

In order to obtain a simultaneous approximation of the time derivative, and then of the space derivatives of $u(x;t)$ of order γ, in time for instance,at each grid point shown in Fig. 1.1, we need to arrange all function values, u_{ij}, at the discretization nodes, in a column vector,

$$\begin{aligned} u_{NM} = \Big[& u_{M,N} \quad u_{M-1,N} \quad \cdots \quad u_{1,N} \quad u_{0,N} \\ & u_{M,N-1} \quad u_{M-1,N-1} \quad \cdots \quad u_{1,N-1} \quad u_{0,N-1} \\ & \cdots\cdots \\ & u_{M,1} \quad u_{M-1,1} \quad \cdots \quad u_{1,1} \quad u_{0,1} \\ & u_{M,0} \quad u_{M-1,0} \quad \cdots \quad u_{1,0} \quad u_{0,0} \Big]^T \end{aligned} \tag{2.3}$$

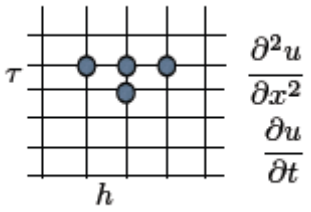

FIGURE 2.1: A stencil for integer-order derivatives.

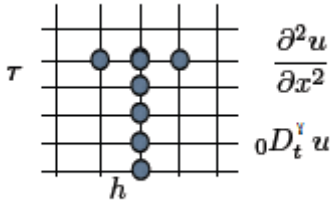

FIGURE 2.2: A stencil in case of fractional time derivative.

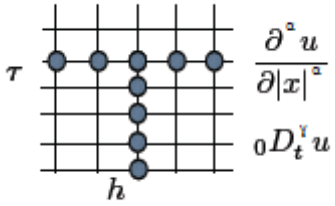

FIGURE 2.3: A stencil in case of fractional time and spatial derivatives.

So, as shown in (2.4), we can put all the numerical values of the solution in a matrix, which contains the numerical results for every discretized value of time and space. The matrix before represent the iteration matrix which give the discretization $U_t^{(\gamma)}$ of the fractional partial derivative of order γ with respect to time, it can be written as the Kronecker product of the matrix $B_N^{(\gamma)}$, corresponding to the fractional ordinary derivative of order γ, and the identity matrix, I_M. Recall that here N is the number of time steps and M is the number of space discretization nodes,

$$T_{MN}^{(\gamma)} = B_N^{(\gamma)} \otimes I_M, \tag{2.4}$$

this stencil is depicted in Fig. 2.5.

Similarly, the matrix governing the transformation of the vector U into the vector $U_x^{(\alpha)}$ of the fractional partial space derivative of order α can be obtained as the Kronecker

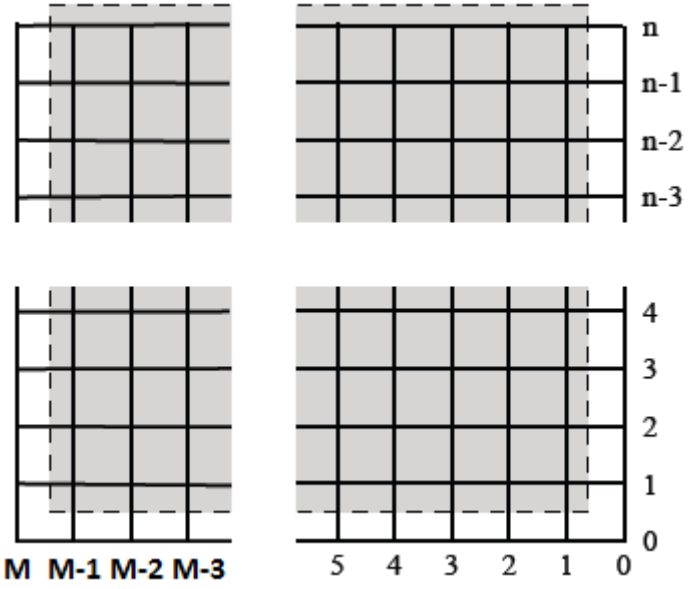

FIGURE 2.4: Nodes and their right-to-left, and bottom-to-top numbering.

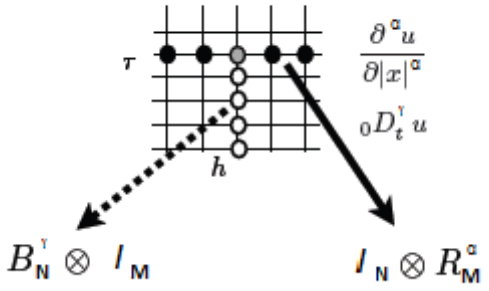

FIGURE 2.5: Discretization nodes of partial derivatives.

product of the unit matrix I_N, where N is the number of the space nodes in the mesh. The matrix $R_N^{(\alpha)}$, which corresponds to a symmetric ordinary derivative of order α [107, 108], where N is the number of time steps is

$$S_{MN}^{(\gamma)} = I_N \otimes R_M^{(\alpha)}. \tag{2.5}$$

All this is illustrated in Fig.2.5, where the nodes in black and gray (corresponding to all discretization nodes, from the leftmost to the rightmost ones) are used to approximate the symmetric fractional-order derivative at the node in gray.

With these approximations for the fractional partial derivatives made with respect to both variables, we can discretize at once the fractional diffusion equation in (4.1), (see

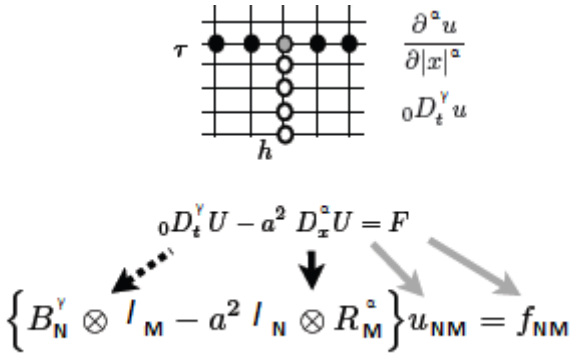

FIGURE 2.6: Discretization nodes of partial derivatives and of the equation.

Fig. 2.6), that is, writing $\chi := a$,

$$ {}_0^C D_t^\gamma - \chi \frac{\partial^\alpha}{\partial |x|^\alpha} = f(x;t), \tag{2.6} $$

we obtain

$$ \left\{ B_M^{(\gamma)} \otimes I_N - \chi I_M \otimes R_N^{(\alpha)} \right\} u_{MN} = f_{MN}. \tag{2.7} $$

Here, the system's matrix has the structure shown in Fig. 2.7.

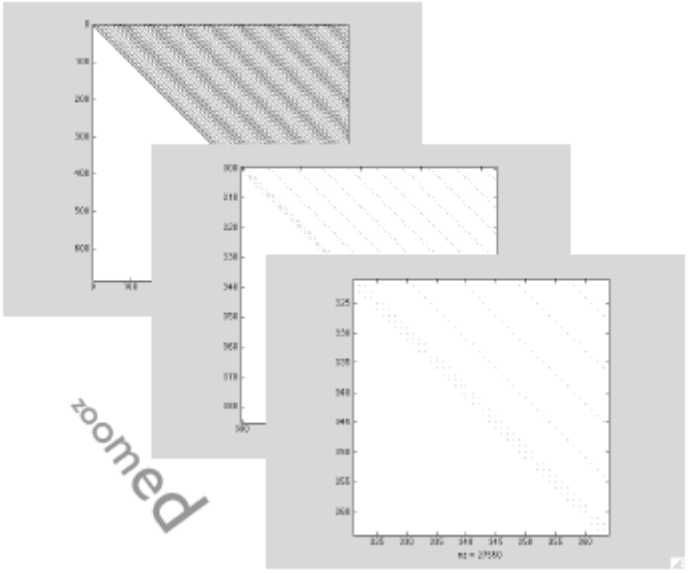

FIGURE 2.7: The structure of the matrix of the resulting algebraic system.

2.3 Numerical examples

Example 1. Consider the one-dimensional fractional diffusion equation in space and time

$$ {}_C D_x^{\gamma} u = K_{GL} D_x^{\alpha} u, \quad (x,t) \in (0,1) \times (0,T], \tag{2.8} $$

satisfied by $u = u(x,t)$, with $K > 0$, $0 < \gamma < 1$, $1 < \alpha < 2$, under the boundary conditions $u(0,t) = u(1,t) = 0$, and the initial condition $u_0(x) = \frac{2}{5}x(1-x)$; u represent the solution provided by the experimental data.

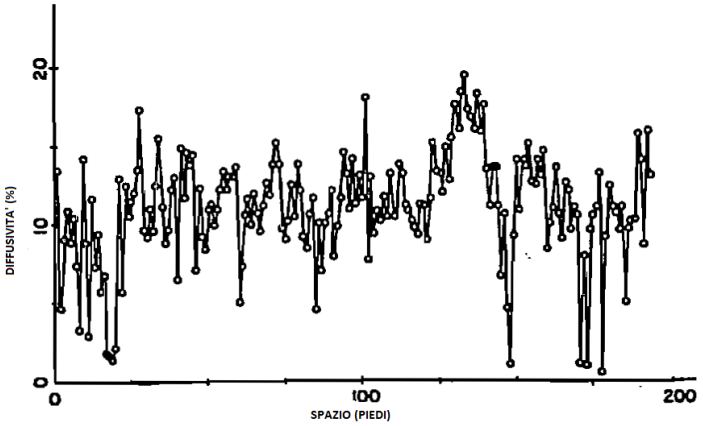

FIGURE 2.8: The data.

Given the data in Fig. 2.1, i.e., the values of $u(x,t)$ obtained experimentally, we performed a fitting of them through equation (2.8), for several values of the parameters involved by the model. All data were normalized in the space range $[0,1]$, by dividing the space values for bigger one of them.

In the next tables, are results of discrepancy between the numerical solution and the laboratory data, applied to the equation with varying values of γ ed α, we computed also the time T when the minimum discrepancy between the numerical solution the solution provided by the experimental data is attained.

In the following Tables, N_x is the number of points of the space-grid, and N_t is the number of time iterations, T_2 and T_∞ are the times required to attain the aforementioned minimum discrepancy between u and $U^{n()}$, computed in l^2 and in l^∞ norm. We considered a few values for γ and α, and chose $K = 1$.

Inspecting all the previous tables, we can see that, in general, the best results are achieved for $\alpha = 1.9$, and, more precisely, correspondingly to the pair $(\gamma, \alpha) = (0.7, 1.9)$. When, roughly, $\gamma = 0.8$ or 0.9, the best results are obtained in the final time. This suggests that

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.1, 1.1)$	21	37	1	1	10.310497	2.747557
$(0.1, 1.2)$	21	37	1	1	8.520539	2.309313
$(0.1, 1.3)$	21	37	1	1	6.989970	1.920353
$(0.1, 1.4)$	21	37	1	1	5.700736	1.583714
$(0.1, 1.5)$	21	37	1	1	4.626083	1.297250
$(0.1, 1.6)$	21	37	1	1	3.736588	1.056232
$(0.1, 1.7)$	21	37	1	1	3.003789	0.854992
$(0.1, 1.8)$	21	37	1	1	2.402051	0.687873
$(0.1, 1.9)$	21	37	1	1	1.909298	0.549699

TABLE 2.1: $\gamma = 0.1$ and $\alpha = 1.1, ..., 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.2, 1.1)$	21	37	1	1	4.974169	1.289789
$(0.2, 1.2)$	21	37	1	1	4.407628	1.147597
$(0.2, 1.3)$	21	37	1	1	3.854292	1.008202
$(0.2, 1.4)$	21	37	1	1	3.327019	0.875724
$(0.2, 1.5)$	21	37	1	1	2.835534	0.752893
$(0.2, 1.6)$	21	37	1	1	2.386288	0.641281
$(0.2, 1.7)$	21	37	1	1	1.982675	0.541553
$(0.2, 1.8)$	21	37	1	1	1.625573	0.453704
$(0.2, 1.9)$	21	37	1	1	1.314016	0.377275

TABLE 2.2: $\gamma = 0.2$ and $\alpha = 1.1, ..., 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.3, 1.1)$	21	37	1	1	1.972317	0.570971
$(0.3, 1.2)$	21	37	1	1	1.804314	0.517336
$(0.3, 1.3)$	21	37	1	1	1.628181	0.462917
$(0.3, 1.4)$	21	37	1	1	1.447909	0.420117
$(0.3, 1.5)$	21	37	1	1	1.267220	0.375917
$(0.3, 1.6)$	21	37	1	1	1.089604	0.331684
$(0.3, 1.7)$	21	37	1	1	0.918333	0.288565
$(0.3, 1.8)$	21	37	1	1	0.756458	0.247465
$(0.3, 1.9)$	21	37	1	1	0.606827	0.209066

TABLE 2.3: $\gamma = 0.3$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.4, 1.1)$	21	37	1	1	0.602885	0.277140
$(0.4, 1.2)$	21	37	1	1	0.553072	0.252688
$(0.4, 1.3)$	21	37	1	1	0.499804	0.228431
$(0.4, 1.4)$	21	37	1	1	0.445729	0.205266
$(0.4, 1.5)$	21	37	1	1	0.394087	0.183812
$(0.4, 1.6)$	21	37	1	1	0.348678	0.164408
$(0.4, 1.7)$	21	37	1	1	0.313716	0.147160
$(0.4, 1.8)$	21	37	1	1	0.293120	0.132018
$(0.4, 1.9)$	21	37	1	1	0.288937	0.118841

TABLE 2.4: $\gamma = 0.4$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.5, 1.1)$	21	37	2	2	0.505385	0.219545
$(0.5, 1.2)$	21	37	2	2	0.489804	0.203944
$(0.5, 1.3)$	21	37	3	2	0.457177	0.187583
$(0.5, 1.4)$	21	37	3	2	0.421077	0.176299
$(0.5, 1.5)$	21	37	3	3	0.387604	0.175057
$(0.5, 1.6)$	21	37	3	3	0.359919	0.157268
$(0.5, 1.7)$	21	37	4	3	0.321337	0.141324
$(0.5, 1.8)$	21	37	4	3	0.290557	0.135579
$(0.5, 1.9)$	21	37	5	4	0.261397	0.122400

TABLE 2.5: $\gamma = 0.5$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.6, 1.1)$	21	37	7	5	0.504675	0.204418
$(0.6, 1.2)$	21	37	7	5	0.483886	0.201709
$(0.6, 1.3)$	21	37	8	6	0.457695	0.189959
$(0.6, 1.4)$	21	37	8	6	0.427616	0.178937
$(0.6, 1.5)$	21	37	9	7	0.393911	0.165700
$(0.6, 1.6)$	21	37	10	8	0.359888	0.155612
$(0.6, 1.7)$	21	37	11	8	0.326056	0.142089
$(0.6, 1.8)$	21	37	12	9	0.293451	0.130169
$(0.6, 1.9)$	21	37	13	10	0.263494	0.120598

TABLE 2.6: $\gamma = 0.6$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.7, 1.1)$	21	37	16	12	0.508688	0.206358
$(0.7, 1.2)$	21	37	17	13	0.488374	0.201000
$(0.7, 1.3)$	21	37	18	13	0.463273	0.191483
$(0.7, 1.4)$	21	37	19	15	0.433532	0.180530
$(0.7, 1.5)$	21	37	20	16	0.400183	0.167638
$(0.7, 1.6)$	21	37	21	17	0.365007	0.154422
$(0.7, 1.7)$	21	37	23	18	0.330013	0.142221
$(0.7, 1.8)$	21	37	24	20	0.296473	0.131737
$(0.7, 1.9)$	21	37	26	22	0.266198	0.122152

TABLE 2.7: $\gamma = 0.7$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
$(0.8, 1.1)$	21	37	30	23	0.512595	0.207893
$(0.8, 1.2)$	21	37	32	24	0.492909	0.201485
$(0.8, 1.3)$	21	37	33	26	0.468352	0.190574
$(0.8, 1.4)$	21	37	35	28	0.439412	0.180699
$(0.8, 1.5)$	21	37	36	30	0.406605	0.168859
$(0.8, 1.6)$	21	37	36	32	0.375159	0.156266
$(0.8, 1.7)$	21	37	36	34	0.349118	0.143818
$(0.8, 1.8)$	21	37	36	36	0.331122	0.132073
$(0.8, 1.9)$	21	37	36	36	0.322888	0.134611

TABLE 2.8: $\gamma = 0.8$ and $\alpha = 1.1, \ldots, 1.9$.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	$\|u - U^{(n)}\|_\infty$
	$\|\|u^n - U^n\|\|_2$	$\|u^n - U^n\|_\infty$				
$(0.9, 1.1)$	21	37	36	36	0.589970	0.225874
$(0.9, 1.2)$	21	37	36	36	0.584715	0.226410
$(0.9, 1.3)$	21	37	36	36	0.578677	0.226932
$(0.9, 1.4)$	21	37	36	36	0.572355	0.227418
$(0.9, 1.5)$	21	37	36	36	0.566443	0.227847
$(0.9, 1.6)$	21	37	36	36	0.561712	0.228198
$(0.9, 1.7)$	21	37	36	36	0.558880	0.228465
$(0.9, 1.8)$	21	37	36	36	0.558494	0.228677
$(0.9, 1.9)$	21	37	36	36	0.560850	0.228916

TABLE 2.9: $\gamma = 0.9$ and $\alpha = 1.1, \ldots, 1.9$.

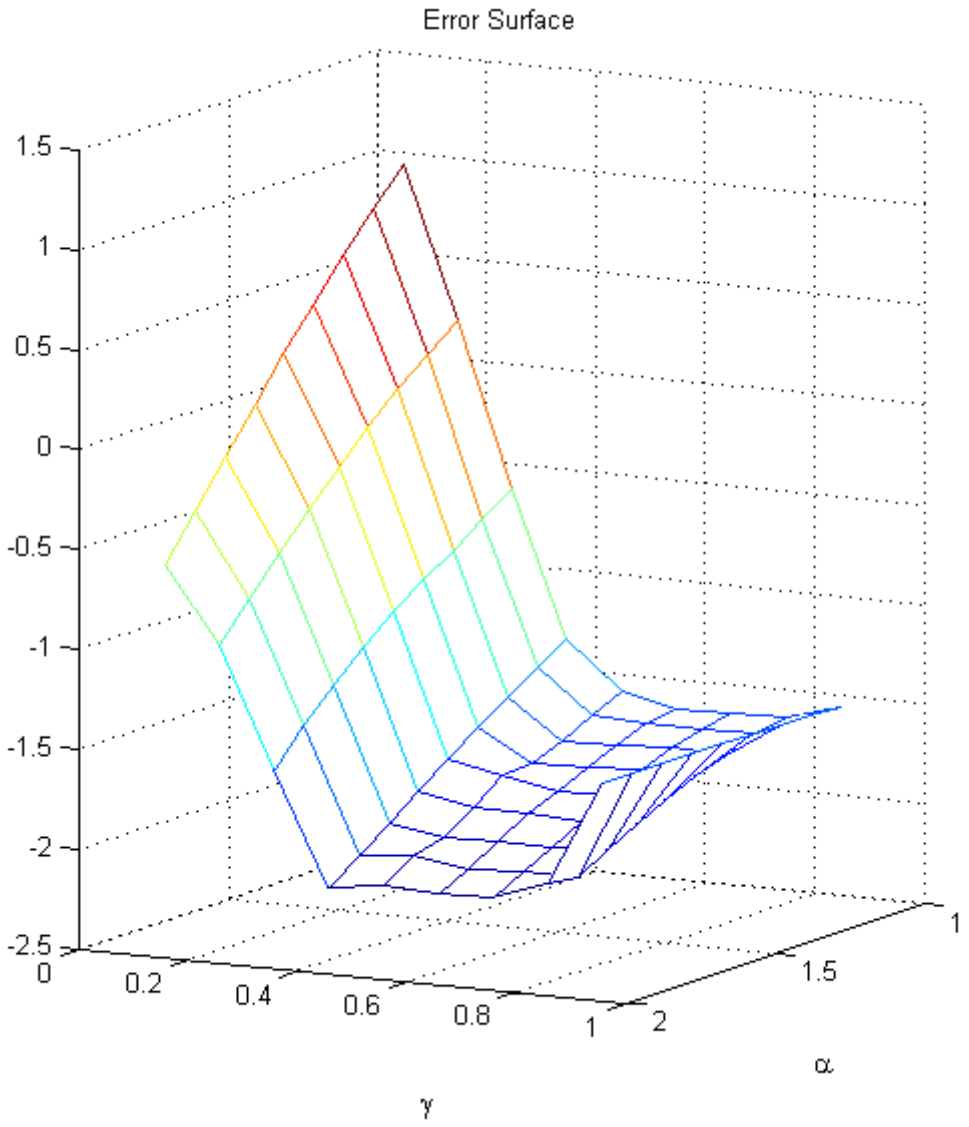

FIGURE 2.9: Error surface in log scale for $k = 1, N_t = 37$ and $N_x = 21$.

the error can be minimize by increasing the final time T. Using $N_t = 148$ iterations, we obtained for $(\alpha, \gamma) = (0.8, 1.9)$, $N_x = 21$, and $N_x = 148$,

$$T_2 = 46, \quad T_\infty = 39$$

$$||u - U^{(n)}||_2 = 0.269241, \quad ||u - U^{(n)}||_\infty = 0.122342,$$

which is possible that do not represent the best case, $(\alpha, \gamma) = (0.7, 1.9)$. In fact we will show in the following Chapters, the next step is to refine the spatial grid, change the initial condition, or add a reaction term, say $f(x, t)$, on the right-hand side of equation (2.8), and consider also the case of higher space dimensions.

We now apply to the experimental data a filter based on a *wavelet decomposition* of the data, see [141]. The best results are obtained correspondingly to $(\gamma, \alpha) = (0.7, 1.9)$. The smallest discrepancy is now observed on the first decimal digit.

(γ, α)	N_x	N_t	T_2	T_∞	$\|\|u - U^{(n)}\|\|_2$	
$(0.7, 1.1)$	21	37	15	12	0.509395	0.195058
$(0.7, 1.2)$	21	37	16	12	0.488999	0.183744
$(0.7, 1.3)$	21	37	17	13	0.464047	0.174749
$(0.7, 1.4)$	21	37	18	14	0.434435	0.164367
$(0.7, 1.5)$	21	37	19	15	0.400974	0.152417
$(0.7, 1.6)$	21	37	21	16	0.365235	0.141848
$(0.7, 1.7)$	21	37	22	18	0.328055	0.130333
$(0.7, 1.8)$	21	37	24	19	0.292461	0.118418
$(0.7, 1.9)$	21	37	26	21	0.259294	0.109248

TABLE 2.10: $\gamma = 0.7$ and $\alpha = 1.1, \ldots, 1.9$ with *filtered* data.

When $\alpha = 2$ and $\gamma = 1$ (that is the case of *classical* diffusion), we obtained larger errors, at least three times larger in the infinite norm. The best results are obtained for $\alpha = 2$ (classical Laplacian), but with a time fractional derivative of order $\gamma = 2.0$. In this case we obtain about $\|u - U^{(n)}\|_\infty = 0.100303$.

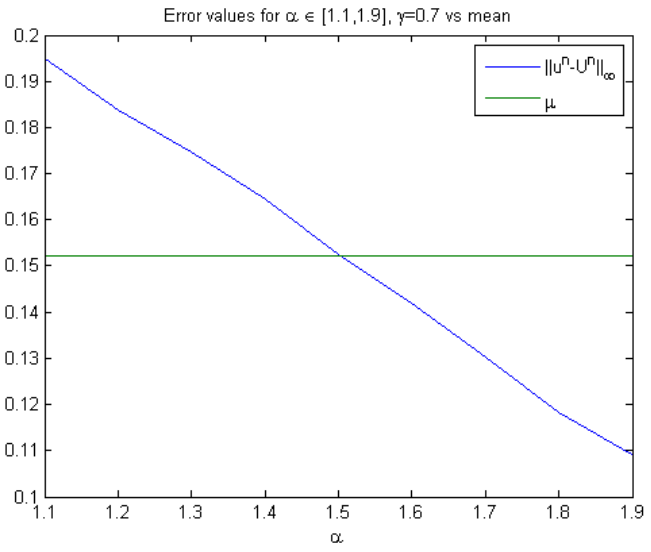

FIGURE 2.10: Error for $a = 1, N = 37, M = 21$.

Chapter 3

Numerical solution of two-dimensional fractional diffusion equations by a high-order ADI method

Fractional partial diffusion or reaction-diffusion differential equations in several space dimensions can be used to model several phenomena in many fields of Science, and in particular in Meteorology. In case of anomalous diffusion, generalized models using fractional derivatives, thus leading to fractional partial diffusion equations have indeed been proposed, especially to describe diffusion and transport dynamics in complex systems [6, 31, 65, 93]. In fact, such equations may describe fluid flow through porous media better than classical diffusion equations. For instance, fractional time-derivatives may account for (time) *delays*, while fractional space-derivatives may explain a *nonlocal* behavior, typically characterized by power law (rather than exponential law) decay. Equations like these are also used in groundwater hydrology to model the transport of passive tracers carried by fluid flows in porous media and seepage [8–10].

3.1 High-order approximations for Riemann-Liouville fractional derivatives

In this section, we recall some known results concerning fractional derivatives. We begin with the definition of the Riemann-Liouville (RL) fractional derivatives, and the properties of their Fourier transform.

Definition 3.1. [112] If $n-1 < \alpha \leq n$ for some $n \in \mathbf{N}$, the RL fractional left and right derivative of order α of the function $u(x)$, whose domain is $[a,b]$, at the point $x \in [a,b]$, are defined as follows.

1. *left* Riemann-Liouville fractional derivative:

$$ {}_aD_x^\alpha u(x) := \frac{1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_a^x \frac{u(\xi)}{(x-\xi)^{\alpha-n+1}}\, d\xi; $$

2. *right* Riemann-Liouville fractional derivative:

$$ {}_xD_b^\alpha u(x) := \frac{-1}{\Gamma(n-\alpha)} \frac{d^n}{dx^n} \int_x^b \frac{u(\xi)}{(\xi-x)^{\alpha-n+1}}\, d\xi, $$

   where $\Gamma(\cdot)$ denotes the Gamma function.

When $\alpha = n$, ${}_aD_x^\alpha = \frac{d^n}{dx^n}$ and ${}_xD_b^\alpha = (-1)^n \frac{d^n}{dx^n}$.

The following property is important (see [57]). Let be $\alpha > 0$, $u \in C_0^\infty(\Omega)$, being Ω open and $\Omega \subseteq \mathbf{R}$. The Fourier transforms of the left and right RL fractional derivatives (denoted by $\mathcal{F}$) satisfy

$$ \mathcal{F}\left[{}_{-\infty}D_x^\alpha u(x)\right] = (i\omega)^\alpha \hat{u}(\omega), $$

$$ \mathcal{F}\left[{}_xD_\infty^\alpha u(x)\right] = (-i\omega)^\alpha \hat{u}(\omega), $$

where $\hat{u}(\omega)$ is the ordinary Fourier transform of u,

$$ \hat{u}(x) := \int_{-\infty}^{+\infty} e^{i\omega x} u(x)\, dx. $$

In [101], it was shown that the "shifted Grünwald difference operator", defined as

$$ A_{h,p}^\alpha u(x) := \frac{1}{h^\alpha} \sum_{k=0}^{\infty} g_k^{(\alpha)}\, u(x-(k-p)h), \tag{3.1} $$

where p is an integer, and $g_k^{(\alpha)} := (-1)^k \binom{\alpha}{k}$, approximates the *left* RL fractional derivative of order α, uniformly, with first-order accuracy, i.e.,

$$ A_{h,p}^\alpha u(x) = {}_{-\infty}D_x^\alpha u(x) + O(h), \tag{3.2} $$

The $g_k^{(\alpha)}$ in (3.1) are the coefficients of the power series expansion of $(1-z)^\alpha$,

$$ (1-z)^\alpha = \sum_{k=0}^{\infty} (-1)^k \binom{\alpha}{k} z^k = \sum_{k=0}^{\infty} g_k^{(\alpha)} z^k, \tag{3.3} $$

for all $|z| \leq 1$, and can be evaluated recursively,

$$g_0^{(\alpha)} = 1, \qquad g_k^{(\alpha)} = \left(1 - \frac{\alpha+1}{k}\right) g_{k-1}^{(\alpha)}, \ k = 1, 2, ... \tag{3.4}$$

Lemma 1. The coefficients $g_k^{(\alpha)}$ in (3.1) enjoy the properties

$$\begin{cases} g_0^{(\alpha)} = 1, \quad g_1^{(\alpha)} = -\alpha < 0, \\ 1 \geq g_2^{(\alpha)} \geq g_3^{(\alpha)} \geq \ldots \geq 0, \\ \sum_{k=0}^{\infty} g_k^{(\alpha)} = 0, \quad \sum_{k=0}^{m} g_k^{(\alpha)} < 0 \qquad \text{for every } m \geq 1, \end{cases} \tag{3.5}$$

provided that $1 < \alpha \leq 2$.

Theorem 6. Let be $u \in L^1(\mathbf{R})$, and hence (as is known), ${}_{-\infty}D_x^{\alpha+2}u$ and its Fourier transform also belong to $L^1(\mathbf{R})$, and define the "weighted and shifted" Grünwald difference ($WSGD$) operator ${}_L\mathcal{D}_{h,p,q}^{\alpha}$ by

$${}_L\mathcal{D}_{h,p,q}^{\alpha}u(x) := \frac{\alpha - 2q}{2(p-q)} A_{h,p}^{\alpha}u(x) + \frac{2p - \alpha}{2(p-q)} A_{h,q}^{\alpha}u(x). \tag{3.6}$$

Then, we have

$${}_L\mathcal{D}_{h,p,q}^{\alpha}u(x) = {}_{-\infty}\mathcal{D}_x^{\alpha}u(x) + O(h^2) \tag{3.7}$$

uniformly for $x \in \mathbf{R}$, where p,$q \in \{-1, 0, 1\}$, with $p \neq q$.

Note that p and q enter symmetrically in (4.9), since ${}_L\mathcal{D}_{h,q,p}^{\alpha}u(x) = {}_L\mathcal{D}_{h,p,q}^{\alpha}u(x)$.

Remark 3.2. For the *right* RL fractional derivative, being p and q integers, and

$$B_{h,q}^{\alpha}u(x) := \frac{1}{h^{\alpha}} \sum_{k=0}^{\infty} g_k^{(\alpha)} u(x + (k-q)h), \tag{3.8}$$

we can check that, as in Theorem 6.

$${}_L\mathcal{D}_{h,p,q}^{\alpha}u(x) = \frac{\alpha - 2q}{2(p-q)} B_{h,p}^{\alpha}u(x) + \frac{2p - \alpha}{2(p-q)} B_{h,q}^{\alpha}u(x) = {}_xD_{\infty}^{\alpha}u(x) + O(h^2), \tag{3.9}$$

uniformly for $x \in \mathbf{R}$, under the condition that u, ${}_xD_{\infty}^{\alpha+2} \in L^1(\mathbf{R})$ and the Fourier transform of the latter belongs to $L^1(\mathbf{R})$.

Remark 3.3. Let $u(x)$ be defined on the bounded interval $[a, b]$. If $u(a) = 0$ or $u(b) = 0$, we extend $u(x)$ to be zero for $x < a$ or for $x > b$, respectively. Then, the left and right RL fractional derivatives of $u(x)$, of order α, at each point x, can be approximated by

the WSGD operators with second-order accuracy,

$$\begin{aligned} {}_aD_x^\alpha u(x) &= \frac{\lambda_1}{h^\alpha}\sum_{k=0}^{[\frac{x-a}{h}]+p} g_k^{(\alpha)}\, u(x+(k-p)h) \\ &\quad +\tfrac{\lambda_2}{h^\alpha}\textstyle\sum_{k=0}^{[\frac{x-a}{h}]+q} g_k^{(\alpha)}\, u(x+(k-q)h)+O(h^2), \\ {}_xD_b^\alpha u(x) &= \frac{\lambda_2}{h^\alpha}\sum_{k=0}^{[\frac{b-x}{h}]+p} g_k^{(\alpha)}\, u(x+(k-p)h) \\ &\quad +\tfrac{\lambda_2}{h^\alpha}\textstyle\sum_{k=0}^{[\frac{b-x}{h}]+p} g_k^{(\alpha)}\, u(x+(k-q)h)+O(h^2), \end{aligned} \tag{3.10}$$

where $\lambda_1 := \frac{\alpha-2q}{2(p-q)}$, $\lambda_2 := \frac{2p-\alpha}{2(p-q)}$.

Remark 3.4. Note that p [q] represents the number of points located at the right [left] of the point x, used for evaluating the left [right] RL fractional derivative of order α at x, respectively. Hence, when employing the difference method with (3.10) to approximate FDEs with non-periodic boundary conditions, p and q should be chosen in the set $\{-1,0,1\}$, to ensure that the nodes where the values of u are needed in (3.10) lie inside the (bounded) interval. Otherwise, when x is close to the right or to the left endpoint of the interval, another kind of discretization for the fractional derivative should be used, see [44]. Using the pair $(p,q)=(0,-1)$, the approximation method turns out to be unstable for time-dependent problems, see [44]. The two sets of values for (p,q) to set up a difference scheme for FDEs, $(1,0)$ and $(1,-1)$ should be chosen instead. The corresponding weights in (4.9) and (3.9) are then $(\frac{\alpha}{2},\frac{2-\alpha}{2})$ and $(\frac{2+\alpha}{4},\frac{2-\alpha}{4})$. For $\alpha=2$, the WSGD operator in (4.9) turns out to be the centered difference approximation of the second-order derivative, when $(p,q)=(1,0)$ or $(p,q)=(1,-1)$. For $\alpha=1$, $(p,q)=(1,0)$, the centered difference scheme for first-order derivative is recovered.

Therefore, the simplified forms of the discrete approximations for RL fractional derivatives with $(p,q)=(1,0),(1,-1)$ given in (3.10) are

$$\begin{aligned} {}_aD_x^\alpha u(x_i) &= \frac{\lambda_1}{h^\alpha}\sum_{k=0}^{[\frac{x-a}{h}]+p} w_k^{(\alpha)}\, u(x_{i-k+1})+O(h^2), \\ {}_xD_b^\alpha u(x_i) &= \frac{\lambda_1}{h^\alpha}\sum_{k=0}^{[\frac{x-a}{h}]+p} w_k^{(\alpha)}\, u(x_{i+k-1})+O(h^2), \end{aligned} \tag{3.11}$$

where

$$\begin{aligned} (p,q)=(1,0),\quad & w_0^{(\alpha)} := \frac{\alpha}{2}g_0^{(\alpha)}, w_k^{(\alpha)} := \frac{\alpha}{2}g_k^{(\alpha)}+\frac{2-\alpha}{2}g_{k-1}^{(\alpha)}, k\geq 1; \\ (p,q)=(-1,1),\quad & w_0^{(\alpha)} := \frac{2+\alpha}{4}g_0^{(\alpha)}, w_1^{(\alpha)} := \frac{2+\alpha}{4}g_1^{(\alpha)}, \\ & w_k^{(\alpha)} := \frac{2+\alpha}{4}g_k^{(\alpha)}+\frac{2-\alpha}{4}g_{k-2}^{(\alpha)}, k\geq 2. \end{aligned} \tag{3.12}$$

Third-order approximations (in time) can be obtained. As in case of second-order approximations for the RL fractional derivatives, we give here a combination of *three* shifted Grünwald difference operators,

$$ {}_L\mathcal{G}^{\alpha}_{h,p,q,r}u(x) := \lambda_1\, A^{\alpha}_{h,p}u(x) + \lambda_2\, A^{\alpha}_{h,q}u(x) + \lambda_3\, A^{\alpha}_{h,r}u(x), \tag{3.13} $$

where p, q, r are some given mutually different integers, and [43]

$$ \begin{aligned} \lambda_1 &:= \tfrac{12qr-(6q+6r+1)\alpha+3\alpha^2}{12(pr-pq-pr+p^2)}, \\ \lambda_2 &:= \tfrac{12pr-(6p+6r+1)\alpha+3\alpha^2}{12(pr-pq-qr+q^2)}, \\ \lambda_3 &:= \tfrac{12pq-(6p+6q+1)\alpha+3\alpha^2}{12(pq-pq-qr+r^2)}. \end{aligned} \tag{3.14} $$

Assuming $u \in L^1(\mathbf{R})$, and taking the Fourier transform in (3.13), we obtain

$$ \begin{aligned} \mathcal{F}[{}_L\mathcal{G}^{\alpha}_{h,p,q,r}u(x)](\omega) &= (i\omega)^{\alpha}\, \left(\lambda_1\, W_p(i\omega h) + \lambda_2\, W_q(i\omega h) + \lambda_3\, W_r(i\omega h)\right) \tilde{u}(\omega) \\ &= (i\omega)^{\alpha}\, \left(1 + C(i\omega h)^3\right) \hat{u}(\omega), \end{aligned} \tag{3.15} $$

where

$$ W_r(z) := \left(\frac{1-e^{-z}}{z}\right)^{\alpha} e^{rz} = 1 + \left(r - \frac{\alpha}{2}\right) + O(z^2), $$

being $r = p$ or q. The function $W_r(z)$ is called the *Wright function*, and is known to play a fundamental role in various applications of fractional calculus, see [94].

If ${}_{-\infty}D^{\alpha+3}_x u(x)$ and its Fourier transform both belong to $L^1(\mathbf{R})$, we have

$$ \begin{aligned} \left|\left[{}_L\mathcal{G}^{\alpha}_{h,p,q,r}u(x) - {}_{-\infty}D^{\alpha}_x u(x)\right](\omega)\right| &\le \frac{1}{2\pi}\int_{-\infty}^{+\infty} \left|\mathcal{F}[{}_L\mathcal{G}^{\alpha}_{h,p,q,r}u - {}_{-\infty}D^{\alpha}_x\right| dx \\ &\le C\, \|\mathcal{F}[{}_{-\infty}D^{\alpha+3}_x](\omega)\|_{L^1}\, h^3 = O(h^3). \end{aligned} \tag{3.16} $$

We have also

$$ \begin{aligned} {}_L\mathcal{G}^{\alpha}_{h,p,q,r}u(x) &= \lambda_1\, B^{\alpha}_{h,p}u(x) + \lambda_2\, B^{\alpha}_{h,q}u(x) + \lambda_3\, B^{\alpha}_{h,r}u(x) \\ &= {}_{-\infty}D^{\alpha}_x u(x) + O(h^3), \end{aligned} \tag{3.17} $$

uniformly for $x \in \mathbf{R}$. The operator $B^{\alpha}_{h,s}$ is given in (3.8), and the λ_i's, $i = 1, 2, 3$, are the same as in (3.14).

As stated in Remark 3.4, the third-order WSGD operator can be used to approximately solve RL fractional differential equations on bounded domains by finite difference methods, choosing the triplet $(p, q, r) = (1, 0, -1)$ and then the corresponding weight coefficients given in (3.14), hence $\lambda_1 = \frac{5}{24}\alpha + \frac{1}{8}\alpha^2$, $\lambda_2 = 1 + \frac{1}{12}\alpha - \frac{1}{4}\alpha^2$, $\lambda_3 = -\frac{7}{24}\alpha - \frac{1}{8}\alpha^2$.

For a function $u(x)$ with $u(a) = u(b) = 0$, the matrix

$$G_1 = \lambda_1 \begin{pmatrix} g_1^\alpha & g_0^\alpha & & & \\ g_2^\alpha & g_1^\alpha & g_0^\alpha & & \\ \vdots & g_2^\alpha & g_1^\alpha & \ddots & \\ g_{n-2}^\alpha & \dots & \ddots & \ddots & g_0^\alpha \\ g_{n-1}^\alpha & g_{n-2}^\alpha & \dots & g_2^\alpha & g_1^\alpha \end{pmatrix} + \lambda_2 \begin{pmatrix} g_0^\alpha & & & & \\ g_1^\alpha & g_0^\alpha & & & \\ \vdots & g_1^\alpha & g_0^\alpha & & \\ g_{n-3}^\alpha & \dots & \ddots & \ddots & \\ g_{n-2}^\alpha & g_{n-3}^\alpha & \dots & g_1^\alpha & g_0^\alpha \end{pmatrix} + \lambda_3 \begin{pmatrix} 0 & & & & \\ g_0^\alpha & 0 & & & \\ \vdots & g_0^\alpha & 0 & & \\ g_{n-4}^\alpha & \dots & \ddots & \ddots & \\ g_{n-3}^\alpha & g_{n-4}^\alpha & \dots & g_0^\alpha & 0 \end{pmatrix} \tag{3.18}$$

with $(p, q, r) = (1, 0, -1)$ and on the grid points $\{x_k = a + kh, h = (b-a)/n, k = 1, ..., n-1\}$, is a finite-dimensional approximation of the operator in (3.13).

3.2 Fractional diffusion equations in two space dimensions

In this section, we review a few numerical methods which have been proposed in the literature to solve FDEs. These will provide the grounds we need to construct our scheme.

Inspired by the definition of the shifted Grünwald difference operator given in (3.1), and by the idea of the fractional steps methods, a second-order approximation has been derived for the RL fractional derivatives, which are used as space operator, while the time derivative remains classical, see [43]. Consider the following FDE in two dimensions,

$$\begin{cases} \frac{\partial u}{\partial t} = \left(K_1^+ \, {}_aD_x^\alpha u + K_2^+ \, {}_xD_b^\alpha u\right) \\ \qquad + \left(K_1^- \, {}_cD_y^\beta u + K_2^- \, {}_yD_d^\beta u\right) + f(x, y, t), \\ \qquad\qquad\qquad\qquad\qquad\qquad (x, y, t) \in \Omega \times [0, T], \\ u(x, y, 0) = u_0(x, y), \qquad\qquad (x, y, t) \in \Omega, \\ u(x, y, 0) = \varphi(x, y, t), \qquad\qquad (x, y, t) \in \partial\Omega \times [0, T], \end{cases} \tag{3.19}$$

where $u \equiv u(x, y, t)$.

Here $\Omega := (a, b) \times (c, d)$, ${}_aD_x^\alpha$, ${}_xD_b^\alpha$, and ${}_cD_y^\beta$, ${}_yD_d^\beta$ are RL fractional operators with $1 < \alpha, \beta \leq 2$. The diffusion coefficients are given, and satisfy the inequalities $K_i^+, K_i^- \geq 0$, $i = 1, 2$, $(K_1^+)^2 + (K_2^+)^2 \neq 0$, and $(K_1^-)^2 + (K_2^-)^2 \neq 0$. The boundary value, φ, satisfies

the condition $\varphi(a,y,t)=0$ if $K_1^+\neq 0$; $\varphi(b,y,t)=0$ if $K_1^-\neq 0$; $\varphi(x,c,t)=0$ if $K_2^+\neq 0$; $\varphi(x,d,t)=0$ if $K_2^-\neq 0$. We assume that the problem in (3.19) has a unique sufficiently smooth solution.

3.2.1 The CN-WSGD scheme

In this section we derive a Crank-Nicolson difference scheme by using the WSGD formula (3.10) for problem (3.19). The resulting schemes which implement this method are named *compact* [37, 44]. We make a partition of the domain Ω by a uniform mesh with space steps $h_x:=(b-a)/N_x$, $h_y:=(d-c)/N_y$, and time step $\tau:=T/M$, where N_x, N_y, and M are positive integers. Then, the grid points will be $x_i:=ih_x$, $y_j:=jh_y$, and $t_n:=n\tau$, for $1\leq i\leq N_x$, $1\leq j\leq N_y$, and $0\leq n\leq M$. Let define $t_{n+1/2}:=(t_n+t_{n+1})/2$ for $0\leq n\leq M-1$, and use the following notation

$$u_{i.j}^n:=u(x_i,y_j,t_n),\quad f_{i,j}^{n+1/2}:=f(x_i,y_j,t_{n+1/2}),\quad \delta_t u_{i,j}^n:=\frac{u_{i,j}^{n+1}-u_{i,j}^n}{\tau}.$$

Time discretization of (3.19) leads to

$$\begin{aligned}\delta_t u_{i,j}^n=&\\ &\frac{1}{2}\Big[K_+^1\,({}_aD_x^\alpha u)_{i,j}^{n+1}+K_+^2\,({}_xD_b^\alpha u)_{i,j}^{n+1}+K_-^1\,({}_cD_y^\beta u)_{i,j}^{n+1}+K_-^2\,({}_yD_d^\beta u)_{i,j}^{n+1}\\ &+K_+^1\,({}_aD_x^\alpha u)_{i,j}^{n}+K_+^2\,({}_xD_b^\alpha u)_{i,j}^{n}+K_-^1\,({}_cD_y^\beta u)_{i,j}^{n}+K_-^2\,({}_yD_d^\beta u)_{i,j}^{n})\Big]\\ &+f_{i,j}^{n+1/2}+O(\tau^2).\end{aligned}\tag{3.20}$$

For the space discretization we used the WSGD operators ${}_L\mathcal{D}_{h_x,p,q}^\alpha u$, ${}_R\mathcal{D}_{h_x,p,q}^\alpha u$, and ${}_L\mathcal{D}_{h_y,p,q}^\beta u$, ${}_R\mathcal{D}_{h_y,p,q}^\beta u$, to approximate the fractional diffusion terms ${}_aD_x^\alpha u$, ${}_xD_b^\alpha u$, and ${}_cD_y^\beta u$, ${}_yD_d^\beta u$, respectively. Multiplying both sides of (3.20) by τ and separating the time levels (i.e., u^n and u^{n+1}), we obtain

$$\begin{aligned}&\left(1-\frac{K_+^1\tau}{2}\,{}_L\mathcal{D}_{h_x,p,q}^\alpha-\frac{K_+^2\tau}{2}\,{}_R\mathcal{D}_{h_x,p,q}^\alpha-\frac{K_-^1\tau}{2}\,{}_L\mathcal{D}_{h_y,p,q}^\beta-\frac{K_-^2\tau}{2}\,{}_R\mathcal{D}_{h_y,p,q}^\beta\right)u_{i,j}^{n+1}\\ &=\left(1+\frac{K_+^1\tau}{2}\,{}_L\mathcal{D}_{h_x,p,q}^\alpha+\frac{K_+^2\tau}{2}\,{}_R\mathcal{D}_{h_x,p,q}^\alpha+\frac{K_-^1\tau}{2}\,{}_L\mathcal{D}_{h_y,p,q}^\beta+\frac{K_-^2\tau}{2}\,{}_R\mathcal{D}_{h_y,p,q}^\beta\right)u_{i,j}^{n}\\ &+\tau f_{i,j}^{n+1/2}+\tau\,\varepsilon_{i,j}^n,\end{aligned}\tag{3.21}$$

where $\varepsilon_{i,j}^n$ denotes the *local truncation error*, $|\varepsilon_{i,j}^n|\leq\tilde{c}\,(\tau^2+h^2)$. Let define also

$$\delta_x^\alpha:=K_1^+\,{}_L\mathcal{D}_{h_x,p,q}^\alpha+K_2^+\,{}_R\mathcal{D}_{h_x,p,q}^\alpha,\qquad \delta_y^\beta:=K_1^-\,{}_L\mathcal{D}_{h_y,p,q}^\beta+K_2^-\,{}_R\mathcal{D}_{h_y,p,q}^\beta.$$

We chose, for simplicity, the same step sizes, $h_x = h_y = h$. By Taylor expansion, we obtain

$$\frac{\tau^2}{4}\delta_x^\alpha \delta_y^\beta (u_{i,j}^{n+1} - u_{i,j}^n) \tag{3.22}$$

$$= \frac{\tau^3}{4}\left(\left(K_{+\,a}^1 D_x^\alpha + K_{+\,x}^2 D_b^\alpha\right)\left(K_{-\,c}^1 D_y^\beta + K_{-\,y}^2 D_d^\beta\right) u_t\right)_{i,j}^{n+1/2} + O(\tau^5 + \tau^2 h^2).$$

Taking into account formula (3.22) on the right-hand side of (3.21), and factoring, leads to

$$\left(1 - \frac{\tau^2}{2}\delta_x^\alpha\right)\left(1 - \frac{\tau^2}{2}\delta_y^\beta\right) u_{i,j}^{n+1} \tag{3.23}$$

$$= \left(1 + \frac{\tau^2}{2}\delta_x^\alpha\right)\left(1 + \frac{\tau^2}{2}\delta_y^\beta\right) u_{i,j}^n + \tau\, f_{i,j}^{n+1/2} + \tau\, \varepsilon_{i,j}^n + O(\tau^3 + \tau^3 h^2).$$

If $U_{i,j}^n$ denotes the numerical approximation to $u_{i,j}^n$, we obtain the finite difference approximation for problem (3.19),

$$\left(1 - \frac{\tau^2}{2}\delta_x^\alpha\right)\left(1 - \frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^{n+1} = \left(1 + \frac{\tau^2}{2}\delta_x^\alpha\right)\left(1 + \frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^n + \tau\, f_{i,j}^{n+1/2}. \tag{3.24}$$

In order to solve (3.24) efficiently, the following techniques can be adopted.

A) The *Peaceman-Rachford ADI* scheme [127],

$$\left(1 - \frac{\tau^2}{2}\delta_x^\alpha\right) V_{i,j}^n = \left(1 + \frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^n + \tau\, f_{i,j}^{n+1/2}, \tag{3.25a}$$

$$\left(1 - \frac{\tau^2}{2}\delta_x^\beta\right) U_{i,j}^{n+1} = \left(1 + \frac{\tau^2}{2}\delta_y^\alpha\right) V_{i,j}^n + \tau\, f_{i,j}^{n+1/2}. \tag{3.25b}$$

B) The *Douglas* ADI scheme [52],

$$(1 - \frac{\tau^2}{2}\delta_x^\alpha) V_{i,j}^n = (1 + \frac{\tau^2}{2}\delta_y^\alpha + \tau\delta_y^\beta) U_{i,j}^n + \tau f_{i,j}^{n+1/2}, \tag{3.26a}$$

$$(1 - \frac{\tau^2}{2}\delta_x^\beta) U_{i,j}^{n+1} = V_{i,j}^n - \frac{\tau^2}{2}\delta_x^\beta U_{i,j}^n. \tag{3.26b}$$

C) The *D'Yakonov* ADI scheme [127],

$$\left(1 - \frac{\tau^2}{2}\delta_x^\alpha\right) V_{i,j}^n = \left[\left(1 + \frac{\tau^2}{2}\delta_y^\beta\right) + \tau\delta_y^\beta\right] U_{i,j}^n + \tau\, f_{i,j}^{n+1/2}, \tag{3.27a}$$

$$\left(1 - \frac{\tau^2}{2}\delta_x^\beta\right) U_{i,j}^{n+1} = V_{i,j}^n. \tag{3.27b}$$

A simple calculation shows that

$$\frac{\tau^3}{2}\delta_x^\alpha\delta_x^\beta f_{i,j}^{n+1/2} = \frac{\tau^3}{2}\left(K_{+\,a}^1 D_x^\alpha + K_{+\,x}^2 D_b^\alpha\right)\left(K_{-\,c}^1 D_y^\beta + K_{-\,y}^2 D_d^\beta\right) f_{i,j}^{n+1/2} \tag{3.28}$$

$$+O(\tau^3 h^2).$$

Then, it follows from (3.23) and (3.28) that

$$\left(1-\frac{\tau^2}{2}\delta_x^\alpha\right)\left(1-\frac{\tau^2}{2}\delta_y^\beta\right) u_{i,j}^{n+1} \tag{3.29}$$

$$= \left(1+\frac{\tau^2}{2}\delta_x^\alpha\right)\left(1+\frac{\tau^2}{2}\delta_y^\beta\right) u_{i,j}^{n} + \tau\, f_{i,j}^{n+1/2} + \frac{\tau^3}{2}\delta_x^\alpha\delta_x^\beta f_{i,j}^{n+1/2} + \tau\,\tilde{\varepsilon}_{i,j}^{n},$$

where

$$\tilde{\varepsilon}_{i,j}^{n} = \varepsilon_{i,j}^{n} - \frac{\tau^2}{2}(K_{+\,a}^1 D_x^\alpha + K_{+\,x}^2 D_b^\alpha)(K_{-\,c}^1 D_y^\beta + K_{-\,y}^2 D_d^\beta) f_{i,j}^{n+1/2} \tag{3.30}$$

$$+O(\tau^2+\tau^2 h^2).$$

Eliminating the truncating error, we have

$$\left(1-\frac{\tau^2}{2}\delta_x^\alpha\right)\left(1-\frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^{n+1} \tag{3.31}$$

$$= \left(1+\frac{\tau^2}{2}\delta_x^\alpha\right)\left(1+\frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^{n} + \tau\, f_{i,j}^{n+1/2} + \frac{\tau^3}{2}\delta_x^\alpha\delta_x^\beta f_{i,j}^{n+1/2}.$$

Introducing the intermediate variable $V_{i,j}^n$, we obtain the locally one-dimensional (LOD) scheme mentioned in [115, 140],

$$\left(1-\frac{\tau^2}{2}\delta_x^\alpha\right) V_{i,j}^{n} = \left(1+\frac{\tau^2}{2}\delta_x^\alpha\right) U_{i,j}^{n} + \frac{\tau}{2}\left(1+\frac{\tau^2}{2}\delta_x^\alpha\right) f_{i,j}^{n+1/2}, \tag{3.32a}$$

$$\left(1-\frac{\tau^2}{2}\delta_y^\beta\right) U_{i,j}^{n+1} = \left(1+\frac{\tau^2}{2}\delta_y^\beta\right) V_{i,j}^{n} + \frac{\tau}{2}\left(1+\frac{\tau^2}{2}\delta_y^\beta\right) f_{i,j}^{n+1/2}, \tag{3.32b}$$

3.2.2 A third-order CN-WSGD scheme

In the space discretization, we choose the so-called 3-WSGD operators, i.e., the weighted and shifted Grünwald difference operators, third-order accurate in time, namely ${}_L\mathcal{G}_{h_x,p,q,r}^{\alpha}u$, ${}_R\mathcal{G}_{h_y,p,q,r}^{\alpha}u$, and ${}_L\mathcal{G}_{h_y,p,q}^{\beta}u$, ${}_R\mathcal{G}_{h_y,p,q}^{\beta}u$, to approximate the fractional diffusion terms ${}_aD_x^\alpha u$, ${}_xD_b^\alpha u$, and ${}_cD_y^\beta u$, ${}_yD_d^\beta u$, respectively, see [128]. Multiplying both sides of (3.20) by τ,

and separating the terms containing u^n and u^{n+1}, we have

$$\begin{aligned}
&\left(1-\frac{K_+^1\tau}{2}\,{}_L\mathcal{G}^{\alpha}_{h_x,p,q}-\frac{K_+^2\tau}{2}\,{}_R\mathcal{G}^{\alpha}_{h_x,p,q}-\frac{K_-^1\tau}{2}\,{}_L\mathcal{G}^{\beta}_{h_y,p,q}-\frac{K_-^2\tau}{2}\,{}_R\mathcal{G}^{\beta}_{h_y,p,q}\right)u_{i,j}^{n+1}\\
&=\left(1+\frac{K_+^1\tau}{2}\,{}_L\mathcal{G}^{\alpha}_{h_x,p,q}+\frac{K_+^2\tau}{2}\,{}_R\mathcal{G}^{\alpha}_{h_x,p,q}+\frac{K_-^1\tau}{2}\,{}_L\mathcal{G}^{\beta}_{h_y,p,q}+\frac{K_-^2\tau}{2}\,{}_R\mathcal{G}^{\beta}_{h_y,p,q}\right)u_{i,j}^{n}\\
&+\tau\, f_{i,j}^{n+1/2}+\tau\,\varepsilon_{i,j}^{n},
\end{aligned}\tag{3.33}$$

where $\varepsilon_{i,j}^n$ denotes the (local) truncation error, and we have $|\varepsilon_{i,j}^n|\le\tilde{c}\,(\tau^2+h^3)$. We also write

$$\delta_x^{\alpha}=K_1^{+}\,{}_L\mathcal{G}^{\alpha}_{h_x,p,q}+K_2^{+}\,{}_R\mathcal{G}^{\alpha}_{h_x,p,q},\qquad \delta_y^{\beta}=K_1^{-}\,{}_L\mathcal{G}^{\beta}_{h_y,p,q}+K_2^{-}\,{}_R\mathcal{G}^{\beta}_{h_y,p,q}.$$

We chose again the same step sizes, $h_x=h_y=h$. A Taylor expansion yields

$$\begin{aligned}
&\frac{\tau^2}{4}\delta_x^{\alpha}\delta_y^{\beta}\left(u_{i,j}^{n+1}-u_{i,j}^{n}\right)\\
&\qquad=\frac{\tau^3}{4}\left[\left(K_+^1\,{}_aD_x^{\alpha}+K_2^{+}\,{}_xD_b^{\alpha}\right)\left(K_1^{-}\,{}_cD_y^{\beta}+K_2^{-}\,{}_yD_d^{\beta}\right)u_t\right]_{i,j}^{n+1/2}\\
&\qquad\quad+O(\tau^5+\tau^2h^3).
\end{aligned}\tag{3.34}$$

3.3 A new numerical method with examples

In this section, we construct a numerical method to solve two-dimensional fractional diffusion problems. Examples are then given to illustrate the performance of the algorithm.

3.3.1 Theoretical considerations

In [153], a numerical method of the ADI type was proposed to solve FDEs in 2D which is second-order accurate in space and third-order in time. This method however was *not* implemented there. Here we did implement it, but, in addition, we generalized their extrapolation technique so to *optimize* the algorithm. In this way, we were able to *accelerate* it, and this was accomplished exploiting the *PageRank* method, widely used by computer scientists, but not much by numerical analysts. Some details follow.

A considerable amount of computing time can be saved just resorting to an *extrapolation* technique. This procedure may also increase the accuracy of the method up to the third order in time [97]. Let describe such a technique.

Step 1. Compute the numbers ζ_1, ζ_2, and ζ_3, solving the three linear algebraic equations

$$\begin{cases} \zeta_1 + \zeta_2 + \zeta_3 = 1 \\ \zeta_1 + \frac{1}{2}\zeta_2 + \frac{1}{4}\zeta_3 = 0 \\ \zeta_1 + \frac{1}{3}\zeta_2 - \frac{1}{9}\zeta_3 = 0. \end{cases}$$

An *extrapolated solution*, depending on U^n, is then used to solve the problem. The quantity U^n requires evaluating certain coefficients, which can be obtained by the "*PageRank* accelerating method" [64]. The previous algebraic system yields the optimal coefficients $\zeta_1, \zeta_2, \zeta_3$.

Step 2. Compute the solution U^n of a *compact difference scheme* [43, 125, 131, 153] (see below, at step 3), with the three time step sizes τ, $\frac{2}{3}\tau$, and $\frac{\tau}{3}$ [131]. This kind of methods is usually adopted for steady convection-diffusion numerical problems on uniform grids (see [125]), rather than for time-dependent problems.

Step 3. Evaluate the *extrapolated solution*, $W^n(\tau)$, by

$$W^n(\tau) = \zeta_1\, U^n(\tau) + \zeta_2\, U^n\left(\frac{2}{3}\tau\right) + \zeta_3\, U^n\left(\frac{\tau}{3}\right),$$

where we have displayed the precise dependence of U^n on τ.

3.3.2 Numerical examples

Consider now the model problem

Example 1. Let be the FDE

$$ {}_0D_t^{\gamma} u = {}_0D_x^{\alpha} u + {}_xD_1^{\alpha} u + {}_0D_y^{\beta} u + {}_yD_1^{\beta} u + f(x,y,t) \tag{3.35}$$

on the space domain $\Omega := (0,1) \times (0,1)$, for $t > 0$, subject to the boundary conditions

$$u(x,y,t) = 0, \quad (x,y) \in \partial\Omega, \quad t \in [0,1],$$

with the initial value

$$u(x,y,0) = x^3(1-x)^3 y^3 (1-y)^3 \quad (x,y) \in [0,1] \times [0,1].$$

Note that, whenever the fractional order α is less than but close to 2 (subdiffusion), the first two terms on the right-hand side of (3.35) replace, in some sense, the second

derivative u_{xx} appearing in the classical diffusion equations. Similarly, the third and fourth terms replace u_{yy}. Moreover even *anisotropic* diffusion effects can be included in this formulation, since α and β do not need to be equal.

Generally speaking, according to the values of the fractional orders, α and β, some derivatives might be considered as replacing gradient terms rather than diffusion. For instance, in this model, choosing for α some values between 1 and 2, and for β some value between 0 and 1, one might think to simulate some fractional generalization of the advection-diffusion equation $u_t = u_{xx} + u_y + f(x, y, t)$.

If in (3.35), we choose the source term to be

$$\begin{aligned} f(x,y,t) = &- e^{-t}x^3(1-x)^3y^3(1-y)^3 \\ &+ (\frac{3!}{\Gamma(4-\alpha)}(x^{3-\alpha}+(1-x)^{3-\alpha}) - \frac{3\cdot 4!}{\Gamma(5-\alpha)}(x^{3-\alpha}+(1-x)^{3-\alpha}) \\ &+ \frac{3\cdot 5!}{\Gamma(6-\alpha)}(x^{5-\alpha}+(1-x)^{5-\alpha}) - \frac{6!}{\Gamma(7-\alpha)}(x^{6-\alpha} \\ &+ (1-x)^{6-\alpha}))y^3(1-y)^3 \\ &+ \frac{3!}{\Gamma(4-\beta)}(y^{3-\beta}+(1-y)^{3-\beta}) - \frac{3\cdot 4!}{\Gamma(5-\beta)}(y^{4-\beta}+(1-y)^{4-\beta}) \\ &+ \frac{3\cdot 5!}{\Gamma(6-\alpha)}(y^{5-\beta}+(1-y)^{5-\beta}) - \frac{6!}{\Gamma(7-\beta)}(y^{6-\beta} \\ &+ (1-y)^{6-\beta}))x^3(1-x)^3, \end{aligned}$$

the analytical solution to this problem turns out to be known, and is

$$u(x,y,t) = e^{-t}x^3(1-x)^3y^3(1-y)^3, \tag{3.36}$$

see [44]. This choice is useful to *validate* our algorithm and test its performance. Then, we will be confident that the code is as good as in this case also when different sources, possibly reflecting specific problems of practical interest replace the forcing term above.

In Fig. 4.1, the effect of replacing ordinary derivatives $(\alpha,\beta,\gamma) = (2, 2, 1)$ with fractional derivatives, in a given diffusion equation, e.g., with $(\alpha,\beta,\gamma) = (1.1, 1.7, 1)$, thus accounting for *anomalous* diffusion, is clear. In general, such a modification implies new geometric patterns in the solution, and a possibly anisotropic behavior. Even a different speed of propagation, depending on the order of the fractional derivatives can be reproduced in this way. Indeed, all these features are observed, e.g., in certain porous media through which a fluid flows [36, 129].

In Fig. 3.2, the numerical errors $\|u^n - W^n_{FADI}\|$ are plotted in the L^2 and in the L^∞ norms, on an exponential scale (here, u^n denotes the exact solution to the FDE).

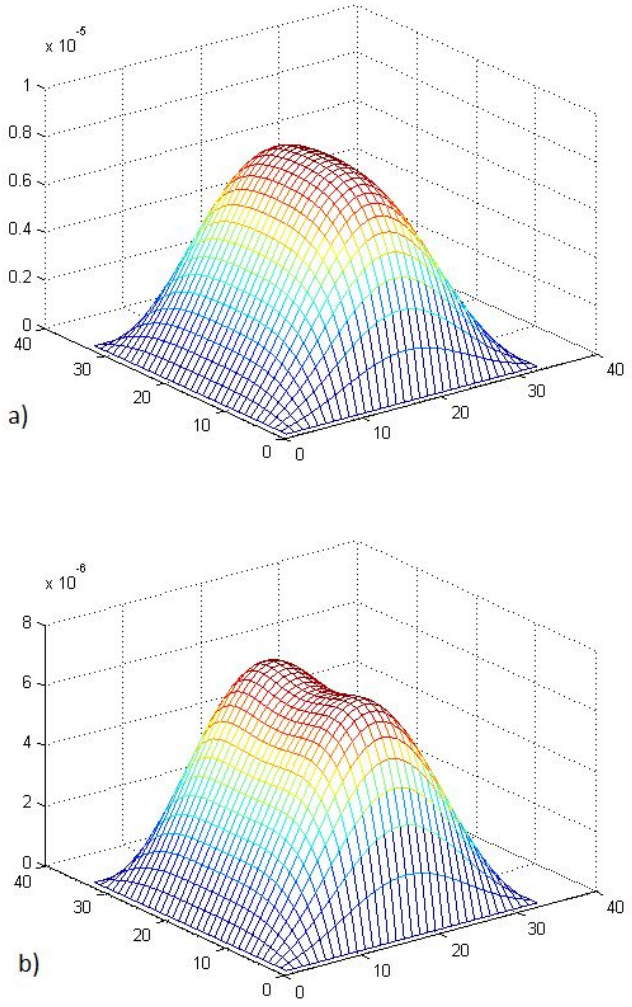

FIGURE 3.1: (a) Classical solution (obtained by a fine grid numerical ADI method with $\tau = h/16$), and (b) exact (analytical) fractional diffusion solution with $(\alpha, \beta, \gamma) = (1.1, 1.7, 1)$. The same forcing function was used in both cases.

Let define, for some $\theta > 0$, the *convergence rate*, say r, in the norm $(\cdot)$, for a given numerical method of order $p > 1$, as

$$r := \log_2 \frac{\|\tilde{u}_{\frac{h}{\theta}} - u^n\|_{(\cdot)}}{\|\tilde{u}_{\frac{h}{2\theta}} - u^n\|^p_{(\cdot)}},$$

where $\tilde{u}_{\frac{h}{\theta}}$ is the numerical solution computed with the discretization step $\tau = \frac{h}{\theta}$.

In Tables 3.1 and 3.2, the absolute numerical errors $||u^n - W^n||_\infty$ and $\|u^n - W^n\|_2$, as well as the corresponding convergence rates achieved using different space step sizes, are shown. Figure 3.2 shows the infinity norm and the L^2 norm errors for Example 1, when the compact difference scheme is implemented, at times $t = 1$, for several values of N, and fixed α, β, and γ.

(α,β,γ)	N	$\|u^n - W^n\|_\infty$	time-rate	space-rate	$\|\|u^n - W^n\|\|_2$	time-rate	space-rate
$(1.1, 1.7, 1)$	8	5.1772E-5	$--$	$--$	6.1824E-5	$--$	$--$
	16	2.3137E-6	2.93	1.92	1.1849E-6	2.89	1.87
	32	9.8789E-6	2.98	1.98	8.9028E-6	2.97	1.95
	64	5.1584E-7	2.99	1.99	3.2641E-6	2.99	1.98
	128	3.3393E-7	3.00	2.00	8.3237E-7	3.00	2.00
	256	5.0727E-12	3.00	2.00	1.1372E-11	3.00	2.00
	512	3.0138E-13	3.00	2.00	5.7824E-13	3.00	2.00
$(1.4, 1.4, 0.6)$	8	4.1884E-5	$--$	$--$	5.8541E-5	$--$	$--$
	16	3.7581E-6	2.91	1.89	1.7852E-6	2.91	1.89
	32	4.8541E-7	2.97	1.96	3.7851E-6	2.96	1.93
	64	5.7822E-7	2.99	1.98	5.7852E-7	2.98	1.98
	128	2.7851E-8	2.99	1.99	6.7852E-7	3.00	1.99
	256	4.0887E-12	3.00	2.00	2.7851E-10	3.00	2.00
	512	7.7852E-13	3.00	2.00	6.4852E-13	3.00	2.00
$(1.8, 1.8, 0.8)$	8	4.7885E-6	$--$	$--$	3.7852E-6	$--$	$--$
	16	6.7854E-6	2.95	1.94	3.8524E-7	2.91	1.89
	32	8.7852E-7	2.98	1.98	6.7852E-7	2.96	1.93
	64	1.8521E-8	3.00	1.99	6.7852E-8	2.98	1.98
	128	8.7852E-9	3.00	2.00	2.7852E-8	3.00	1.99
	256	3.7852E-12	3.00	2.00	6.8521E-11	3.00	2.00
	512	3.7852E-14	3.00	2.00	7.7852E-13	3.00	2.00
$(2.0, 2.0, 0.8)$	8	6.7852E-7	$--$	$--$	4.7852E-7	$--$	$--$
	16	6.7854E-7	2.97	1.98	1.1236E-8	2.96	1.95
	32	9.7453E-8	3.00	1.99	6.9173E-8	2.99	1.98
	64	6.7852E-9	3.00	2.00	5.7785E-9	3.00	2.00
	128	6.7852E-10	3.00	2.00	3.7852E-9	3.00	2.00
	256	4.8521E-12	3.00	2.00	6.8785E-12	3.00	2.00
	512	6.7852E-14	3.00	2.00	3.7185E-14	3.00	2.00

TABLE 3.1: L^∞ and L^2 norm errors and convergence rates for Example 1, when the (LOD) CN-WSGD scheme, that is a FADI method, is used, at time $t = 1$, for several values of (α, β, γ), N, and $\tau = h$.

(α,β)	N	$\|\|u^n - W^n\|\|_\infty$	time-rate	space-rate	$\|\|u^n - W^n\|\|_2$	time-rate	space-rate
$(1.1, 1.7)$	8	3.4852E-7	$--$	$--$	9.745252E-5	$--$	$--$
	16	2.4585E-8	2.95	1.92	7.7852E-5	2.93	1.87
	32	2.7852E-8	2.99	1.98	5.3255E-6	2.98	1.95
	64	4.8420E-9	3.00	1.99	4.7852E-6	2.99	1.98
	128	1.9651E-9	3.00	2.00	5.4525E-7	3.00	2.00
	256	4.4582E-11	3.00	2.00	3.6321E-10	3.00	2.00
	512	3.8512E-14	3.00	2.00	6.78521E-13	3.00	2.00

TABLE 3.2: L^∞ and L^2 norm errors, and convergence rates for Example 1, when the (LOD) CN-WSGD scheme, that is a FADI method, is used, at time $t = 5$, for several values of N, and $\tau = h$.

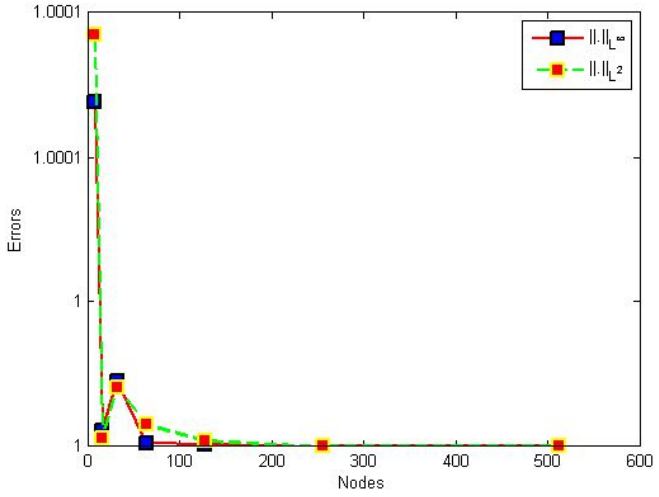

FIGURE 3.2: Absolute numerical error between the exact and the numerical solution of the FDE of Example 1, at $t = 1$, with $(\alpha, \beta, \gamma) = (1.1, 1.7, 1)$.

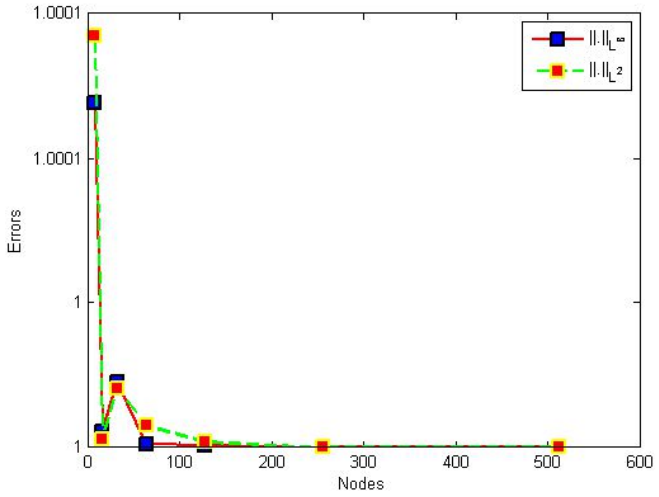

FIGURE 3.3: L^∞ and L^2 discrepancy between the numerical solution of the classical problem and that of the fractional problem with $(\alpha, \beta, \gamma) = (1.1, 1.7, 1)$, at $t = 1$.

In Fig. 3.3, the discrepancies between the numerical solutions to the classical and the fractional equations, obtained by the ADI and the $FADI$ method, respectively, i.e. $\|W^n_{ADI} - W^n_{FADI}\|$, are plotted in the L^2 and in the L^∞ norms, on an exponential scale. This is done to appreciate the differences when one switches from one to the other model. In Tables 3.3 and 3.4, such discrepancies $||W^n_{ADI} - W^n_{FADI}||_\infty$ and $\|W^n_{ADI} - W^n_{FADI}\|_2$, as well as the corresponding convergence rates, achieved using different space step sizes, are shown. Here, again, $W^n_i(\tau) := -\frac{1}{36}U^n_i(\tau) - \frac{1}{4}U^n_i(\frac{5}{36}\tau) + \frac{1}{6}U^n_i(\frac{\tau}{3})$ is the extrapolated solution, and U^n_i satisfies the compact scheme (3.32). Figure 3.3 shows the convergence rates of the maximum norm and the L^2 norm errors for Example 1, when the compact difference scheme is implemented, at times $t = 1$, for several values of N, and fixed α, β,

and γ.

(α, β, γ)	N	$\|W^n_{ADI} - W^n_{FADI}\|_\infty$	$\|\|W^n_{ADI} - W^n_{FADI}\|\|_2$
$(1.1, 1.7, 1)$	256	5.1025E-12	1.1295E-11
	512	3.0148E-13	5.6952E-13
$(1.4, 1.4, 0.6)$	256	4.0887E-12	2.4284E-10
	512	7.8152E-13	6.4985E-13
$(1.8, 1.8, 0.8)$	256	3.8025E-12	6.8521E-11
	512	3.7825E-14	7.7745E-13
$(2.0, 2.0, 0.8)$	256	4.8521E-12	6.7852E-12
	512	6.8658E-14	3.7758E-14

TABLE 3.3: L^∞ and L^2 norm discrepancy for Example 1, when the (LOD) CN-WSGD scheme, that is with a FADI method, is used, at time $t = 1$, for several values of N, and $\tau = h$.

(α, β, γ)	N	$\|\|W^n_{ADI} - W^n_{FADI}\|\|_\infty$	$\|\|W^n_{ADI} - W^n_{FADI}\|\|_2$
$(1.1, 1.7, 1)$	256	4.4582E-11	3.4852E-10
	512	3.8145E-14	6.6695E-13

TABLE 3.4: L^∞ and L^2 norm discrepancy for Example 1, when the (LOD) CN-WSGD scheme, that is with a FADI method, is used, at time $t = 5$, for several values of N, and $\tau = h$.

Example 2. Let choose $(\alpha,\beta,\gamma) = (1.1, 1.7, 1)$ in equation (3.35), and the source term

$$f(u) = \frac{u}{2},$$

instead of the source $f(x, y, t)$ defined above. This is a kind of forcing term often used for instance in modeling dissolution and precipitation in porous media [63, 143].

In Table 3.5, we show the corresponding numerical results obtained using the ADI and the FADI method for the FDE with $(\alpha,\beta, \gamma) = (1.1, 1.7, 1)$.

N	τ	$\|W^n_{ADI} - W^n_{FADI}\|_2$	time-rate	space-rate
16	$h/16$	$1.8512E-7$	$--$	$--$
32	$h/16$	$6.4852E-8$	3.00	2.00
64	$h/16$	$2.8962E-10$	3.00	2.00

TABLE 3.5: L^2 norm discrepancies, and convergence rates for Example 2, when the (LOD) CN-WSGD scheme (that is a FADI method) is used, at time $t = 1$, source term $f(u) = \frac{1}{2}u$, $(\alpha,\beta,\gamma) = (1.1, 1.7, 1)$ and several values of N.

Note that, in both examples, the numerical results shown in the tables provide an evidence of second-order accuracy in space and third-order in time, as claimed.

Consider now the following problem.

Example 3. Let be the fractional space-time diffusion equation

$$ {}_0D_t^{\gamma}u = {}_0D_x^{\alpha}u + {}_xD_1^{\alpha}u + {}_0D_y^{\beta}u + {}_yD_1^{\beta}u \tag{3.37} $$

on the space domain $\Omega := (0,1)\times(0,1)$, for $t>0$, subject to the boundary conditions.

$$ u(x,y,t) = 0, \quad (x,y)\in\partial\Omega, \quad t\in[0,1], $$

and to the initial value

$$ u(x,y,0) = e^{x+y}, \quad (x,y)\in[0,1]\times[0,1]. $$

If $(\alpha,\beta,\gamma) = (2,2,1)$, we obtain the classical diffusion equation

$$ u_t = \Delta u, \tag{3.38} $$

whose analytical solution, with the previous initial and boundary conditions, is $u(x,y,t) = e^{x+y+2t}$.

Consider now the analytical solution of (3.37) with $\alpha=\beta=2, \gamma=1$, the numerical obtained by the classical ADI method [51, 53], and a "close" fractional generalization of it, namely the case $\alpha=\beta=1.4, \gamma=0.6$, computed by its fractional (LOD) version.

In Fig. 3.4 the absolute numerical errors between the exact solution and the numerical solution obtained by a FADI method, in Fig. 3.5 the absolute numerical errors between the numerical classical ADI method and the numerical solution obtained by the FADI method, both are plotted in the L^2 norm.

The Table 3.6 show the convergence rates in time and space, of the L^2 norm errors in Example 3, when the compact difference scheme is implemented, at times $t=1$, for several values of τ and N, and fixed α and β. Here, u^n is the exact solution, while W^n_{ADI} is the solution obtained by classical ADI method, and W^n_{FADI} is that obtained by the FADI method.

As one would expect, in both, the classical ADI and the FADI methods, a faster convergence is observed as the grid is refined.

Example 4. If one wants to predict the fluid motion for a concentrate introduced at the left side of a tank, which advects rightwards and diffuses in the y direction, the fractional evolutionary advection-diffusion equation

$$ {}_0D_t^{\gamma}u = K_1({}_0D_x^{\alpha}u + {}_yD_1^{\alpha}u) + K_2({}_0D_y^{\beta}u + {}_yD_1^{\beta}u) + a_1\frac{\partial u}{\partial x} + a_2\frac{\partial u}{\partial 2}, \tag{3.39} $$

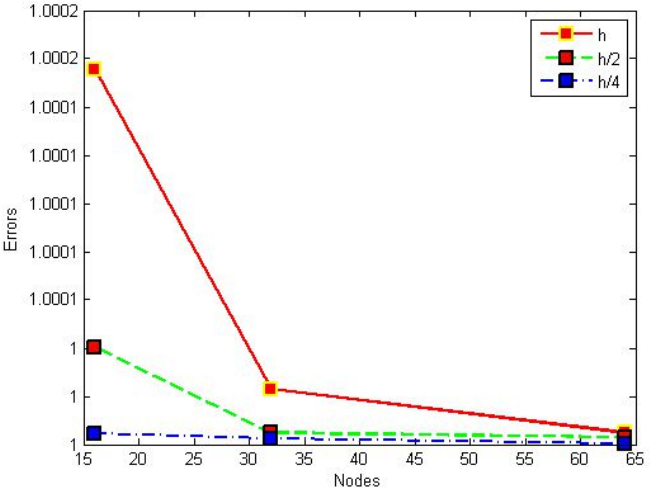

FIGURE 3.4: Numerical absolute errors $\|u^n - W^n_{FADI}\|$ for three values of the time and the space stepsizes, $\tau = h$ (solid-red), $\tau = h/2$ (dashed-green), $\tau = h/4$ (dashed and dotted-blue), for $\alpha = \beta = 1.4, \gamma = 0.6$, on an exponential scale.

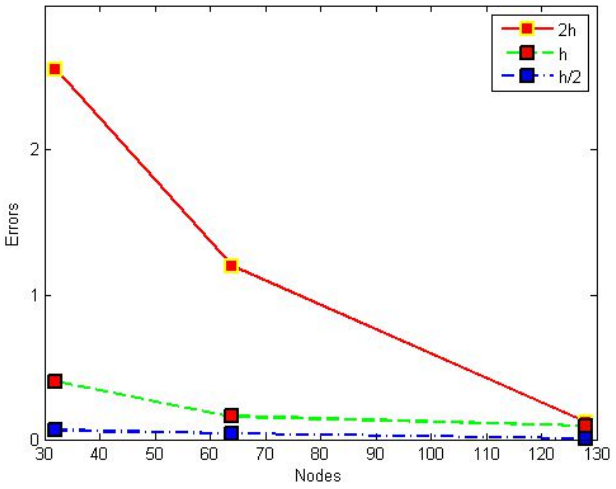

FIGURE 3.5: Discrepancy $\|W^n_{ADI} - W^n_{FADI}\|$ for three values of the time and the space stepsizes, $\tau = h$ (solid-red), $\tau = h/2$ (dashed-green), $\tau = h/4$ (dashed and dotted-blue), for $\alpha = \beta = 1.4, \gamma = 0.6$, on an exponential scale.

can be solved, for suitable values of the fractional orders, α, β, and γ. An example of a real case, where a dye is continuously introduced into a tank filled in with a fluid, is shown in Fig. 3.6. Here, the diffusion occurring in the direction of the y-axis is due to turbulence. Classical diffusion yields a parabolic profile, while in the picture the flow described by anomalous diffusion looks conic (in 3D) [38, 110].

Let consider the space domain $\Omega := [0, 10] \times [0, 10]$, for $t \in [0, 10]$. We impose the boundary condition

$$u(10, y, t) = u_y(x, 0, t) = u_y(x, 10, t) = 0, \tag{3.40}$$

N	τ	$\|u^n - W^n_{FADI}\|_2$	$\|W^n_{ADI} - W^n_{FADI}\|_2$	time-rate$(u^n - W^n_{FADI})$	space-rate$(u^n - W^n_{FADI})$
16	h	$6.7601E-5$	$2.5656E-4$	$--$	$--$
32	h	$1.0083E-5$	$1.2561E-4$	$--$	$--$
64	h	$2.1252E-6$	$1.2634E-5$	$--$	$--$
16	$h/2$	$1.7516E-5$	$1.0226E-5$	$2.92-2.89$	$2.00-1.98$
32	$h/2$	$2.1312E-6$	$1.5877E-5$	$2.99-2.97$	$2.00-1.99$
64	$h/2$	$1.3699E-6$	$9.5152E-6$	$3.00-2.99$	$2.00-2.00$
16	$h/4$	$2.0125E-6$	$6.2565E-6$	$2.98-2.96$	$2.00-1.99$
32	$h/4$	$1.1075E-6$	$4.1485E-6$	$3.00-2.99$	$2.00-2.00$
64	$h/4$	$1.8248E-7$	$8.6635E-7$	$3.00-3.00$	$2.00-2.00$

TABLE 3.6: L^2 norm errors, discrepancies and convergence rates for Example 3, when the (LOD) CN-WSGD (that is a FADI) scheme is used, with $\alpha = \beta = 1.4, \gamma = 0.6$, at time $t = 1$, for several values of N and τ.

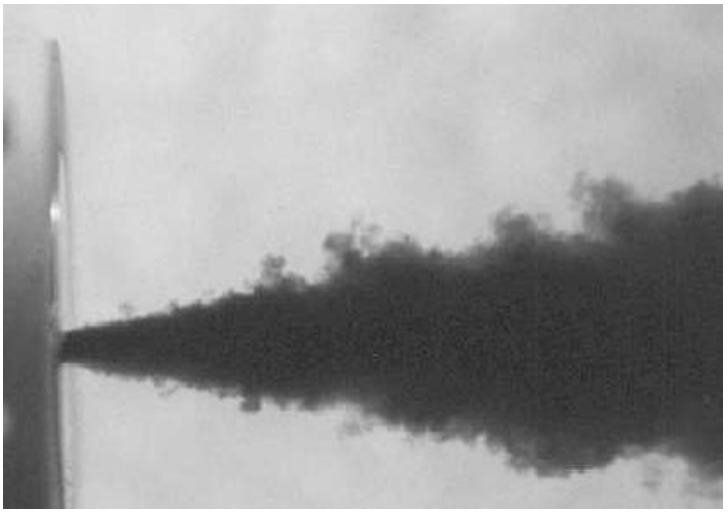

FIGURE 3.6: Example of an advecting plume, which is better described through anomalous rather than classical diffusion (from [38]).

and the initial condition

$$u(0, y, 0) = \delta(y - 5). \tag{3.41}$$

What we are doing is setting the boundary back to its initial value each time step. We do not want to allow to pass through the horizontal walls of the tank so the boundary conditions for both y axis are set to zero.

Setting then $K_1 = a_2 = 0$, we obtain

$$ {}_0D_t^\gamma u = K_2({}_0D_y^\beta u + {}_yD_1^\beta u) + a_1 \frac{\partial u}{\partial x}. \tag{3.42}$$

Note that, if $(\alpha, \beta, \gamma) = (2, 2, 1)$, $K_1 = K_2 = 1$, and $\mathbf{a} = (a_1, a_2)$, we obtain the classical advection-diffusion equation

$$u_t = \Delta u + \mathbf{a} \cdot \nabla u, \tag{3.43}$$

Consider now the solution of equation (3.42) by the classical ADI method ($\alpha = \beta =$

$2, \gamma = 1$) [51, 53], and its fractional generalization (the LOD version). We will consider for the fractional exponents the values $\alpha = \beta = 1.6, \gamma = 0.6$. In Fig. 3.7, the numerical results are shown for the advection-diffusion plume, corresponding to the parameters $K_2 = 1$, $a_1 = 2$, $\Delta x = \Delta y = 1/20$, $\Delta t = 0.002$.

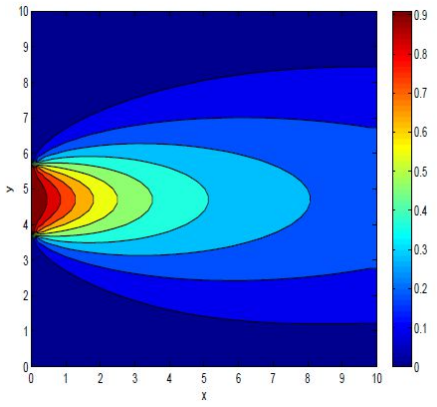

FIGURE 3.7: Numerical results for advection-diffusion plume, $\alpha = \beta = 1.6, \gamma = 0.6$ (from [38]).

Chapter 4

An ADI method for the numerical solution of 3D fractional diffusion equations in porous media

In 1998, J. He [71] proposed a generalized version of the Darcy's law, containing *fractional* Riemann-Liouville derivatives,

$$\mathbf{q} = -\mathbf{K}\,\nabla^{\alpha} p, \qquad \nabla^{\alpha} := \left(\frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}}, \frac{\partial^{\alpha_2}}{\partial y^{\alpha_2}}, \frac{\partial^{\alpha_3}}{\partial z^{\alpha_3}} \right),$$

where $p \equiv p(x, y, z, t)$ is the pressure, $\mathbf{q} = (q_x, q_y, q_z)$ is the fluid velocity, $\mathbf{K} = (K_x, K_y, K_z)$ is the percolation tensor (here assumed to be diagonal), and $\alpha := (\alpha_1, \alpha_2, \alpha_3)$.

Therefore, the more general equation for seepage flow with fractional Riemann-Liouville derivatives proposed by J. He [71] is

$$\begin{aligned} \frac{1}{\nu}\frac{\partial p}{\partial t} \quad &= \frac{\partial^{\beta_1}}{\partial x^{\beta_1}}\left(K_x \frac{\partial^{\alpha_1} p}{\partial x^{\alpha_1}} \right) + \frac{\partial^{\beta_2}}{\partial y^{\beta_2}}\left(K_y \frac{\partial^{\alpha_2} p}{\partial y^{\alpha_2}} \right) \\ &\quad + \frac{\partial^{\beta_3}}{\partial z^{\beta_3}}\left(K_z \frac{\partial^{\alpha_3} p}{\partial z^{\alpha_3}} \right) + f(p; x, y, z, t), \qquad (x, y, z) \in \Omega. \end{aligned} \tag{4.1}$$

Here, $\frac{1}{\nu}$ is the specific storage coefficient (assumed to be constant), $f(p; x, y, z, t)$ is a source or sink term, Ω is a given open bounded domain of $\mathbb{R}^3$, where the percolation takes place, t is time, and $0 < \beta_i < 1$, $0 < \alpha_i \leq 1$, and $1 < \beta_i + \alpha_i \leq 2$, for $i = 1, 2, 3$, see [88]. Moreover, the boundary condition

$$p(x, y, z, t)|_{\partial\Omega} = \Phi(x, y, z, t), \tag{4.2}$$

as well as the initial condition

$$p(x, y, z, 0) = \varphi(x, y, z). \tag{4.3}$$

are imposed, with suitable functions Φ and φ.

Given the interest for models based on fractional partial differential equations (fPDEs), and recalling, needless to say, that analytic solutions to most fPDEs are usually not available, a number of authors have proposed over the years a few numerical methods to solve fPDEs, see, e.g., [26, 28, 29, 69, 83–87, 100, 102, 117, 121, 122, 135, 150–153]. However, in the existing literature, numerical methods capable to handle higher-dimensional fPDEs seem to be rather few, not to mention their performance. It should be observed that, being the fractional derivatives *nonlocal* operators, discretizing fPDEs can be expected to be more demanding, e.g., in terms of memory storage and CPU time, compared to classical PDEs. All this calls for devising effective numerical methods to solve high-dimensional fPDE problems.

In this Thesis, we will consider the 3D fractional seepage flow equation in (4.1), on the bounded domain, $\Omega := (0, L_x) \times (0, L_y) \times (0, L_z)$, on the time interval $(0, T]$. Moreover, we assume that such an initial-boundary value problem equation has a unique, sufficiently smooth solution, when the Dirichlet boundary conditions

$$\begin{aligned} p(0, y, z, t) &= \Phi_1(y, z, t) = 0, & p(L_x, y, z, t) &= \Phi_2(y, z, t) \\ p(x, 0, z, t) &= \Phi_3(x, z, t) = 0, & p(x, L_y, z, t) &= \Phi_4(x, z, t) \\ p(x, y, 0, t) &= \Phi_5(x, y, t) = 0, & p(x, y, L_z, t) &= \Phi_6(x, y, t), \end{aligned} \tag{4.4}$$

and the initial condition

$$p(x, y, z, 0) = \varphi(x, y, z),$$

are imposed, for some given functions Φ_i, $i = 1, \ldots, 6$, and φ. The operator $\frac{\partial^{\alpha_1}}{\partial x^{\alpha_1}}$ is the fractional Riemann-Liouville derivative of order α_1 with respect to x.

A new well "balanced", fractional version of the Alternating Direction Implicit (ADI) method is introduced to solve numerically 3D diffusion and reaction-diffusion fPDEs, which can be applied, e.g., to describe fluid flows through porous media. We term "balanced" our scheme since we distribute the right-hand side in the ADI scheme in equal amount among the thre equations representing the three steps steps of the ADI method, instead of using the full right-hand side in the first equation only, see subsection 2.1 below. This choice allows us to increase the space accuracy from the first to the second order. On the other hand, Liu et al [88] derived a scheme accurate to the second order in space resorting to Richardson extrapolation.

Our method is shown to be *unconditionally stable* for every fractional order of space derivatives, is *second-order* accurate *in space* (as stated here above), and *third-order* accurate *in time*. This latter property is here established resorting to an extrapolation technique along with an optimization realized through Google's *PageRank* algorithm. The method followed here to do this is similar to that used for the 2D case in [33]. Liu et al.'s method is only of the second order in time.

The for the 3-D cases we set up the typical problems, and discuss some fractional versions of the ADI (fADI) 3D schemes, we establish a convergence result, and some details about optimization and extrapolation are given, and a few numerical examples are presented.

4.1 Fractional diffusion and seepage flow in homogeneous media

To construct our fADI scheme on a bounded domain, we first discretize space and time, as usual, setting $x_i := ih_x$, for $i = 0, 1, 2, \ldots, M_x$, with $h_x := L_x/M_x$; $y_j := jh_y$, for $j = 0, 1, 2, \ldots, M_y$, with $h_y := L_y/M_y$; $z_k := kh_z$, for $k = 0, 1, 2, \ldots, M_z$, with $h_z := L_z/M_z$; and $t_n := n\tau$, for $n = 0, 1, 2, \ldots, N$, $\tau := T/N$. Here, M_x, M_y, M_z, and N are positive integers, while h_x, h_y, h_z, and τ are the space and time step sizes, respectively. The numerical approximation to $p(x_i, y_j, z_k, t_n)$ provided by the scheme at such points will be then denoted by $p^n_{i,j,k}$. Similarly, we write $f^n_{i,j,k}$ to denote an approximation of $f(p(x_i, y_j, z_k, t_n); x_i, y_j, z_k, t_n)$ for the source term, $p^0_{i,j,k} = \varphi_{i,j,k}$ for $\varphi(x_i, y_j, z_k, 0)$, the initial value, and $p^n_{0,j,k} = \Phi_1(y_j, z_k, t_n) = 0, p^n_{M_x,j,k} = \Phi_2(y_j, z_k, t_n) = 0$, $p^n_{i,0,k} = \Phi_3(x_i, z_k, t_n) = 0$, $p^n_{i,M_y,k} = \Phi_4(x_i, z_k, t_n)$, $p^n_{i,j,0} = \Phi_5(x_i, y_j, t_n) = 0$, and $p^n_{i,j,M_z} = \Phi_6(x_i, y_j, t_n)$, for the boundary conditions.

We discretize the fPDE, approximating the first-order derivative $\frac{\partial p}{\partial t}$ in (3.1) by forward finite differences. Assuming that the solution, p, has a first-order continuous space derivative and that its second-order space derivatives are integrable, the operators $\frac{\partial^{\gamma_1}}{\partial x^{\gamma_1}}$, $\frac{\partial^{\gamma_2}}{\partial y^{\gamma_2}}$, and $\frac{\partial^{\gamma_3}}{\partial z^{\gamma_3}}$ in (3.1) can be discretized using the shifted fractional Grünwald-Letnikov derivative in (4). Thus we obtain the implicit finite-difference scheme

$$\begin{aligned}\frac{p^{n+1}_{i,j,k} - p^n_{i,j,k}}{\tau} &= \frac{K_x}{h_x^{\gamma_1}} \sum_{s=0}^{i+1} g_{\gamma_1,s} p^{n+1}_{i+1-s,j,k} + \frac{K_y}{h_y^{\gamma_2}} \sum_{s=0}^{j+1} g_{\gamma_2,s} p^{n+1}_{i,j+1-s,k} \\ &\quad + \frac{K_z}{h_z^{\gamma_3}} \sum_{s=0}^{k+1} g_{\gamma_3,s} p^{n+1}_{i,j,k+1-s} + f^{n+1}_{i,j,k}.\end{aligned} \tag{4.5}$$

It is known that the fractional discrete operator

$$\delta_x^{\gamma_1} p^{n+1}_{i,j,k} := \frac{1}{h_x^{\gamma_1}} \sum_{s=0}^{i+1} g_{\gamma_1,s} p^{n+1}_{i+1,j,k}, \tag{4.6}$$

provides an $\mathcal{O}(h_x)$ approximation to the Grünwald-Letnikov shifted fractional derivative of order γ_1 [100]. Similarly, the discrete operators

$$\delta_y^{\gamma_2} p_{i,j,k}^{n+1} := \frac{1}{h_y^{\gamma_2}} \sum_{s=0}^{j+1} g_{\gamma_2,s} p_{i,j+1,k}^{n+1}, \quad \delta_z^{\gamma_3} p_{i,j,k}^{n+1} := \frac{1}{h_z^{\gamma_3}} \sum_{s=0}^{k+1} g_{\gamma_3,s} p_{i,j,k+1}^{n+1}, \tag{4.7}$$

provide, respectively, $\mathcal{O}(h_y)$ and $\mathcal{O}(h_z)$ approximations to the Grünwald-Letnikov shifted fractional derivatives of order γ_2 and γ_3. Using such operators, the implicit difference scheme in (4.5) can be rewritten in a more compact form as

$$(1 - K_x \tau \delta_x^{\gamma_1} - K_y \tau \delta_y^{\gamma_2} - K_z \tau \delta_z^{\gamma_3})\, p_{i,1,k}^{n+1} = p_{i,j,k}^{n} + \tau f_{i,j,k}^{n}, \tag{4.8}$$

Note that, evaluating $f_{i,j,k}^n$ on the right-hand side at time t_n, makes it more tractable the case when f depends on p. In Section 4.2, we will show that this scheme is characterized by a *local truncation error* of the order $\mathcal{O}(\tau) + \mathcal{O}(h_x) + \mathcal{O}(h_y) + \mathcal{O}(h_z)$, and is *unconditionally stable*. However, adopting suitable strategies, we have been able to improve such orders attaining the order $\mathcal{O}(\tau) + \mathcal{O}(h_x^2) + \mathcal{O}(h_y^2) + \mathcal{O}(h_z^2)$.

Equation (4.8) (or (4.5)) yields a *linear* system of equations satisfied by $p_{i,j,k}^{n+1}$. The corresponding system's matrix, however, is *neither* sparse *nor* band structured, as it happens in the corresponding classical ADI method. This implies that the scheme in (4.8) requires, at each step, the solution of a *large dense* linear system of equations. Therefore, we are facing a computationally demanding numerical problem. This calls for constructing suitable efficient numerical schemes, possibly unconditionally stable.

To this purpose, we exploit the idea of the classical ADI method to design an implicit difference scheme for each direction, but in the framework of *fractional* differential equations. The aim is to split the computations into three steps, each requiring a reduced computational load. On the first step, we solve the problem in the x-direction, on the second one we solve it in the y-direction, and in the third one in the z-direction.

To devise such a splitting, we add an extra higher-order term to the left-hand side of (4.8), so that we are able to factor the operator into three factors,

$$\begin{aligned}(K_x K_y \tau^2 \delta_x^{\gamma_1} \delta_y^{\gamma_2} + K_x K_z \tau^2 \delta_x^{\gamma_1} \delta_z^{\gamma_3} \\ + K_y K_z \tau^2 \delta_y^{\gamma_2} \delta_z^{\gamma_3} - K_x K_y K_z \tau^3 \delta_x^{\gamma_1} \delta_y^{\gamma_2} \delta_z^{\gamma_3})\, p_{i,j,k}^{n+1},\end{aligned} \tag{4.9}$$

into three factors, without affecting the overall convergence rate, thus obtaining the scheme

$$(1 - K_x \tau \delta_x^{\gamma_1})(1 - K_y \tau \delta_y^{\gamma_2})(1 - K_z \tau \delta_z^{\gamma_3})\, p_{i,j,k}^{n+1} = p_{i,j,k}^{n} + \tau f_{i,j,k}^{n}. \tag{4.10}$$

Splitting now (4.10) in the three dimensions, we obtain the *unbalanced* version scheme which provides the solution at time t_{n+1},

$$\begin{aligned}
(1 - K_x\tau\delta_x^{\gamma_1})\, p_{i,j,k}^{n+1/3} &= p_{i,j,k}^{n} + \tau f_{i,j,k}^{n},\\
(1 - K_y\tau\delta_y^{\gamma_2})\, p_{i,j,k}^{n+2/3} &= p_{i,j,k}^{n+1/3},\\
(1 - K_z\tau\delta_z^{\gamma_3})\, p_{i,j,k}^{n+1} &= p_{i,j,k}^{n+2/3},
\end{aligned} \tag{4.11}$$

which ends up with the solution at time t_{n+1} and turns out to be of the first order in space [88]. We can however choose to distribute the effect of the source term among the three steps in a more *balanced* way, obtaining from (4.10) the fADI scheme

$$\begin{aligned}
(1 - K_x\tau\delta_x^{\gamma_1})\, p_{i,j,k}^{n+1/3} &= p_{i,j,k}^{n} + \tfrac{\tau}{3}\, f_{i,j,k}^{n},\\
(1 - K_y\tau\delta_y^{\gamma_2})\, p_{i,j,k}^{n+2/3} &= p_{i,j,k}^{n+1/3} + \tfrac{\tau}{3}\, f_{i,j,k}^{n},\\
(1 - K_z\tau\delta_z^{\gamma_3})\, p_{i,j,k}^{n+1} &= p_{i,j,k}^{n+2/3} + \tfrac{\tau}{3}\, f_{i,j,k}^{n}.
\end{aligned} \tag{4.12}$$

which also provide the solution at time t_{n+1}. The algorithm (4.12) is of the second-order in space and of the third-order in time. These facts will be established in Section 4.2 below.

As it will be clear in the Examples below, it is noteworthy that the "balanced" algorithm wins over the previous one, where the entire forcing term, $\tau\, f_{i,j,k}^{n}$, affects only the first equation in (4.12). The balanced scheme, instead, distributes an equal fraction of the forcing term, i.e., one third of it, among the three equations. Of course, equally unbalanced (and less performing) schemes are obtained if the full forcing term is introduced on the right-hand side of the second and third equations in (4.12). The "unbalanced" scheme is accurate to the *first* order in space and requires a little shorter computing time, compared to the better "balanced" scheme. The latter however is accurate to the *second* order in space, see Theorem 3.1 below.

Therefore, it seems that using a more balanced scheme, which equally affects all (the three) directions, results in a better method. This algorithm seems to be new, even within the framework of three-dimensional fractional ADI methods.

Thus, the computations are split into three (time) fractional steps, in each of which a one-dimensional equation is solved at time, as follows:

Step 1. Solve the problem in the x-direction (for each fixed pair (y_j, z_k)), to obtain the intermediate value of the solution, say $p_{i,j,k}^{n+1/3}$, from the first equation in (4.12).

Step 2. Solve the problem in the y-direction (for each fixed pair (x_i, z_k)), to obtain the intermediate value $p_{i,j,k}^{n+2/3}$, from the second equation in (4.12), using the results obtained at Step 1.

Step 3. Solve the problem in the z-direction (for each fixed pair (x_i, y_j)), from the third equation in (4.12), using the results of Step 2.

Taking into account the boundary values $p_{0,j,k}^{n+1/3}$ and $p_{M_x,j,k}^{n+2/3}$, we can construct the coefficients of the matrix $A := \{a_{s,t}\}$ of the linear system in (4.12): for each fixed (j,k), we have

$$a_{s,t} = \begin{cases} 0, & t \geq s+2, s = 1, 2, \ldots, M_x - 3, \\ -K_x \frac{\tau}{h_x^{\gamma_1}} g_{\gamma_1,0}, & t = s+1, s = 2, ..., M_x - 2, \\ 1 - K_x \frac{\tau}{h_x^{\gamma_1}} g_{\gamma_1,1}, & t = s = 2, \ldots, M_x - 1, \\ -K_x \frac{\tau}{h_x^{\gamma_1}} g_{\gamma_1,s-t+1}, & t \leq s-1, s = 2, \ldots, M_x - 1. \end{cases} \tag{4.13}$$

Then, using the boundary values $p_{i,0,k}^{n+2/3}$ and $p_{i,M_y,k}^{n+2/3}$, we obtain the matrix $B := \{b_{s,t}\}$ of the linear system in the second equation of (4.12). For each fixed (i,k), such a matrix turns out to be similar to A (given in (4.13)). Finally, using the boundary values $p_{i,j,0}^{n+1}$ and p_{i,j,M_z}^{n+1}, we obtain the matrix $C := \{c_{s,t}\}$ of the linear system in the third equation in (4.12), which, again, for each fixed (i,j), is similar to A.

Of course, as in the classical ADI *Peaceman-Rachford* algorithm [109], used to solve "integer-order" (i.e., non-fractional) PDEs, it is necessary to provide certain boundary values to be associated to the diffusion in the x-direction, that is $p_{0,j,k}^{n+1/3}$ and $p_{M_x,j,k}^{n+1/3}$, as well as $p_{i,0,k}^{n+2/3}$ and $p_{i,M_y,k}^{n+2/3}$, associated to the diffusion along y, and similarly for the diffusion along z, when solving the systems with the coefficient matrices A, B, and C. For instance, we need the boundary values $p_{0,j,k}^{n+1/3}$, $p_{M_x,j,k}^{n+1/3}$, which can be obtained from

$$p_{i,j,k}^{n+1/3} = (1 - K_y \tau \delta_y^{\gamma_2})(1 - K_z \tau \delta_z^{\gamma_3})\, p_{i,j,k}^{n+1}, \quad i = 0, \ldots, M_x,$$

for $j = 1, 2, \ldots, M_y - 1$, $k = 1, 2, \ldots, M_z - 1$, $n = 0, 1, \ldots, N-1$, and $p_{i,0,k}^{n+2/3}$, $p_{i,M_y,k}^{n+2/3}$, which can be obtained from

$$p_{i,j,k}^{n+2/3} = (1 - K_z \tau \delta_z^{\gamma_3})\, p_{i,j,k}^{n+1}, \quad i = 0, \ldots, M_y,$$

for $i = 1, 2, \ldots, M_x - 1$, $k = 1, 2, \ldots, M_z - 1$, $n = 0, 1, \ldots, N-1$.

Examining the three coefficient matrices A, B, C, above, it can be seen that, at each time step, it is merely required to solve, for each fixed pair (j,k) (at each x-level) a linear upper *triangular* system of size $M_x - 1$, *or*, for each fixed pair (i,k) (at each y-level), a linear upper triangular system of size $M_y - 1$, *or*, for each fixed pair (i,k) (at each z-level) a linear upper triangular system of size $M_z - 1$.

4.2 Theoretical analysis of the 3D fADI algorithm

As mentioned in Section 4.1, the fractional discrete operators in (4.6), (4.7) provide an $\mathcal{O}(h_x)$, $\mathcal{O}(h_y)$, and $\mathcal{O}(h_z)$ approximation to the Grünwald-Letnikov shifted fractional derivative of order γ_1, γ_2, and γ_3, respectively, see [100, 132].

The "unbalanced" implicit difference scheme (4.11), is characterized by a *local truncation error* of order $\mathcal{O}(\tau)+\mathcal{O}(h_x)+\mathcal{O}(h_y)+\mathcal{O}(h_z)$, and is *unconditionally stable*. These facts were established in [123]. Let us now consider our better "balanced" scheme, (4.12). We will prove the following

Theorem 7. The scheme in (4.12) is unconditionally stable, third-order accurate in time, and second-order accurate in space, for every choice of the (space) fractional orders, γ_i, $i=1,2,3$.

Proof. In [144], the unbalanced scheme in (4.11) was analyzed and its numerical stability proved. Consider now our balanced scheme in (4.12). Here we use the so called "Lubich second-order backward finite difference" [124], which is defined, at time n, in the x-direction, at the point x_{i+1}, and for fixed y and z, as

$$ {}^{L}\delta_x^{\gamma_1} p\Big|_{x_{i+1}} := \frac{1}{h_x^{\gamma_1}} \sum_c \omega_c^{\gamma_1} p^n_{i-c,j,k}. \tag{4.14} $$

Here, $c\in\{-1,0,-1\}$, γ_1 represents the fractional order derivative, and the $\omega_c^{\gamma_1}$'s are the first three coefficients of the Taylor expansions of the corresponding *generating function*, say $f_a(z)$, see [124]. Here, a denotes the order of accuracy, and for $a=2$ such function is given by

$$ f_2^{\gamma_1}(z) := \left(\frac{3}{2}-2z+\frac{1}{2}z^2\right)^{\gamma_1}. \tag{4.15} $$

The Grünwald-Letnikov definition of the discretized Riemann-Liouville fractional derivative differs from that given by Lubich by an $\mathcal{O}(h_x^2)$ term, see [124]. Thus, evaluating the fractional Riemann-Liouville derivatives at the point x_{i+1}, to approximate the space fractional derivative in the sense of *Lubich* of order γ_1 in the x-direction, we obtain

$$ \delta_x^{\gamma_1} p\Big|_{x_{i+1}} = \frac{3p^{n+1/3}_{i+1,j,k} - 4p^{n+1/3}_{i,j,k} + p^{n+1/3}_{i-1,j,k}}{2\tau^{\gamma_1}} + \mathcal{O}(h_x^2), \tag{4.16} $$

where $\tau = h_x$. To be more general, we could set $\tau := s\,h_x$, for some $s>0$, but this would not entail any loss of generality because the stability analysis would not affect the location of the roots by this. This holds for each fixed i and n,j,k. Using the

approximation (4.16) in (4.12), we obtain

$$p^{n+1/3}_{i+1,j,k} - K_x\tau\frac{3p^{n+1/3}_{i+1,j,k} - 4p^{n+1/3}_{i,j,k} + p^{n+1/3}_{i-1,j,k}}{2\tau^{\gamma_1}} = p^{n}_{i-1,j,k} + \frac{\tau}{3}f^{n}_{i,j,k}, \tag{4.17}$$

with a truncation error of order $\mathcal{O}(h_x^2)$. We now approximate the x-space operator applied to $p^{n+1/3}_{i+1,j,k}$ [144] by a second-order Taylor expansion, obtaining

$$1 - K_x\,\tau\,\delta_x^{\gamma_1} = L_x + \frac{4}{3}h_x\partial_x - \frac{4}{9}h_x^2 D_x \tag{4.18}$$

where $D_x := \partial_{xx}$ and $L_x := 1 + \theta h_x D_x$, $0 < \theta < 1$. Then, we can replace the $\mathcal{O}(h_x^3)$ with the explicit form of $p^{n+1/3}_{i+1,j,k}$ given by the extrapolation formula

$$\begin{aligned} \tilde{p}^{n+1/3}_{i+1,j,k} : &= \frac{147}{2500}\,p^{n+1/3}_{i+1,j,k} - \frac{29409}{100000}\,p^{n+**}_{i+i,j,k} - \frac{147}{2500}\,p^{n+*}_{i+1,j,k} + \frac{147}{2500}\,p^{n}_{i+1,j,k} \\ &= p^{n+1/3}_{i+1,j,k} + \mathcal{O}(h_x^2) \end{aligned} \tag{4.19}$$

(see (4.24) in Section 4.1 below, and (4.27) in Section 4.2), where $*$ and $**$ represent respectively the first and the second intermediate sub-steps between n and $n + 1/3$, and $p^{n+1/3}_{i+1,j,k}$ is implicitly defined through (4.17). We can then insert L_x in (4.18) $\tilde{p}^{n+1/3}_{i+1,j,k}$, obtaining

$$L_x p^{n+1/3}_{i+1,j,k} = (1 - K_x\tau)\tilde{p}^{n+1/3}_{i+1,j,k} - \frac{4}{3}\tau\partial_x\tilde{p}^{n+1/3}_{i+1,j,k} + \frac{4}{9}\tau^2 D_x\tilde{p}^{n+1/3}_{i+1,j,k}. \tag{4.20}$$

Subtracting $L_x\tilde{p}^{n+1/3}_{i+1,j,k}$ from both sides of (4.20), we obtain finally

$$\begin{aligned} L_x v \;=\; & -(K_x\tau + \theta h_x\delta_x^{\gamma_1})\tilde{p}^{n+1/3}_{i+1,j,k} - (1 + \frac{4}{3}\tau\partial_x + \theta h_x\delta_x^{\gamma_1})\tilde{p}^{n+1/3}_{i+1,j,k} \\ & -(1 + \theta h_x + \frac{4}{9}\tau^2)\delta_x^{\gamma_1}\tilde{p}^{n+1/3}_{i+1,j,k}, \end{aligned} \tag{4.21}$$

where we set, for short, $v := p^{n+1/3}_{i+1,j,k} - \tilde{p}^{n+1/3}_{i+1,j,k}$, and replaced the classical second order derivative D_x with his fractional version, $\delta_x^{\gamma_1}$. A von Neumann stability analysis of (4.21) yields an equation for the amplification factor, σ,

$$(\theta h_x - K_x\tau)\sigma^4(\sigma - 1)^2 = \frac{7}{3}(1 + K_x\theta\tau)\sigma^3(\sigma - 1) - \frac{8}{3}\tau\sigma^2(2\sigma - 1). \tag{4.22}$$

Similar results can be obtained for the y and the z directions. It can be proved that the roots of (4.22) are in the range $0 \le |\sigma| \le 1$, for every value of K_x, K_y, K_z, θ, h_x and τ [142]. Consequently, the scheme in (4.21) turns out to be of the second order in space, and unconditionally stable. The third-order accuracy in time can be established proceeding as in the 2D case worked out in [33]. □

4.3 The numerical implementation and some examples

In this section, we apply our numerical method to solve 3D fractional diffusion as well as reaction-diffusion problems. Examples, including reaction-diffusions, are then given to illustrate the performance of the algorithm.

4.3.1 Some theoretical preliminary considerations

We set up an extrapolation technique to *optimize* our algorithm, thus attaining an appreciable *acceleration*. This can be accomplished exploiting the *PageRank* algorithm, widely used by Computer Scientists, but not so much in Numerical Analysis.

The Google's *PageRank* algorithm [78] makes it possible to index material in Internet, using the degree of popularity of a given web page, to define the position in searching results through the computation of certain optimal coefficients. In this way, this algorithm provides for instance an optimization method to make it faster a given web search. This approach will be associated to an *extrapolation* technique, to determine the coefficients which make it optimal. Not only a considerable amount of CPU time can be saved in this way, [26], but this procedure also increases the accuracy of the overall method up to the *third order* in time, see [33, 97]. Let us describe such a technique.

Step 1. Compute first the array $\mathbf{\Psi} = (\psi_1, \psi_2, \psi_3, \psi_4)^T$, solving the system

$$\mathbf{\Psi} = \mathbf{v} + d\,\mathbf{L}\,\mathbf{\Psi}, \tag{4.23}$$

of four linear algebraic equations, where d is the so-called *damping factor*, in general assumed to be around 0.85 (see [16]; such a numerical value for d was determined by empirical trials), $\mathbf{v} := (1-d)\,\left(0, (N-2)^{-1}, (N-1)^{-1}, N^{-1}\right)^T$, N being the dimension of the problem, which in our case is equal to 4, and $\mathbf{L} = \{\ell_{ij}\}$ for $i, j = 1, \ldots, 4$, where l_{ij} are certain nonnegative coefficients such that, for each j, $\sum_{i=1}^{N} l_{i,j} = 1$. Note that (4.23) is an implicit but linear equation for Ψ, hence immediately solvable. An *extrapolated solution*, depending on $p^n \equiv p^n_{i,j,k}$, is then used to solve the problem. The quantity p^n requires evaluating certain coefficients, which can be obtained by the "*PageRank* accelerating algorithm" [78]. The previous algebraic system yields the optimal coefficients array $\mathbf{\Psi}$.

Step 2. Compute the solution p^n to a *compact difference scheme* [125, 131, 134, 153] (see step 3 below), with the four time step sizes τ, $\frac{3}{4}\tau$, $\frac{\tau}{2}$, and $\frac{\tau}{4}$ [131]. This kind of methods is usually adopted to treat steady convection-diffusion numerical problems on uniform grids [125], rather than time-dependent problems.

Step 3. Evaluate the *extrapolated solution*, $q^n(\tau)$, by

$$q^n(\tau) = \psi_1\, p^n(\tau) + \psi_2\, p^n\left(\frac{3}{4}\tau\right) + \psi_3\, p^n\left(\frac{\tau}{2}\right) + \psi_4\, p^n\left(\frac{\tau}{4}\right), \tag{4.24}$$

where the dependence of p^n on τ has been displayed.

4.3.2 Numerical examples

In this section, we present a few numerical results to support the theoretical analysis developed in the previous sections. In the following, the symbol ${}_aD_b^\alpha p$ will be used to define the fractional Riemann-Liouville derivative of p of order α, in the interval $[a, b]$.

Example 1. Consider the three dimensional fPDE

$$\begin{aligned}\frac{\partial p}{\partial t} \;=\; & K_{1x}\,{}_0D_x^\alpha p + K_{2x}\,{}_xD_2^\alpha p + K_{1y}\,{}_0D_y^\beta p + K_{2y}\,{}_yD_2^\beta p + K_{1z}\,{}_0D_z^\gamma p + K_{2z}\,{}_zD_2^\gamma p \\ & + d_x\,\frac{\partial p}{\partial x} + d_y\,\frac{\partial p}{\partial y} + d_z\,\frac{\partial p}{\partial z} + f(x,y,z,t),\end{aligned} \tag{4.25}$$

that is a linear transport-diffusion model equation (with ordinary transport and fractional diffusion), on the domain $(x, y, z) \in [0, 2]^3$, $0 < t \le T$, with coefficients

$$\begin{aligned} K_{1x} &:= \Gamma(3-\alpha)\,x^\alpha, & K_{2x} &= \Gamma(3-\alpha)(2-x)^\alpha \\ K_{1y} &:= \Gamma(3-\beta)\,y^\beta, & K_{2y} &= \Gamma(3-\beta)(2-y)^\beta \\ K_{1z} &:= \Gamma(3-\gamma)\,x^\gamma, & K_{2z} &= \Gamma(3-\gamma)(2-z)^\gamma, \\ d_x &:= \frac{x}{4}, & d_y &:= \tfrac{y}{4}, \qquad d_z := \tfrac{z}{4}. \end{aligned}$$

We define the forcing term

$$\begin{aligned} f(x,y,z,t) &:= -4\,e^{-t}x^2y^2z^2(x-2)(y-2)(z-2)(3xyz-5x-5y-5z+8) \\ &\quad -\left[l_\alpha(x,z,t) + l_\gamma(z,x,t) + l_\alpha(y,x,t) + l_\beta(x,y,t) + l_\beta(z,y,t) + l_\gamma(y,z,t)\right], \end{aligned} \tag{4.26}$$

where $l_\delta(u, v, t) := g(u, t)\, h_\delta(v)$, being

$$g(u,t) := 32\,e^{-t}u^2(2-u)^2,$$

and

$$h_\delta(u) := u^2 + (2-u)^2 - \frac{3\left[u^3 + (2-u)^3\right]}{3-\delta} + \frac{3\left[u^4 + (2-u)^4\right]}{(3-\delta)(4-\delta)},$$

and impose furthermore, for all $t > 0$, the homogeneous Dirichlet boundary conditions

$$p(0, y, z, t) = p(x, 0, y, t) = p(x, y, 0, t) = 0,$$
$$p(2, y, z, t) = p(x, 2, y, t) = p(x, y, 2, t) = 0,$$

as well as the initial condition

$$p(x, y, z, 0) = 4x^2(2 - x)^2y^2(2 - y)^2z^2(2 - z)^2.$$

The (exact) analytical solution to equation (4.25), under such initial and boundary conditions, is known, and is given by

$$p(x, y, z, t) = 4x^2e^{-t}(2 - x)^2y^2(2 - y)^2z^2(2 - z)^2,$$

see [45]. All this can be used to *validate* our 3D algorithm and test its performance. We will feel then confident that it might perform well also when other source terms, that might be encountered in real world problems, enter the model. Below, we will indeed replace the special forcing term in (4.25) with some other more realistic ones.

From Fig. 4.1, it is clear what is the effect of replacing ordinary derivatives, $(\alpha, \beta, \gamma) = (2, 2, 2)$, with fractional derivatives, in a given diffusion equation. Here, $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$, thus accounting for *anomalous* diffusion. In general, such a modification implies new *geometric patterns* in the solution, and a possibly *anisotropic* behavior. Even a different *speed of propagation*, depending on the order of the fractional derivatives, can be reproduced in this way. Indeed, all these features have been observed, e.g., in certain porous media through which some fluid flows [36, 129]. Hereafter T denotes the final time at which the solution is computed.

In Fig. 4.2, the numerical errors $\|q^n - p^n\|$ are plotted in the L^2 and in the L^∞ norms, on a log-scale, where $p^n \equiv p(x_i, y_j, z_k, t_n)$ denotes the exact solution to the fPDE (on the grid points defined in Section 2.1, and $q^n \equiv q^n(\tau)$ is its approximation given by our scheme through equation (4.24).

In order to assess the convergence rate when space and time step size are refined, we write our approximate solution q^n as $q(h, \tau)$ to display its dependence on the space step sizes, $h_x = h_y = h_z = h$ (assumed to be equal), and the time step size, τ. The, we define, for every $N \in \mathbf{N}$, the *time* and the *space convergence rate*, in the 2-norm, for a given method of order $m > 1$, as

$$r_{time}(N) := \log_2 \left[\frac{\|q(h, \frac{h}{N}) - p^n\|_2}{\|q(h, \frac{h}{2N}) - p^n\|_2^m} \right], \quad r_{space}(N) := \log_2 \left[\frac{\|q(\frac{h}{N}, h) - p^n\|_2}{\|q(\frac{h}{2N}), h) - p^n\|_2^m} \right].$$

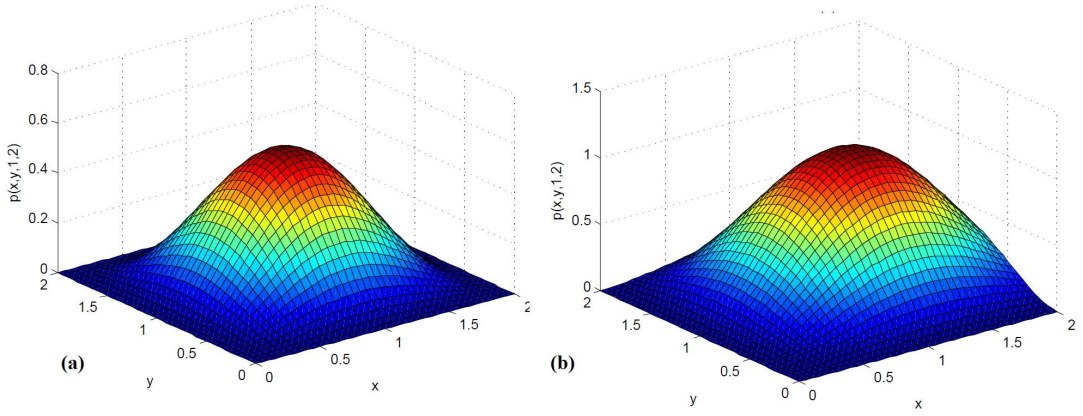

FIGURE 4.1: (a) Classical solution (obtained by a fine grid numerical ADI method with $\tau = h/16$ and $h = 1/64$), and (b) exact (analytical) fractional diffusion solution with $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$, for $z = 1$ and $T = 2$. The same forcing function was used in both cases.

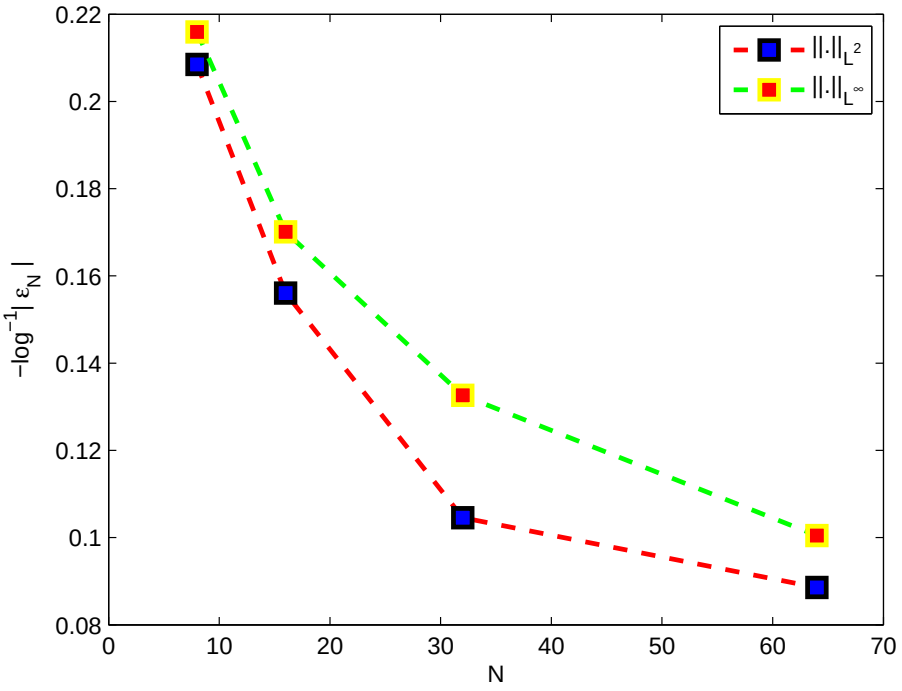

FIGURE 4.2: Absolute numerical error ε_N (in log scale) between the exact and the numerical solution of the fPDE of Example 1, at $T = 2$, with $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$.

In Tables 4.1 and 4.2, the absolute numerical errors $\|q^n - p^n\|_\infty$ and $\|q^n - p^n\|_2$, as well as the corresponding convergence rates attained using different space step sizes, are shown. Figure 4.2 displays the infinity norm and the L^2 norm errors for Example 1, when the optimized difference scheme is implemented in (4.24), at times $T = 1$ and $T = 2$, for several values of $h_x = h_y = h_z = h = \tau = 2/N$, $N = 8, 16, 32, 64$, and fixed α, β, and γ. Here there is evidence that numerical errors decrease when increasing N, i.e., decreasing

τ. Note that, due to the logarithmic scale, smaller and smaller errors correspond to larger and larger values.

TABLE 4.1: L^∞ and L^2 norm errors (in log scale) and convergence rates for Example 1, when the *balanced* scheme is used, at time $T = 1$, for several values of (α, β, γ), N.

(α, β, γ)	N	$\|q^n - p^n\|_\infty$	r_{space}	$\|q^n - p^n\|_2$	r_{space}	r_{time}
$(1.2, 1.2, 1.2)$	8	$1.5023 \cdot 10^{-2}$	$--$	$7.5264 \cdot 10^{-2}$	$--$	$--$
	16	$2.8925 \cdot 10^{-3}$	2.002	$1.6581 \cdot 10^{-2}$	2.015	3.009
	32	$6.8952 \cdot 10^{-4}$	2.000	$5.3265 \cdot 10^{-3}$	2.007	3.006
	64	$2.3561 \cdot 10^{-4}$	1.996	$2.3654 \cdot 10^{-4}$	1.999	3.001
$(1.4, 1.5, 1.6)$	8	$2.0253 \cdot 10^{-2}$	$--$	$3.2564 \cdot 10^{-2}$	$--$	$--$
	16	$2.9541 \cdot 10^{-3}$	2.001	$3.1254 \cdot 10^{-2}$	2.011	3.010
	32	$7.1254 \cdot 10^{-4}$	2.000	$7.9856 \cdot 10^{-3}$	2.007	3.005
	64	$2.5648 \cdot 10^{-4}$	1.988	$2.8930 \cdot 10^{-2}$	1.998	2.998
$(1.9, 1.9, 1.9)$	8	$2.324 \cdot 10^{-2}$	$--$	$3.5852 \cdot 10^{-2}$	$--$	$--$
	16	$3.5984 \cdot 10^{-3}$	2.005	$3.9852 \cdot 10^{-2}$	2.009	3.002
	32	$7.5214 \cdot 10^{-4}$	2.002	$7.7815 \cdot 10^{-3}$	2.008	3.001
	64	$2.8594 \cdot 10^{-4}$	1.992	$3.2852 \cdot 10^{-4}$	1.989	3.000
$(2.0, 2.0, 2.0)$	8	$5.2154 \cdot 10^{-3}$	$--$	$6.7952 \cdot 10^{-3}$	$--$	$--$
	16	$6.7854 \cdot 10^{-4}$	2.001	$7.1248 \cdot 10^{-4}$	2.001	3.000
	32	$5.7453 \cdot 10^{-5}$	1.999	$7.9173 \cdot 10^{-5}$	1.998	3.000
	64	$6.7852 \cdot 10^{-5}$	2.000	$7.7945 \cdot 10^{-5}$	2.000	2.999

TABLE 4.2: L^∞ and L^2 norm errors, and convergence rates for Example 1, when the *balanced* scheme is used, at time $T = 2$, for several values of N.

(α, β, γ)	N	$\|q^n - p^n\|_\infty$	r_{space}	$\|q^n - p^n\|_2$	r_{space}	r_{time}
$(1.4, 1.5, 1.6)$	8	$8.2654 \cdot 10^{-3}$	$--$	$9.7452 \cdot 10^{-3}$	$--$	$--$
	16	$2.6584 \cdot 10^{-4}$	2.001	$2.7852 \cdot 10^{-3}$	2.001	3.002
	32	$7.0145 \cdot 10^{-5}$	2.000	$5.3255 \cdot 10^{-4}$	1.999	3.000
	64	$1.2420 \cdot 10^{-5}$	1.999	$4.7852 \cdot 10^{-5}$	2.000	2.999

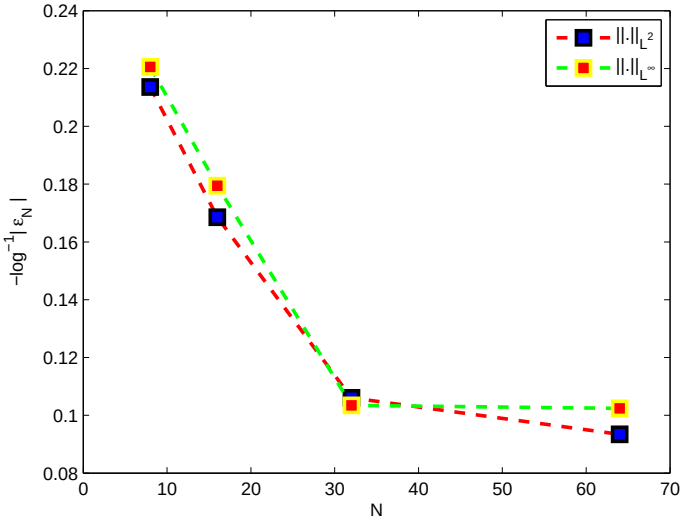

FIGURE 4.3: L^∞ and L^2 discrepancy, again denoted by ε_N (in log scale) between the numerical solution of the classical problem and that of the fractional problem with $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$, at $T = 2$.

In Fig. 4.3, the discrepancies between the numerical solutions to the classical and the fractional differential equations, obtained by the ADI and the $fADI$ method, respectively, i.e., $\|q^n_{ADI} - q^n_{fADI}\|$, are plotted in the L^2 and in the L^∞ norms, on an log-scale. This is done to appreciate the differences visible when one switches from one to the other model. In Tables 4.3 and 4.4, such discrepancies, $\|q^n_{ADI} - q^n_{fADI}\|_\infty$ and $\|q^n_{ADI} - q^n_{fADI}\|_2$, as well as the corresponding convergence rates, achieved using different space step sizes, are shown. Here again $q^n(\tau)$, given by

$$\frac{147}{2500}\,p^n(\tau) - \frac{29409}{100000}\,p^n\left(\frac{3}{4}\tau\right) - \frac{147}{2500}\,p^n\left(\frac{\tau}{2}\right) + \frac{147}{2500}\,p^n\left(\frac{\tau}{4}\right), \tag{4.27}$$

is the extrapolated solution, and $p^n \equiv p^n_{i,j,k}$ satisfies the optimized scheme in (4.8). Figure 4.3 shows the convergence rates of the maximum norm and the L^2 norm errors for Example 1, when the optimized difference scheme in (4.27) is implemented, at times $T = 1$, for several values of N, and fixed α, β, and γ.

TABLE 4.3: L^∞ and L^2 norm discrepancy for Example 1, when the *balanced* scheme is used, at time $T = 1$, for several values of N.

(α, β, γ)	N	$\|q^n_{fADI} - q^n_{ADI}\|_\infty$	$\|q^n_{fADI} - q^n_{ADI}\|_2$
$(1.2, 1.2, 1.2)$	256	$6.1758 \cdot 10^{-7}$	$3.8458 \cdot 10^{-6}$
	512	$2.2548 \cdot 10^{-7}$	$7.8442 \cdot 10^{-7}$
$(1.4, 1.5, 1.6)$	256	$2.8287 \cdot 10^{-7}$	$1.1205 \cdot 10^{-6}$
	512	$3.7852 \cdot 10^{-8}$	$3.7854 \cdot 10^{-7}$
$(1.9, 1.9, 1.9)$	256	$1.1582 \cdot 10^{-7}$	$5.7852 \cdot 10^{-7}$
	512	$1.7855 \cdot 10^{-8}$	$5.0023 \cdot 10^{-7}$
$(2.0, 2.0, 2.0)$	256	$8.1158 \cdot 10^{-8}$	$8.7852 \cdot 10^{-7}$
	512	$1.7852 \cdot 10^{-8}$	$6.7852 \cdot 10^{-8}$

TABLE 4.4: L^∞ and L^2 norm discrepancy for Example 1, when the *balanced* scheme is used, at time $T = 2$, for several values of N.

(α, β, γ)	N	$\|q^n_{fADI} - q^n_{ADI}\|_\infty$	$\|q^n_{fADI} - q^n_{ADI}\|_2$
$(1.4, 1.5, 1.6)$	256	$5.7854 \cdot 10^{-7}$	$1.4002 \cdot 10^{-8}$
	512	$2.7584 \cdot 10^{-8}$	$7.7852 \cdot 10^{-8}$

Table 4.5 shows CPU times by our algorithm on a *Dual Core Pentium* with 4GB RAM and with 32bit-MATLAB programs, for different values of fractional orders and advection coefficients d_x, d_y, d_z. The CPU time is measured in terms of *clocks ticks*.

Tables 4.5 and 4.6, indicate that the our *balanced* ADI method's performance is comparable to that proposed in [88], though slightly worse. Recall that our method and that of Liu et al. differ in the strategies followed to accelerate them. Our *unbalanced* method however performs better than that, requiring smaller CPU times. This is due to the fact that the model problem is anisotropic, behaving along the direction x rather differently than along the other directions. Indeed, all Tables, 4.7, 4.8, and 4.9, show that when the

TABLE 4.5: CPU times in *clocks ticks* units (CT) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, and CT_{unbal}, respectively, required to attain an error of order 10^{-5}, for $h = 1/100$ and several values of the fractional orders, α, β, γ and the advection coefficients d_x, d_y, d_z.

(α, β, γ)	(d_x, d_y, d_z)	CT_{unbal}	CT_{bal}
$(1.8, 1.5, 1.2)$	$(10, 5, 1)$	$2.256 \cdot 10^{-3}$	$8.236 \cdot 10^{-2}$
$(1.8, 1.5, 1.3)$	$(8, 4, 1)$	$4.365 \cdot 10^{-3}$	$4.255 \cdot 10^{-2}$
$(1.8, 1.6, 1.4)$	$(7, 3, 1)$	$3.256 \cdot 10^{-2}$	$5.778 \cdot 10^{-2}$
$(1.8, 1.6, 1.5)$	$(5, 2, 1)$	$7.289 \cdot 10^{-2}$	$2.389 \cdot 10^{-3}$
$(1.8, 1.7, 1.6)$	$(3, 1, 1)$	$6.258 \cdot 10^{-1}$	$1.756 \cdot 10^{-3}$
$(1.8, 1.7, 1.7)$	$(2, 1, 1)$	$8.236 \cdot 10^{-1}$	$1.586 \cdot 10^{-3}$
$(1.8, 1.8, 1.8)$	$(1, 1, 1)$	$9.266 \cdot 10^{-1}$	$1.256 \cdot 10^{-3}$

TABLE 4.6: CPU times in *clocks ticks* units (CT) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, and CT_{unbal}, respectively, required to attain an error of order 10^{-5}, for $N = 100$ and several values $h = \frac{1}{\gamma N}$, and of the fractional orders, α, β, γ.

(α, β, γ)	h	CT_{unbal}	CT_{bal}
$(1.8, 1.5, 1.2)$	0.08	$5.256 \cdot 10^{-3}$	$7.256 \cdot 10^{-2}$
$(1.8, 1.5, 1.3)$	0.08	$6.001 \cdot 10^{-2}$	$3.856 \cdot 10^{-2}$
$(1.8, 1.6, 1.4)$	0.07	$7.898 \cdot 10^{-2}$	$4.258 \cdot 10^{-2}$
$(1.8, 1.6, 1.5)$	0.07	$8.856 \cdot 10^{-2}$	$1.759 \cdot 10^{-2}$
$(1.8, 1.7, 1.6)$	0.07	$4.125 \cdot 10^{-1}$	$1.896 \cdot 10^{-2}$
$(1.8, 1.7, 1.7)$	0.06	$7.156 \cdot 10^{-1}$	$1.325 \cdot 10^{-3}$
$(1.8, 1.8, 1.8)$	0.05	$8.780 \cdot 10^{-1}$	$1.126 \cdot 10^{-3}$

anisotropy occurs in another direction, others than x, an *appropriately balanced* method wins over the others.

TABLE 4.7: CPU times in *clocks ticks* units (CT) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, and CT_{unbal}, respectively, required to attain an error of order 10^{-5}, for $h = 1/100$ and several values of the fractional orders, α, β, γ and the advection coefficients d_x, d_y, d_z.

(α, β, γ)	(d_x, d_y, d_z)	CT_{unbal}	CT_{bal}
$(1.5, 1.8, 1.2)$	$(5, 10, 1)$	$7.152 \cdot 10^{-2}$	$8.256 \cdot 10^{-2}$
$(1.5, 1.8, 1.3)$	$(4, 8, 1)$	$9.856 \cdot 10^{-2}$	$7.026 \cdot 10^{-2}$
$(1.6, 1.8, 1.4)$	$(3, 7, 1)$	$6.856 \cdot 10^{-1}$	$5.785 \cdot 10^{-2}$
$(1.6, 1.8, 1.5)$	$(2, 5, 1)$	$7.019 \cdot 10^{-1}$	$3.256 \cdot 10^{-3}$
$(1.7, 1.8, 1.6)$	$(1, 3, 1)$	$8.786 \cdot 10^{-1}$	$1.756 \cdot 10^{-3}$
$(1.7, 1.8, 1.7)$	$(1, 2, 1)$	$9.325 \cdot 10^{-1}$	$1.586 \cdot 10^{-3}$
$(1.8, 1.8, 1.8)$	$(1, 1, 1)$	1.256	$1.025 \cdot 10^{-3}$

TABLE 4.8: CPU times in *clocks ticks* units (CT) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, and CT_{unbal}, respectively, required to attain an error of order 10^{-5}, for $N = 100$ and several values $h = \frac{1}{\gamma N}$, and of the fractional orders, α, β, γ.

(α, β, γ)	h	CT_{unbal}	CT_{bal}
$(1.5, 1.8, 1.2)$	0.08	$7.266 \cdot 10^{-2}$	$5.125 \cdot 10^{-2}$
$(1.5, 1.8, 1.3)$	0.08	$9.014 \cdot 10^{-2}$	$4.856 \cdot 10^{-2}$
$(1.6, 1.8, 1.4)$	0.07	$5.888 \cdot 10^{-1}$	$2.558 \cdot 10^{-2}$
$(1.6, 1.8, 1.5)$	0.07	$7.896 \cdot 10^{-1}$	$1.325 \cdot 10^{-3}$
$(1.7, 1.8, 1.6)$	0.07	$3.965 \cdot 10^{-1}$	$1.756 \cdot 10^{-3}$
$(1.7, 1.8, 1.7)$	0.06	7.156	$1.256 \cdot 10^{-3}$
$(1.8, 1.8, 1.8)$	0.05	8.780	$9.254 \cdot 10^{-4}$

TABLE 4.9: CPU times in *clocks ticks* units (CT) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, and CT_{unbal}, respectively, required to attain an error of order 10^{-5}, for $h = 1/100$, $d_x = d_y = d_z = 0$, and several values of the fractional orders, α, β, γ.

(α, β, γ)	CT_{unbal}	CT_{bal}
$(1.5, 1.5, 1.5)$	8.458	$7.758 \cdot 10^{-4}$
$(1.3, 1.8, 1.3)$	$2.786 \cdot 10^{-1}$	$7.256 \cdot 10^{-2}$
$(1.2, 1.5, 1.2)$	$6.289 \cdot 10^{-1}$	$2.389 \cdot 10^{-3}$
$(1.5, 1.7, 1.5)$	$6.258 \cdot 10^{-1}$	$1.756 \cdot 10^{-3}$
$(1.7, 1.8, 1.7)$	$3.276 \cdot 10^{-1}$	$5.785 \cdot 10^{-3}$
$(1.8, 1.8, 1.8)$	$1.256 \cdot$	$8.785 \cdot 10^{-4}$

Example 2. Let choose, in equation (4.25), $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$ and the *nonlinear* source term $f(p) = \frac{p}{2}$, instead of the linear source $f(x, y, z, t)$ defined in (4.26) above. Therefore, we now face a fractional *reaction-diffusion* problem (a semilinear fractional PDE). A forcing term like this occurs often, for instance, in modeling dissolution and precipitation phenomena in porous media [63, 139].

In Table 4.10, we show the numerical results pertaining to this problem, obtained using the the fADI method for the fPDE with $(\alpha, \beta, \gamma) = (1.4, 1.5, 1.6)$.

TABLE 4.10: L^2 norm discrepancies, and convergence rates for Example 2, when the *balanced* scheme is used, at time $T = 2$, source term $f(p) = \frac{p}{2}$, $(\alpha, \beta, \gamma) = (1.1, 1.7, 1)$, $h = 1/N$, and several values of N.

N	τ	$\|q^n_{fADI} - q^n_{ADI}\|_2$	$rate_{space}$	$rate_{time}$
16	$h/16$	$8.5825 \cdot 10^{-4}$	$--$	$--$
32	$h/16$	$3.8501 \cdot 10^{-4}$	2.000	3.001
64	$h/16$	$1.7854 \cdot 10^{-5}$	2.000	3.000

Example 3. Consider now the linear fractional diffusion equation with an impulsive source,

$$\frac{\partial p}{\partial t} = K_x \frac{\partial^\alpha p}{\partial x^\alpha} + K_y \frac{\partial^\beta p}{\partial y^\beta} + K_x \frac{\partial^\gamma p}{\partial z^\gamma} + c\,\delta(x - x_0, y - y_0, z - z_0, t - t_0), \tag{4.28}$$

where δ is the Dirac delta function, c, x_0, y_0, z_0, and t_0 are all positive constants, $(x, y, z) \in \Omega = (x_1, x_2) \times (y_1, y_2) \times (z_1, z_2)$, $t > 0$, and subject to the boundary conditions

$$p(x, y, z, t)|_{\partial\Omega} = h(x, y, z, t), \quad (x, y, z) \in \partial\Omega, \quad t \in (0, T],$$

for some $T > 0$, and to the initial value

$$p(x, y, z, 0) = p_0(x, y, z), \quad (x, y, z) \in \overline{\Omega}.$$

Here, the forcing term mimics the behavior of a seepage flow in a porous medium which can be either isotropic or even anisotropic in the x, y, and z direction (depending on the

coefficients K_x, K_y, K_z). In view of a numerical treatment of such a problem, we replace the ideally impulsive source with a function likely to have a similar behavior, see [39]. Let be $\alpha = \beta = \gamma = 1.8$, the domain $\Gamma = (0,1)^3$,

$$K_x = 0.1\,\frac{\Gamma(1.4)}{\Gamma(3.2)}, \quad K_y = 0.1\,\frac{\Gamma(1.2)}{\Gamma(3)}, \quad K_z = 0.1\,\frac{\Gamma(1.4)}{\Gamma(3)},$$

and the source term, which reproduces approximately the behavior of the δ function in (4.28),

$$\begin{aligned} f(x,y,z,t) = \; & -e^{-t}\,(x^{2.2}\,y^2 z^2 - 0.1\,x^{0.4}\,y^2 z^2 + \\ & -0.1\,x^2 y^{0.2}\,z^2 - 0.1\,x^2 y^2 z^{0.4}). \end{aligned}$$

We impose the boundary conditions

$$p(0,y,z,t) = p(x,0,y,t) = p(x,y,0,t) = 0,$$

$$p(1,y,z,t) = e^{-t} y^2 z^2, \;\; p(x,1,y,t) = e^{-t} x^{2.2}\,z^2, \;\; p(x,y,1,t) = e^{-t} x^{2.2}\,y^2,$$

and the initial value

$$p_0(x,y,z) = 0.$$

The analytic solution to this problem is known to be

$$p(x,y,z,t) = e^{-t} x^{2.2}\,y^2 z^2, \tag{4.29}$$

see [88]. Note that, when $(\alpha,\beta,\gamma) = (2,2,2)$, we obtain the classical linear forced diffusion equation $p_t = \Delta p + f$.

We now compare the previous numerical solution with α, β, γ in the range $[1.2, 1.8]$, computed using our balanced fADI method, with that of a classical (i.e., nonfractional) case, that we can consider rather "close" to it, taking $\alpha = \beta = \gamma = 2$ and the same source term defined in (4.29).

In Fig. 4.4 the absolute numerical errors between the exact solution and the numerical solution obtained by a fADI method are shown. In Fig. 4.5 the same is done for the absolute numerical errors between the numerical classical ADI method and the numerical solution obtained by the fADI method. Both errors are plotted in the L^2 norm.

Table 4.11 illustrates the convergence rates of the algorithm, measured in the L^2 norm, in Example 3 for several values of τ and N, and fixed α, β, and γ. Here, p^n is the exact solution, while q^n_{ADI} is the solution obtained by the classical ADI method, and q^n_{fADI} is that provided by the fADI method.

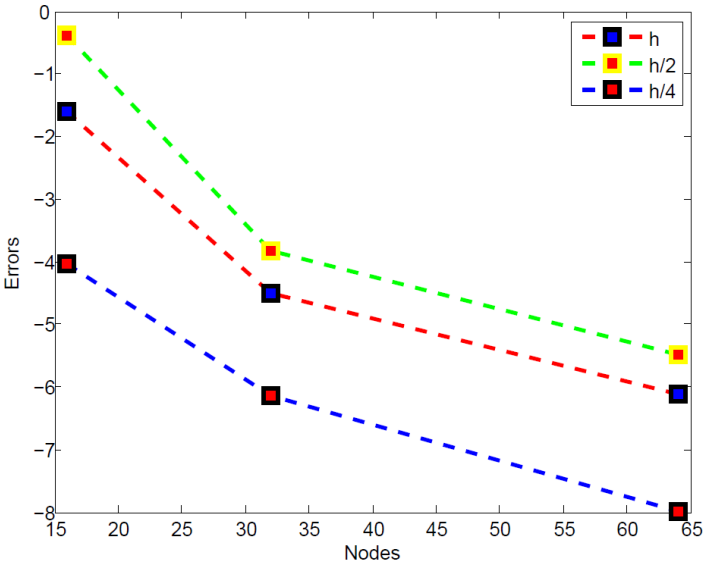

FIGURE 4.4: Numerical absolute errors $\|q^n_{fADI} - p^n\|$ for three values of the time and the space step-sizes, $\tau = h$ (solid-red), $\tau = h/2$ (dashed-green), $\tau = h/4$ (dashed and dotted-blue), for $\alpha = \beta = \gamma = 1.8$, $h = 1/N$, on a *log*-scale.

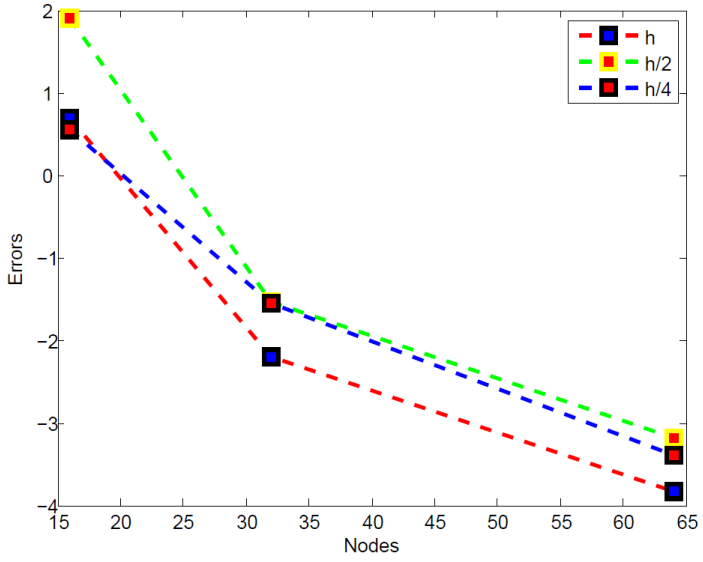

FIGURE 4.5: Discrepancy $\|q^n_{fADI} - q^n_{ADI}\|$ for three values of time and space step-sizes, $\tau = h$ (solid-red), $\tau = h/2$ (dashed-green), $\tau = h/4$ (dashed and dotted-blue), for $\alpha = \beta = \gamma = 1.8$, $h = 1/N$, on a log-scale.

As one would expect in the fADI methods, coupled with the *optimized extrapolation*, a *faster* convergence is observed as the grid is refined. In the following, we show these results in a table of computational times, see Table 4.12.

Table 4.13 shows CPU times and memory used by our algorithm for various grid sizes. The CPU time is evaluated at $T = 1$. Table 4.13 indicates that the *balanced* ADI method performs slightly worse than that one proposed in [88]. However, this little price is paid to obtain a third-order (instead of a second-order) in time method. It was also

TABLE 4.11: L^2 norm errors, discrepancies and convergence rates for Example 3, when the *balanced* scheme is used, with $\alpha = \beta = \gamma = 1.8$, at time $T = 1$, for $h = 1/N$ and several values of N and τ. The two numbers in the rates columns refer to the first and second algorithms, respectively.

N	τ	$\|q^n_{fADI} - p^n\|_2$	$\|q^n_{fADI} - q^n_{ADI}\|_2$	space conv. rates	time conv. rates
16	h	$2.0122 \cdot 10^{-1}$	2.5684	$--$	$--$
32	h	$3.8524 \cdot 10^{-2}$	$1.8524 \cdot 10^{-1}$	$--$	$--$
64	h	$2.1852 \cdot 10^{-3}$	$3.2658 \cdot 10^{-2}$	$--$	$--$
16	$h/2$	$5.7616 \cdot 10^{-1}$	1.0226	$1.96 - 1.82$	$2.98 - 2.92$
32	$h/2$	$3.1658 \cdot 10^{-2}$	$2.5802 \cdot 10^{-1}$	$1.99 - 1.93$	$2.99 - 2.95$
64	$h/2$	$3.5800 \cdot 10^{-3}$	$6.3552 \cdot 10^{-2}$	$1.98 - 1.89$	$2.99 - 2.96$
16	$h/4$	$1.7525 \cdot 10^{-2}$	3.2753	$2.00 - 1.96$	$3.00 - 2.98$
32	$h/4$	$2.1358 \cdot 10^{-3}$	$2.1258 \cdot 10^{-1}$	$2.00 - 1.98$	$3.00 - 2.99$
64	$h/4$	$3.3654 \cdot 10^{-4}$	$9.4528 \cdot 10^{-2}$	$2.00 - 2.00$	$3.00 - 3.01$

TABLE 4.12: Computational times (in seconds) for the *unbalanced* and *balanced* version of the fADI scheme, respectively, T_{unbal} and T_{bal}, with $(\alpha, \beta, \gamma) = (1.8, 1.6, 1.4)$, at time $T = 1$, for $h = 1/N$ and several values of N and τ.

N	τ	T_{unbal}	T_{bal}
16	h	7	7
32	h	15	16
64	h	32	34
16	$h/2$	11	12
32	$h/2$	20	23
64	$h/2$	37	39
16	$h/4$	19	20
32	$h/4$	41	43
64	$h/4$	83	81

observed that balanced and unbalanced algorithm require about the same computational resources.

TABLE 4.13: CPU times in *clocks ticks* units (CT), and memory (RAM in MB) for the *unbalanced* and the *balanced* versions of the fADI scheme, CT_{bal}, RAM_{bal} and CT_{unbal}, RAM_{unbal}, respectively, with $(\alpha, \beta, \gamma) = (1.8, 1.6, 1.4)$, at time $T = 1$, for $h = 1/N$ and several values of N and τ.

N	τ	CT_{bal}	RAM_{bal}	CT_{unbal}	RAM_{unbal}
16	h	$1.944 \cdot 10^{-3}$	0.15	$2.778 \cdot 10^{-3}$	0.23
32	h	$4.416 \cdot 10^{-3}$	0.23	$5.278 \cdot 10^{-3}$	0.34
64	h	$8.889 \cdot 10^{-2}$	0.81	$1.027 \cdot 10^{-2}$	0.92
16	$h/2$	$3.050 \cdot 10^{-2}$	0.24	$3.611 \cdot 10^{-3}$	0.33
32	$h/2$	$5.555 \cdot 10^{-3}$	0.42	$6.944 \cdot 10^{-3}$	0.51
64	$h/2$	$1.028 \cdot 10^{-2}$	1.23	$1.361 \cdot 10^{-2}$	1.38
16	$h/4$	$5.278 \cdot 10^{-3}$	0.59	$6.389 \cdot 10^{-3}$	0.67
32	$h/4$	$1.139 \cdot 10^{-2}$	0.83	$1.310 \cdot 10^{-3}$	0.95
64	$h/4$	$2.778 \cdot 10^{-2}$	1.70	$2.667 \cdot 10^{-2}$	1.92

Example 4. We consider the semilinear (reaction-diffusion) equation [7]

$$\frac{\partial p}{\partial t} = C\frac{\partial^\alpha p}{\partial x^\alpha} + D\frac{\partial^\beta p}{\partial y^\beta} + E\frac{\partial^\gamma p}{\partial z^\gamma} + f_{r,k}(p), \tag{4.30}$$

on the set

$$\Omega \;=\; \Omega_1 \times \Omega_2 \times \Omega_3 = ([-20,10] \times \overline{I}_{20}^2) \times ([10,20] \times [2,4] \times \overline{I}_{20}) \times ([20,100] \times \overline{I}_{20}^2),$$

where $I_{20} := (-20,20)$, $t \geq 0$, (D,E,F) is the percolation tensor, and $f_{r,k}(p) := rp\big(1-\frac{p}{k}\big)$ is the well-known Kolmogorov-Fisher function [104, 105], often adopted in population biology to model the spread of invasive species. Here, r and k are parameters, r representing the intrinsic increase rate of the fluid, and k the environmental carrying capacity, that is the maximum sustainable fluid density. We computed the numerical solution to the initial-boundary value problem above, assuming the radially symmetric initial condition

$$p(x,y,z,0) = \min_{(x,y,z)\in\Omega_1} \{0.8, 10\, e^{-(x^2+y^2+z^2)}\}, \tag{4.31}$$

and the boundary conditions

$$p(x,y,z,t)|_{\partial\Omega} = p_y(x,\pm 20,z,t) = p_x(\pm 20,y,z,t) = p_x(10,y,z,t) = 0. \tag{4.32}$$

Next, we consider the solution to a fPDE like that in (4.30), but with a reaction term having space variable coefficients. Let be $C = 0.15, D = 0,4, E = 1, r = 0.2$, and assume that k varies in space as follows: for $(x,y,z) \in \Omega' := \Omega'_1 \cup \Omega'_2 \cup \Omega'_3$,

$$k = k(x,y,z) := \begin{cases} 10^{-6} & \text{for } (x,y,z) \in \Omega'_1 := (-30,10) \times I_{20}^2 \\ 1 & \text{for } (x,y,z) \in \Omega'_2 := (20,10) \times I_{20}^2, \end{cases}$$

and k being smoothly interpolated in $\Omega'_3 := (10,20) \times (2,4) \times I_{20}$, see Figs. 4.6, 4.7, 4.8.

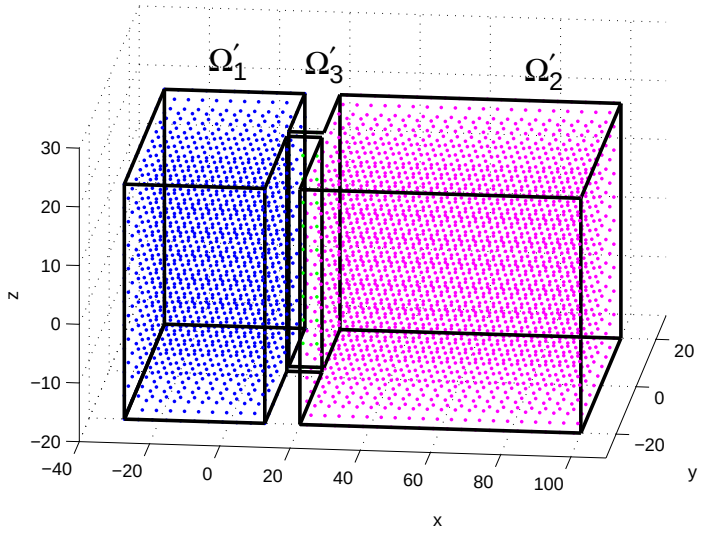

FIGURE 4.6: Domain for the variable reaction term coefficient $k(x,y,z)$ divided into the regions Ω'_1 (dotted blue), Ω'_3 (dotted green) and Ω'_2 (dotted magenta).

In many applications to meteorology, this case may model a region which can contains at most a certain given amount of fluid, due to certain environmental conditions. The geometry is a silt barrier through which the fluid will eventually penetrate. We first consider the case $\alpha = \beta = \gamma = 2$. In Fig. 4.5 the solution pertaining to this case is shown, in a plan view, at time $T = 90$. Here, due to classical diffusion along x, the pressure can be seen to penetrate the barrier very slowly.

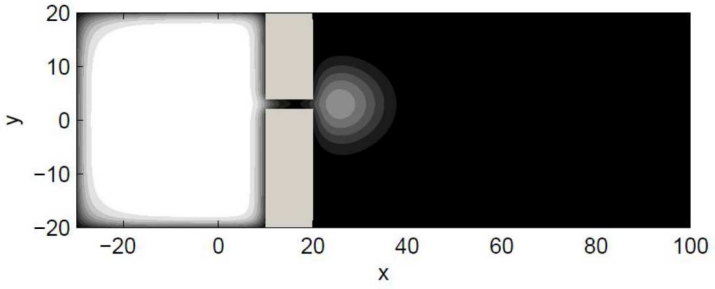

FIGURE 4.7: Solution to equation (4.30), at time $T = 90$, with the initial condition (4.31) and parameters $\alpha = \beta = \gamma = 2$, $C = 0.15, D = 0.4, r = 0.2$, on the cross section in the plane (x, y), this plane is divided into two regions, the pressure propagates slowly from a region (left) to the other (right) through a silt barrier which links them [7].

We then changed parameters, choosing $\alpha = 1.7, \beta = \gamma = 2$, to represent a certain *anomalous* diffusion. In Fig. 4.8, it is shown that at time $T = 50$, even before the "snapshot time" $T = 90$ (which refers to Fig. 4.7), the pressure has penetrated significantly the barrier, and it spreads at the same time in the y direction.

This is a striking peculiarity of fractional reaction-diffusion model which allows to predict the effects of controlling the fluid flow, for instance describing pollution due to a pollutant expanding from a given environment to another.

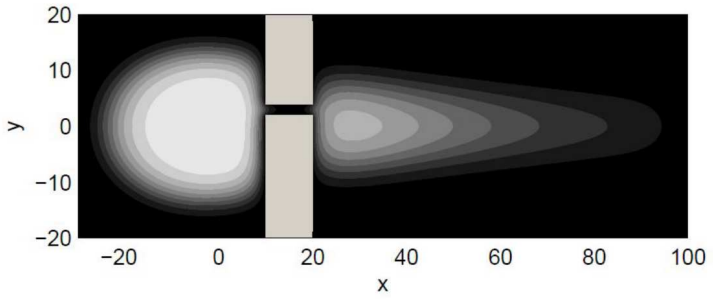

FIGURE 4.8: Solution to equation (4.30), at time $T = 50$, with initial condition (4.31) and parameter values $\alpha = 1.7, \beta = \gamma = 2$, $C = 0.15, D = 0.4, r = 0.2$. This creates a slit barrier, through which the fluid is shown to penetrates when fractional derivative model is adopted [7].

Conclusions and future directions

Here we give some conclusions from the investigations conducted in the course of this Thesis and suggest future directions of research.

In chapter 1, we gave a partial contribution to the Mainardi's conjecture, concerning only small intervals of the variable t. The method we proposed to evaluate numerically the M-L function $e_\alpha(t) \equiv E_\alpha(-t^\alpha)$, is based on the integration of a simple fODE satisfied by it. We did it using the predictor-corrector algorithm of K. Diethelm, and showed that it always wins, in terms of CPU time and RAM, over the code implemented by *Mathematica* code, for all values of $\alpha \in (0, 1)$ and for arbitrary times. It also wins over the MATLAB code with some limitations on α and t. An *adaptive* predictor-corrector algorithm we then implemented, however, always outperformed even the MATLAB code, for all α and t. No comparison can be made with the evaluation of the more general two-parameters M-L functions as well as with that of the one-parameter M-L function on the complex domain, that we did not consider.

In chapter 1, a few models describing *fractional relaxation* as well as *fractional oscillations* have been studied. These are based on simple fractional fODEs, some being related to the so-called Érdelyi-Kober operators.

In chapter 2, we compared the numerical results, obtained modeling one-dimensional anomalous diffusion by fPDEs, with some experimental (hence realistic) data, and showed that the numerical solution of the aforementioned fPDE fits the laboratory results better than using classical PDEs (with integer order derivatives). More precisely, we found that, in the cases considered, the results match within a small error, both in L^2 and in L^∞ norm, when the space fractional order is slightly below 2 (say, about 1.9, but the time fractional exponent seems to be optimal choosing a value around $0.6 - 0.7$. This shows that indeed several porous media exhibit memory effects and the departure from the classical description is evident.

In chapter 3, a weighted and shifted Grünwald-Letnikov difference (WSGD) operator is used to approximate RL fractional diffusion operators. It is shown that indeed *third-order* accuracy in time can be achieved solving numerically two-dimensional fPDEs by ADI-like methods. A new technique, designed to accelerate the algorithm, which is competitive with respect to the methods existing to date in the literature [44], has also been developed. While the present method seems to outperform all the other existing algorithms, using very dense grids to attain low errors may require, however, as one may expect, a considerable computational time.

In chapter 4 we introduced a new well "*balanced*", fractional version of the ADI method, to solve numerically a number of 3D diffusion as well as reaction-diffusion problems for fPDEs. These are important in a variety of problems arising from porous media modeling. We proved that such a method is *unconditionally stable* for every fractional order of the space derivatives, *second-order* accurate *in space*, and *third-order* accurate *in time*. The speed of convergence has been improved adopting an extrapolation technique, coupled with the optimization method used by the *PageRank* algorithm.

Future directions can easily be envisaged, since nowadays fPDEs seem to be relevant in many other fields, such as Economy and Finance, Biology and Demography, are being involved beside the more traditional of Viscoelasticity and Seismology. There is an increasing demand for numerical methods to tackle complex problems in several dimensions. Researchers, especially mathematicians, should resist the temptation of "fractionalizing" every classical model equation, checking first whether this kind of generalization ha sound (e.g. physical) bases which motivates this choice. It may be however useful to explore what could be extracted, which further or better explanation could be inferred form such approach.

Confining to the models considered in chapter 3 and chapter 4, it would be interesting to study the effects of *anisotropy* due to different fractional orders affecting different space directions.

Another sensible topic is the identification of the fractional orders themselves (inverse problems), in a given fractional differential equation. To date, very little can be found in the literature [13, 14]. Such identification was attempted, over the years, in a few laboratory experiments, hence basing on real measurements, e.g., for anomalous diffusion in several kinds of porous media [50, 75].

Appendix A

The Page Rank Algorithm

PageRank is an algorithm used by Google Search to rank websites in their search engine results. PageRank was named after Larry Page, one of the founders of Google. PageRank is a way of measuring the importance of website pages. According to Google:

> PageRank works by counting the number and quality of links to a page to determine a rough estimate of how important the website is. The underlying assumption is that more important websites are likely to receive more links from other websites.

It is not the only algorithm used by Google to order search engine results, but it is the first algorithm that was used by the company, and it is the best-known.

PageRank is a link analysis algorithm and it assigns a numerical weighting to each element of a hyperlinked set of documents, such as the World Wide Web, with the purpose of "measuring" its relative importance within the set. The algorithm may be applied to any collection of entities with reciprocal quotations and references. The numerical weight that it assigns to any given element E is referred to as the PageRank of E and denoted by $PR(E)$. Other factors like Author Rank can contribute to the importance of an entity.

A PageRank results from a mathematical algorithm based on the web graph, created by all World Wide Web pages as nodes and hyperlinks as edges, taking into consideration authority hubs such as cnn.com or usa.gov. The rank value indicates an importance of a particular page. A hyperlink to a page counts as a vote of support. The PageRank of a page is defined recursively and depends on the number and PageRank metric of all pages that link to it ("incoming links"). A page that is linked to by many pages with high PageRank receives a high rank itself.

Numerous academic papers concerning PageRank have been published since Page and Brin's original paper. In practice, the PageRank concept may be vulnerable to manipulation. Research has been conducted into identifying falsely influenced PageRank rankings. The goal is to find an effective means of ignoring links from documents with falsely influenced PageRank.

Other link-based ranking algorithms for Web pages include the HITS algorithm invented by Jon Kleinberg (used by Teoma and now Ask.com), the IBM CLEVER project, the TrustRank algorithm and the hummingbird algorithm.

The PageRank algorithm outputs a probability distribution used to represent the likelihood that a person randomly clicking on links will arrive at any particular page. PageRank can be calculated for collections of documents of any size. It is assumed in several research papers that the distribution is evenly divided among all documents in the collection at the beginning of the computational process. The PageRank computations require several passes, called "iterations", through the collection to adjust approximate PageRank values to more closely reflect the theoretical true value.

A probability is expressed as a numeric value between 0 and 1. A 0.5 probability is commonly expressed as a "50% chance" of something happening. Hence, a PageRank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with the 0.5 PageRank.

Assume a small universe of four web pages: A, B, C and D. Links from a page to itself, or multiple outbound links from one single page to another single page, are ignored. PageRank is initialized to the same value for all pages. In the original form of PageRank, the sum of PageRank over all pages was the total number of pages on the web at that time, so each page in this example would have an initial value of 1. However, later versions of PageRank, and the remainder of this section, assume a probability distribution between 0 and 1. Hence the initial value for each page is 0.25.

The PageRank transferred from a given page to the targets of its outbound links upon the next iteration is divided equally among all outbound links.

If the only links in the system were from pages B, C, and D to A, each link would transfer 0.25 PageRank to A upon the next iteration, for a total of 0.75.

$$PR(A) = PR(B) + PR(C) + PR(D).$$

Suppose instead that page B had a link to pages C and A, page C had a link to page A, and page D had links to all three pages. Thus, upon the next iteration, page B would

transfer half of its existing value, or 0.125, to page A and the other half, or 0.125, to page C. Page C would transfer all of its existing value, 0.25, to the only page it links to, A. Since D had three outbound links, it would transfer one third of its existing value, or approximately 0.083, to A. At the completion of this iteration, page A will have a PageRank of 0.458.

$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}.$$

In other words, the PageRank conferred by an outbound link is equal to the document's own PageRank score divided by the number of outbound links $L(\dot{})$.

$$PR(A) = \frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)}.$$

In the general case, the PageRank value for any page u can be expressed as:

$$PR(u) = \sum_{v \in B_u} \frac{PR(v)}{L(v)},$$

i.e. the PageRank value for a page u is dependent on the PageRank values for each page v contained in the set Bu (the set containing all pages linking to page u), divided by the number $L(\mathbf{v})$ of links from page **v**.

The PageRank theory holds that an imaginary surfer who is randomly clicking on links will eventually stop clicking. The probability, at any step, that the person will continue is a damping factor d. Various studies have tested different damping factors, but it is generally assumed that the damping factor will be set around 0.85.

The damping factor is subtracted from 1 (and in some variations of the algorithm, the result is divided by the number of documents (N) in the collection) and this term is then added to the product of the damping factor and the sum of the incoming PageRank scores. That is,

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

So any page's PageRank is derived in large part from the PageRanks of other pages. The damping factor adjusts the derived value downward. The original paper, however, gave the following formula, which has led to some confusion:

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

The difference between them is that the PageRank values in the first formula sum to one, while in the second formula each PageRank is multiplied by N and the sum becomes N. A statement in Page and Brin's paper that "the sum of all PageRank is one" and claims by other Google employees support the first variant of the formula above.

Page and Brin confused the two formulas in their most popular paper "The Anatomy of a Large-Scale Hypertextual Web Search Engine", where they mistakenly claimed that the latter formula formed a probability distribution over web pages.

Google recalculates PageRank scores each time it crawls the Web and rebuilds its index. As Google increases the number of documents in its collection, the initial approximation of PageRank decreases for all documents.

The formula uses a model of a random surfer who gets bored after several clicks and switches to a random page. The PageRank value of a page reflects the chance that the random surfer will land on that page by clicking on a link. It can be understood as a Markov chain in which the states are pages, and the transitions, which are all equally probable, are the links between pages.

If a page has no links to other pages, it becomes a sink and therefore terminates the random surfing process. If the random surfer arrives at a sink page, it picks another URL at random and continues surfing again.

When calculating PageRank, pages with no outbound links are assumed to link out to all other pages in the collection. Their PageRank scores are therefore divided evenly among all other pages. In other words, to be fair with pages that are not sinks, these random transitions are added to all nodes in the Web, with a residual probability usually set to $d = 0.85$, estimated from the frequency that an average surfer uses his or her browser's bookmark feature.

So, the equation is as follows:

$$PR(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PR(p_j)}{L(p_j)}$$

where $p_1, p_2, ..., p_N$ are the pages under consideration, $M(p_i)$ is the set of pages that link to p_i, $L(p_j)$ is the number of outbound links on page p_j, and N is the total number of pages.

The PageRank values are the entries of the dominant eigenvector of the modified adjacency matrix. This makes PageRank a particularly elegant metric: the eigenvector is

$$\mathbf{R} = \begin{bmatrix} PR(p_1) \\ PR(p_2) \\ \vdots \\ PR(p_N) \end{bmatrix}$$

where R is the solution of the equation

$$\mathbf{R} = \begin{bmatrix} (1-d)/N \\ (1-d)/N \\ \vdots \\ (1-d)/N \end{bmatrix} + d \begin{bmatrix} \ell(p_1,p_1) & \ell(p_1,p_2) & \cdots & \ell(p_1,p_N) \\ \ell(p_2,p_1) & \ddots & & \vdots \\ \vdots & & \ell(p_i,p_j) & \\ \ell(p_N,p_1) & \cdots & & \ell(p_N,p_N) \end{bmatrix} \mathbf{R}$$

where the adjacency function $\ell(p_i, p_j)$ is 0 if page p_j does not link to p_i, and normalized such that, for each j

$$\sum_{i=1}^{N} \ell(p_i, p_j) = 1,$$

i.e. the elements of each column sum up to 1, so the matrix is a stochastic matrix (for more details see the computation section below). Thus this is a variant of the eigenvector centrality measure used commonly in network analysis.

Because of the large eigengap of the modified adjacency matrix above, the values of the PageRank eigenvector can be approximated to within a high degree of accuracy within only a few iterations.

As a result of Markov theory, it can be shown that the PageRank of a page is the probability of arriving at that page after a large number of clicks. This happens to equal t^{-1} where t is the expectation of the number of clicks (or random jumps) required to get from the page back to itself.

One main disadvantage of PageRank is that it favors older pages. A new page, even a very good one, will not have many links unless it is part of an existing site (a site being a densely connected set of pages, such as Wikipedia).

Several strategies have been proposed to accelerate the computation of PageRank.

Various strategies to manipulate PageRank have been employed in concerted efforts to improve search results rankings and monetize advertising links. These strategies have severely impacted the reliability of the PageRank concept, which purports to determine which documents are actually highly valued by the Web community.

Since December 2007, when it started actively penalizing sites selling paid text links, Google has combated link farms and other schemes designed to artificially inflate PageRank. How Google identifies link farms and other PageRank manipulation tools is among Google's trade secrets [145].

Bibliography

[1] Adolfsson, K., Enelund, M., and Olsson, P., *On the fractional order model of viscoelasticity*, Mech. Time Depend. Mater. 9, 15-34. 2005.

[2] Alsina, C., Thomas, M.S., *A geometrical proof of a new inequality for the gamma function*, J. Inequal. Pure & Appl. Math., 6 (2), Art. 48, 2005.

[3] Atanackovic, T., Dolicanin, D., Pilipovic, S., Stankovic, B., *Cauchy problems for some classes of linear fractional differential equations*, Fractional Calculus and Applied Analysis, Volume 17, Issue 4, pp 1039-1059, December 2014 .

[4] Bagley, R.L. and Torvik, P.J., *On the fractional calculus model of viscoelastic behavior*, J. Rheol. 30(1), 133-155. 1986.

[5] Bell, M.L. and Nur, A., *Strength changes due to reservoir-induced pore pressure stresses and application to Lake Oroville*, J. Geophys. Res. 83, 4469-4483, 1978.

[6] Baumer, B., Benson, D.A., Meerschaert, M.M.„ Scheffler HP, *Stochastic solution of space-time fractional diffusion equations*, Phys. Rev. and Stat. Nonlin. Soft. Matter. Phys., 2002.

[7] Baumer, B., Kovács, M., Meerschaert, M., *Numerical solution for fractional reaction-diffusion equation*, USA National Science Foundation, 2005.

[8] Benson, D., Wheatcraft, S., Meerschaert, M., *Application of a fractional advection-dispersion equation*, Water Resources Res, 1403-1412,36,2000.

[9] Benson D., Wheatcraft S., Meerschaert M., *The fractional-order governing equation of Lévy motion*, Water Resources Res,1413-1423,36,2000.

[10] Benson, D., Wheatcraft, S., Meerschaert,Schumer, R., *Fractional dispersion, Lévy motion, and the MADE tracer tests*, Water Resources Res, 211-240, 42, 2001.

[11] Boleantu, M., *Fractional dynamical systems defined on fractional jet bundle and applications in economics*, arXiv:0710.0474, 2007.

[12] Boleantu, M., *Fractional dynamical systems and applications in economy*, Differential Geometry - Dynamical Systems, Vol.10, pp. 62-70, 2008.

[13] Bondarenko, A.N., Ivaschenko, D.S. , *Numerical methods for solving inverse problems for time fractional diffusion equation with variable coefficient*, Journal of Inverse and Ill-posed Problems. Volume 17, Issue 5, Pages 419–440, ISSN (Online) 1569-3945, ISSN (Print) 0928-0219, DOI: 10.1515/JIIP.2009.028, July 2009.

[14] Bondarenko, A.N., Ivaschenko, D.S. , *Generalized Sommerfeld problem for time fractional diffusion equation: analytical and numerical approach*, ournal of Inverse and Ill-posed Problems. Volume 17, Issue 4, Pages 321–335, ISSN (Online) 1569-3945, ISSN (Print) 0928-0219, DOI: 10.1515/JIIP.2009.022, June 2009.

[15] Brzeziǹski, D.W., Ostalczyk, P., *High-accuracy numerical integration methods for fractional order derivatives and integrals computations*, Bulletin of the Polish Academy of Sciences Technical Sciences, Vol. 62, No. 4, 2014.

[16] Brin, S., Page, L., *The Anatomy of a Large-Scale hypertextual Web search engine*, Computer networks and ISDN systems 30(1), 107-117, 1998..

[17] Butzer, P.L. and Westphal, U., *Introduction to fractional calculus*, Applications of Fractional Calculus in Physics, page 1, Singapore, 2000. World Scientific.

[18] Caffarelli, L, Silvestre, L, *An extension problem related to the fractional Laplacian*, http://arxiv.org/abs/math/0608640v2, 2008

[19] Caputo, M., , *Linear model of dissipation whose Q is almost frequency independent-II*, Geophys. J. R. Astr. Soc. 13: 529–539, 1967.

[20] Caputo, M., *Elasticitá e dissipazione*, Zanichelli, Bologna, 1969.

[21] Caputo, M., *Diffusion of fluids in porous media with memory.*, Geothermics. 28, 113-130, 1999.

[22] Caputo, M., *Models of flux in porous media with memory.*, Water Resour. Res. 36(3), 693-705, 2000.

[23] Caputo, M. and Kolari, J., *An analytical model of the Fisher equation with memory function.*, Altern. Persp. Financ. Account. 1, 1-16, 2001.

[24] Caputo, M. and Plastino, W., *Diffusion in porous layers with memory.*, Geophys. J. Int. 158, 385-396, 2004.

[25] Cesarone, F., *Modello di diffusione con memoria in membrane biologiche (Italian)*, GDegree Thesis, University of Rome "La Sapienza", 2004.

[26] Chen, S. and Liu, F., *ADI-Euler and extrapolation methods for the two-dimensional fractional advection-dispersion equation*, J. Appl. Math. and Computing. **26**, 295-311, 2008.

[27] Cloot, A. and Botha, J.F., *A generalized groundwater flow equation using the concept of non-integer order derivatives*, Water SA 32, 1-7, 2006.

[28] Chen, M., Deng, W. , *WSLD operators II: the new fourth order difference approximations for space Riemann-Liouville derivative*, Applied Mathematical Modelling, 38, 3244-3259, 2014.

[29] Chen, M., Deng, W. , *Fourth Order Accurate Scheme for the Space Fractional Diffusion Equations*, SIAM Journal on Numerical Analysis, 52(3), 1418-1438, 2014.

[30] Chen, Y., Liu, C., Qian, D., *On Riemann-Liouville and Caputo Derivatives*, Hindawi Publishing Corporation, Discrete Dynamics in Nature and Society, ID 562494, 2011.

[31] Chen, Z.Q., Meerschaert, M.M., and Nane, E., *Space-time fractional diffusion on bounded domains*, J. Math. Anal. Appl., 479-488, 393(2), 2012.

[32] Chou, H., Lee, B., and Chen, C, *The transient infiltration process for seepage flow from cracks*, Western Pacific Meeting, Advances in Subsurface Flow and Transport: Eastern and Western Approaches III, Eos Trans. AGU, 2006.

[33] Concezzi, M., Spigler, R., *Numerical solution of two-dimensional fractional diffusion equations by a high-order ADI method*, Commun. Appl. Ind. Math., 2013, printed online; **3**, No. 2 (2012), in print. [http://dx.doi.org/10.1685/journal.caim.421.]

[34] Concezzi, M., Spigler, R., *Some Analytical and Numerical Properties of the Mittag-Leffler Functions.*, (accepted) Fractional Calculus and Applied Analysis, 2014,.

[35] Concezzi, M., Garra, R., Spigler, R., *Fractional relaxation and fractional oscillation models involving Erdélyi-Kober integrals*, 2015, in preparation.

[36] Crank, J., Nicolson, P., Hartree, D. R., *A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type*, J. Appl. Math. and Computing Proceedings of the Cambridge Philosophical Society, 43, 50-67, 1947.

[37] Cui, M., *Compact finite difference method for the fractional diffusion equation*, J. Appl. Math. and Computing, 228, 20, 7792-7804, 2009.

[38] Cushman, B.R., *Beyond Adolf Fick: A new model of turbulent dispersion*, Presentation, 2006.

[39] Dannon, V. H., *The Delta Function*, Gauge Institute Journal **8**, No. 1, February 2012.

[40] Darcy, H., *Les fontaines publique de la Ville de Dijon*, Western Paris, France: Dalmont, 1856.

[41] De Espindola, J.J., da Silva Neto, and J.M., Lopes, E.M.O., *A generalized fractional derivative approach to viscoelastic material properties measurement*, Appl. Math. Comput. 164, 493-506, 2005.

[42] Deng, W., Tian, W. Y., Zhou, H., *A Class of Second Order Difference Approximations for Solving Space Fractional Diffusion Equations*, arXiv:1201.5949, 2010.

[43] Deng, W., Tian, W.Y., Zhou, H., *A Class of Second Order Difference Approximations for Solving Space Fractional Diffusion Equations*, arXiv:1201.5949, 2010.

[44] Deng, W., Tian, W. Y., Zhou, H., *Compact Finite Difference Approximations for Space Fractional Diffusion Equations*, arXiv:1204.4870, 2012.

[45] Deng, W., Chen, M., *Efficient numerical algorithms for three-dimensional fractional partial differential equations*, Journal of Computational Mathematics, 32(4), 371-391, 2014.

[46] Diethelm K., and Freed, A.D., *The FracPECE subroutine for the numerical solution of differential equations of fractional order*, S. Heinzel, T. Plesser (Eds.), Forschung und Wissenschaftliches Rechnen 1998, Gessellschaft fur Wissenschaftliche Datenverarbeitung, Gottingen, 1999, pp. 57-71.

[47] Diethelm, K., Ford, N. J., *Analysis of Fractional Differential Equations*, Journal of Mathematical Analysis and Applications 265, 229–248, 2002.

[48] Diethelm, K. *Efficient solution of multi-term fractional differential equations using P(EC)mE methods*, Computing **71** (2003), 305-319. [http://www.scielo.br/pdf/cam/v23n1/a02v23n1.pdf]

[49] Diethelm, K., Ford, N.J., and Freed, A.D. *Detailed error analysis for a fractional Adams method*, Numer. Algorithms **36** (1) (2004), 31-52.

[50] Di Giuseppe, E., Moroni M, Caputo, M., *Flux in Porous Media with Memory: Models and Experiments.*, Transp. Porous Med. DOI 10.1007/s11242-009-9456-4, 2009.

[51] Douglas, J., Gunn, J., *A general formulation of alternating direction methods Part I. Parabolic and hyperbolic problems*, Numer. Math., 6, 428-453, 1964.

[52] Douglas, J., Kimy, S., *Improved accuracy for locally one-dimensional methods for parabolic equations*, Math. Models Methods Appl. Sci, 11, 1563-1579, 2011.

[53] Douglas, J., Peaceman, D., *Numerical solution of two-dimensional heat flow problems*, American Institute of Chemical Engineering Journal, 1, 505-512, 1955.

[54] Dubbeldam, J. L. A., Milchev, A., Rostiashvili, V.G. , Vilgis, , T.A., *Polymer translocation through a nanopore - a showcase of anomalous diffusion*, arXiv:cond-mat/0701664.

[55] Elias, B.P. and Hajash, A., *Changes in quartz solubility and porosity due to effective stress: An experimental investigation of pressure solution*, Geology 20, 451-454, 1992. .

[56] Erdélyi, A., Magnus, W., Oberhettinger, F., and Tricomi, F.G., Higher Transcendental Functions [Bateman Manuscript Project], Vol.s 1, 2, and 3, McGraw-Hill, New York, 1953.

[57] Ervin, V.J., Roop, J.P., *Finite difference approximations for fractional advection-dispersion flow equations*, J. Comput. Appl. Math., 172, 65-77, 2004 .

[58] Ervin, V.J., Heuer, N., and Roop, J.P., *Numerical approximation of a time dependent, nonlinear, space-fractional diffusion equation*, SIAM Journal on Numerical Analysis,572-591,45(2), 2007.

[59] https://www.ma.utexas.edu/mediawiki/index.php/Fractional_Laplacian.

[60] Garrappa, R., *On linear stability of predictor-corrector algorithms for fractional differential equations. Internat. J. Comput. Math.*, **87**, No 10 (2010), 2281–2290.

[61] Garrappa, R., *Numerical evaluation of two and three parameters Mittag-Leffler functions*, submitted.

[62] Garrappa, R., http://www.mathworks.com/matlabcentral/fileexchange/48154-the-mittag-leffler-function.

[63] Geiser, J., *Iterative operator-splitting methods with high-order time integration methods and applications for parabolic partial differential equations*, J. Comput. Appl. Math. **217**, 227-242, 2008.

[64] Golub, G., Haveliwala, T., Kamvar, S., Manning, C., *Extrapolation Methods for Accelerating PageRank Computations*, Twelfth International World Wide Web Conference, 2003.

[65] Gorenflo, R., Luchko , Yu., and Mainardi, F., *Wright functions as scale-invariant solutions of the diffusion-wave equation*, J. Computational and Applied Mathematics,175-191, 118, 2000.

[66] Gorenflo, R., J. Loutchko, j., and Luchko, Yu., *Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z)$ and its derivative. Fract.Calc. Appl. Anal.* **5**, No 4 (2002), 491–518.

[67] Gorenflo, R., J. Loutchko, j., and Luchko, Yu., *Correction to "Computation of the Mittag-Leffler function $E_{\alpha,\beta}(z)$ and its derivative" [Fract. Calc. Appl. Anal.* **5***, No 4 (2002), 491–518]. Fract. Calc. Appl. Anal.* **6**, No 1 (2003), 111–112.

[68] Gorenflo, R., Kilbas, A. A., Mainardi, F., Rogosin, S.V., *Mittag-Leffler Functions, Related Topics and Applications*, DOI 10.1007/978-3-662-43930-2, Springer, 2014.

[69] Jafari, H. and Daftardar-Gejji, V., *Solving linear and nonlinear fractional diffusion and wave equations by Adomian decomposition*, Appl. Math. Comput. **180**, 488-497, 2006.

[70] Hairer, E., Lubich, C., and Schlichte, M., *Fast numerical solution of nonlinear Volterra convolution equations.*, *SIAM J. Sci. Statist. Comput.* **6**, No 3 (1985), 532–541.

[71] He, J., *Approximate analytical solution for seepage flow with fractional derivatives in porous media*, Comput. Methods Appl. Mech. Eng. **167**, 57-68, 1998.

[72] Hilfer, R„ *Threefold Introduction to Fractional Derivatives*, *Anomalous Transport: Foundations and Applications*, R. Klages et al. (eds.),Wiley-VCH,Weinheim, 2008, page 17 ISBN: 978-3-527-40722-4 .

[73] Huang, A. X., *A new decomposition for solving percolation equations in porous media*, Beijing, China, Third International Symposium on Aerothermodynamics of Internal Flows, pp. 417-420, 1996.

[74] Huang, Y., Oberman, A., *Numerical methods for the fractional Laplacian: a finite difference-quadrature approach*, SIAM J. Numer. Anal., Vol. 52, No. 6, pp. 3056–3084, 2014 .

[75] Iaffaldano, G., Caputo, M., and Martino, S., *Experimental and theoretical memory diffusion of water in sand*, Hydrol. Earth Syst. Sci. 10, 93-100, 2006.

[76] Jacquelin, J., *Use of fractional derivatives to express the properties of energy storage phenomena in electrical networks*, Laboratoires Alcatel de Marcoussis, 1984.

[77] Kaviany, M., *Extrapolations Principles of Heat Transfer in Porous Media*, Springer, New York, 1995.

[78] Kamvar, S., Haveliwala, T., Manning, C., and Golub, G., *Extrapolation Methods for Accelerating PageRank Computations*, Wealth International World Wide Web Conference, 2003.

[79] Kilbas, A. A., Srivastava, H.M., Trujillo, J.J., *Theory and Applications of Fractional Differential Equations*, North-Holland Mathematics Studies, 204, Elsevier, 2006 .

[80] Kowankar, K.M.,Gangal, A.D., *Fractional Differentiability of nowhere differentiable functions and dimensions*, CHAOS V.6, No. 4, 1996, American Institute of Physics.

[81] Landkorf, N.S., *Foundations of Modern Potential Theory*, Die Grundlehren der Mathematischen Wissenschaften 180, Springer-Verlag, New York, 1972.

[82] Le Mehaute, A. and Crepy, G., *Introduction to transfer motion in fractal media: the geometry of kinetics.*, Solid State Ion. 9 & 10, 1983.

[83] Lin, R. Liu, F., *Fractional high order methods for the nonlinear fractional ordinary differential equation*, Nonlinear Anal. **66**, 856-869, 2006

[84] Liu, F., Anh, V., Turner, I., *Numerical solution of the space fractional Fokker-Planck equation*, Comput. Appl. Math. **166**, 209-219, 2004.

[85] Liu, F., Anh, V., Turner, I., Zhuang, P.*Numerical simulation for solute transport in fractal porous media*, ANZIAM J. **45**, 461-473, 2004.

[86] Liu, F., Zhuang, P., Anh, V., Turner, I., Burrage, K., *Stability and convergence of the difference methods for the space-time fractional advection-diffusion equation.* Appl. Math. Comput. **191**, 12-20, 2007. .

[87] Liu, Q., Liu, F., Anh, V. Turner, I., *Approximation of the Lévy-Feller advection-dispersion process by random walk and finite difference method*, J. Comput. Phys. **222**, 57-70, 2007.

[88] Liu, Q., Liu, F., Turner, I., and Anh, V., *Numerical simulation for the* $3D$ *seepage flow with fractional derivatives in porous media*, IMA J. Appl. Math. **74** , 178-200, 2009.

[89] Loverro, A., *Fractional Calculus: History, Definitions and Applications for the Engineer*, Department of Aerospace and Mechanical Engineering, University of Notre Dame, Notre Dame, IN 46556, U.S.A., 2004.

[90] Luo, Z.-J., Zhang, Y.-Y. and Wu, Y.-X., *Finite element numerical simulation of three-dimensional seepage control for deep foundation pit dewatering*, Hydrodyn., 20, 596-602, 2008.

[91] Tenreiro Machado, J. A., *The Persistence of Memory*, International Conference on Fractional Differentiation and its Application, Catania, 2014.

[92] Mainardi, F., *Fractional relaxation-oscillation and fractional diffusion-wave phenomena*, Chaos Solitons Fractals 7, 1461-1477, 1996.

[93] Mainardi, F., *Fractional calculus: some basic problems in continuum and statistical mechanics*, Fractals and Fractional Calculus in Continuum Mechanics, 291-348, 378, 1997.

[94] Mainardi, F., *On the distinguished role of the Mittag-Leffler and Wright functions in fractional calculus*, Special Functions in the 21st Century: Theory and Applications, 6-8, Washington DC, USA, 2011.

[95] Mainardi, F., *On some properties of the Mittag-Leffler function $E_\alpha(-t^\alpha)$, completely monotone for $t > 0$ with $0 < \alpha < 1$"*, Discrete Continuous Dynamical Systems - Series B (DCDS-B), to appear; see also FRACALMO PRE-PRINT: `http://www.fracalmo.org`; and `arXiv: 1305.0161v3 [math-ph] 2013`.

[96] Mao, C.-X., Duan, X.-B., and and Li, Z.-Y., *Numerical Computation in Seepage Flow and Programs Application*, Nanjing, China: Hohai University Press (in Chinese), 1999.

[97] Marchuk, G.I., Shaidurov, V.V., *Difference Methods and Their Extrapolations*, Springer-Verlag, New York, 1983.

[98] http://reference.wolfram.com/mathematica/ref/MittagLefflerE.html.

[99] http://www.mathworks.com/matlabcentral/fileexchange/8738-mittag-leffler-function.

[100] Meerschaert, M. M. and Tadjeran, C., *Finite difference approximations for fractional advection-dispersion flow equations*, J. Comput. Appl. Math. **172**, 65-77. 2004.

[101] Meerschaert, M. M. and Tadjeran, C., *Variational formulation for the stationary fractional advection dispersion equation*, Numer. Methods Partial Differential Equations, 22, 558-576, 2006.

[102] Meerschaert, M. M., Scheffler, H., and Tadjeran, C., *Finite difference methods for two-dimensional fractional dispersion equation*, J. Comput. Phys. **211**, 249-261, 2006.

[103] Metzler, R., Jeon, J., Cherstvya, A.G., and Barkaid,L., *Anomalous diffusion models and their properties: non-stationarity, non-ergodicity, and ageing at the centenary of single particle tracking*, Phys. Chem. Chem. Phys., 2014, 16, 24128.

[104] Murray, J.D., *Mathematical biology. I,II,*, Interdisciplinary Applied Mathematics, Vol.s 17, 18, third ed., Springer-Verlag, New York, 2002.

[105] Neubert, M., and Caswell, H., *Demography and dispersal: Calculation and sensitivity analysis of invasion speed for structured populations*, Ecology **81** (6), 1613-1628, 2000.

[106] Nishimoto, K., *An essence of Nishimoto's Fractional Calculus*, Descartes Press Co. 1991.

[107] Ortigueira, M.D., *Riesz potential operators and inverses via fractional centered derivatives*, International Journal of Mathematics and Mathematical Sciences Article ID 48391 1-12, 2006.

[108] Ortigueira, M.D. and Batista, A.G., *On the relation between the fractional Brownian motion and the fractional derivatives*, Physics Letters A 372 958-968, 2008.

[109] Peaceman, D. W., and Rachford Jr., H.H., *The numerical solution of parabolic and elliptic differential equations*, J. Soc. Ind. Appl. Math. **3** (1), 28-41; doi:10.1137/0103003, MR 0071874, 1995.

[110] Pritchard, R., *Evaluation of Fractional Dispersion Models*, University of Reading School of Mathematics, Meteorology and Physics, 2008.

[111] Petford, N. & Koenders, M. A., *Seepage flow and consolidation in a deforming porous medium*, Geophys. Res. Abstr. **5**, 13329, 2003.

[112] Podlubny, I., *Fractional Differential Equations*, New York: Academic Press, 1999.

[113] Podlubny, I., *Matrix approach to discrete fractional calculus.*, Fractional Calculus and Applied Analysis 3 (4) 359-386, 2000.

[114] Podlubny, I., *The Evolution of Generalized Differentiation*, International Conference on Fractional Differentiation and its Application, Catania, 2014.

[115] Qin,J., Wang,T., *A compact locally one-dimensional finite difference method for nonhomogeneous parabolic differential equations*, Int. J. Numer. Meth. Biomed. Engng,28,128-142,2011.

[116] Roeloffs, E.A., *Fault stability changes induced beneath a reservoir with cyclic variations in water level.*, J. Geophys. Res. 93, 2107-2124, 1988.

[117] Roop, J. P., *Computational aspects of FEM approximation of fractional advection dispersion equations on bounded domains in* R^2, J. Comput. Appl. Math. **193**, 243-268, 2006.

[118] Rushton, K. R. and Redshaw, S. C., *Seepage and Groundwater Flow*. Brisbane, Australia: Wiley-Interscience Publication, 1979.

[119] Shi, M., Chen, Y., *Determination of permeability using fractal method for porous media*, Science in China (Series E), Vol. 44, No. 6, 2001.

[120] Simon T., *Comparing Fréchet and positive stable laws*, Electron. J. Probab. **19** (16) (2014), 1-25. http://arxiv.org/pdf/1310.1888v1.pdf

[121] Shen, S. and Liu, F., *Error analysis of an explicit finite difference approximation for the space fractional diffusion with insulated ends*, ANZIAM J. **46**, 871-887, 2005.

[122] Shen, S., Liu, F., Anh, V., and Turner, I., *The functional solution and numerical solution of the Riesz fractional advection-dispersion equation*, IMA J. Appl. Math. **73**, 850-872, 2008.

[123] Smith, G.D., *Numerical Solution of Partial Differential Equations: Finite Difference Methods*, Oxford Applied Mathematics and Computing Sciences Series, Oxford University Press, 1990.

[124] Sousa, E., *How to approximate the fractional derivative of order* $1 < \alpha \leq 2$, Int. J. Bifurcat., **22**, pp. 1250075, (2012) .

[125] Spotz, W. F., *High-Order Compact Finite Difference Schemes for Computational Mechanics*, 1995.

[126] Stein, E.M. , *Singular Integrals and Differentiability Properties of Functions*, Princeton Math. Ser. 30, Princeton University Press, Princeton, NJ, 1970.

[127] Sun, Z.Z., *Numerical Methods of Partial Differential Equations (in Chinese)*, Science Press, Beijing, 2005.

[128] Tadjeran, C., Meerschaert, M.M., Scheffler, H., *A second-order accurate numerical approximation for the fractional diffusion equation*, Journal of Computational Physics, 213,205-213,2006.

[129] Thambynayagam, R. K. M., *The Diffusion Handbook: Applied Solutions for Engineers*, McGraw-Hill Professional, ISBN 978-0-07-175184-1, 2001.

[130] Thusyanthan, N.I., and Madabhushi, S.P.G., *Scaling of seepage flow velocity in centrifuge models*, CUED/D-SOILS/TR326, http://www-civ.eng.cam.ac.uk/geotech new/publications/TR/TR326.pdf, 2003.

[131] Tolstykh, A. I., *High Accuracy Non-Centered Compact Difference Schemes for Fluid Dynamics Applications*, World Scientific, 1994.

[132] Tuan, V. K., and Gorenflo, R., *Extrapolation to the Limit for Numerical Fractional Differentiation*, J. Appl. Math. Mechanics **75**, 645-648, 1995.

[133] Tuan, V. K., and Gorenflo, R., *Extrapolation to the limit for numerical fractional differentiation*, ZAMM Z. Angew. Math. Mech. **75** (1995), 646-648.

[134] Yi Tian, W., Zhou, H., and Deng, W. *A Class of Second Order Difference Approximations for Solving Space Fractional Diffusion Equations*, Math. Comp., in press (arXiv:1201.5949, 2010).

[135] Yu, Q., Liu, F., Anh, V., and Turner, I., *Solving linear and nonlinear space-time fractional reaction-diffusion equations by Adomian decomposition method*, Int. J. Numer. Methods Eng. **74**, 138-158., 2008.

[136] Valdinoci, E., *From the long jump random walk to the fractional Laplacian*, Bol. Soc. Esp. Mat. Apl. SeMA, 49, pp. 33–44. 2009.

[137] Valério, D., and Machado, J.T., *On the numerical computation of the Mittag-Leffler function*, Commun. Nonlinear Sci. Numer. Simul. **19** (2014), 3419-3424; doi: http://dx.doi.org/10.1016/j.cnsns.2014.03.014.

[138] Wang, L., *Flows through porous media: a theoretical development at macroscale*, Transp. Porous Media 39, 1-24, 2000.

[139] Wangen, M., *Physical Principles of Sedimentary Basin Analysis*, Cambridge Univ. Press, 2010.

[140] Wang, C., Wang, T., *Extended locally one-dimensional finite difference and finite element schemes for non-homogeneous parabolic differential equations with nonhomogeneous boundary conditions (in Chinese)*, Numer. Math. J. Chinese Univ, 28, 138-150, 2006.

[141] it.mathworks.com/help/wavelet/ref/wfilters.html .

[142] Wang, Y., *Maximum Norm Error Estimates of ADI Methods for a Two-Dimensional Fractional Subdiffusion Equation*, Adv. in Math. Phys., Article ID 293706, 2013.

[143] Wangen, M., *Physical Principles of Sedimentary Basin Analysis*, Cambridge University Press, 2010.

[144] Witelski, T.P., and Bowenb, M., *ADI schemes for higher-order nonlinear diffusion equations*, Applied Numerical Mathematics, **45**,331-351, 2003.

[145] http://en.wikipedia.org/wiki/PageRank.

[146] Yuste, S.B., *Weighted average finite difference methods for fractional diffusion equations*, Journal of Computational Physics, Volume 216, Issue 1, 264-274, 2006.

[147] Yuste, S.B., Quintana-Murillo, J., *A finite difference method with non-uniform timesteps for fractional diffusion equations*, Computer Physics Communications 183, 2594–2600, 2014.

[148] Yuste, S.B., Quintana-Murillo, J., *Adaptive Finite Difference Method with Variable Timesteps for Fractional Diffusion and Diffusion-Wave Problems*, International Symposium on Fractional PDEs: Theory, Numerics and Applications, Salve Regina University, Newport, 2013.

[149] Zeng, C., and Chen, Y,Q., *Global Padé approximations of the generalized Mittag-Leffler function and its inverse*, `http://arxiv.org/abs/1310.5592`.

[150] Zhang, H., Liu, F., and Anh, V., *Numerical approximation of Lévy-Feller diffusion equation and its probability interpretation*, J. Comput. Appl. Math. **206**, 1098-1115, 2007.

[151] Zhou, H, Tian, W., Deng, W. , *Quasi-compact finite difference schemes for space fractional diffusion equations*, Journal of Scientific Computing, 56(1), 45-66, 2013.

[152] Zhuang, P., and Liu, F., *Implicit difference approximation for the time fractional diffusion equation*, J. Appl. Math. Comput. **22**, 87-99, 2006.

[153] Zhuang, P., Liu, F., Anh, V., and Turner, I., *New solution and analytical techniques of the implicit numerical method for the anomalous subdiffusion equation*, SIAM J. Numer. Anal. **46**, 1079-1095, 2008.